

AN INTRODUCTION TO CONTINUOUS DETER- MINISTIC SIMULATION

2nd EDITION

Jens Chr. Behrens

LYNGBY 1979

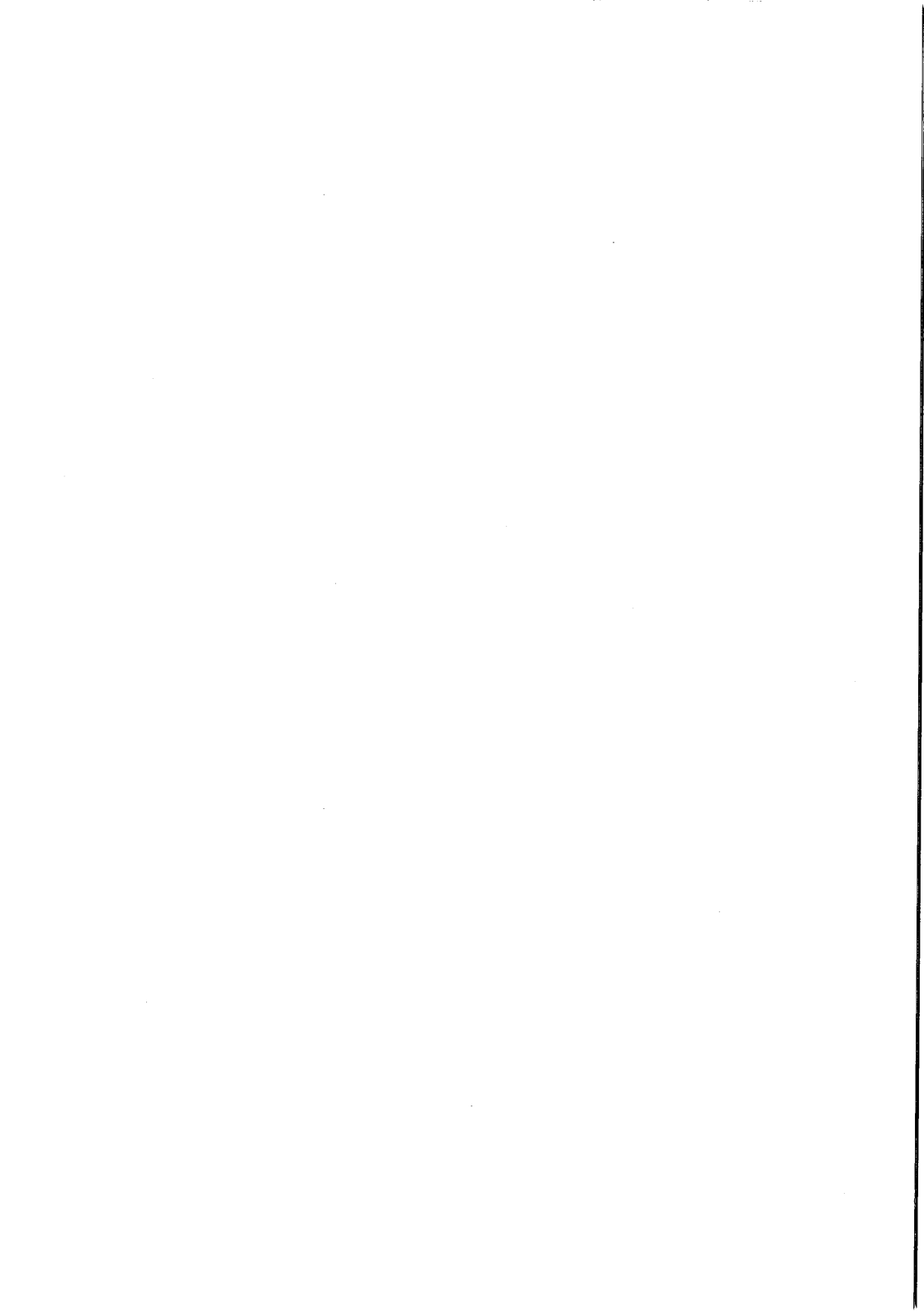
imsor

Trykt af **IMEO**, DTH

"When you've hit a really tough one, tried everything, racked your brain and nothing works, and you know that this time Nature has really decided to be difficult, you say, "Okay, Nature, that's the end of the *nice* guy", and you crank up the formal scientific method."

Robert M. Pirsig

"Zen and the Art of Motorcycle
Maintenance
- an Inquiry into Values".



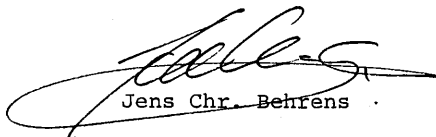
PREFACE

These notes have been prepared for the course "SIMULATION" given by IMSOR, the Institute of Mathematical Statistics and Operations Research, the Technical University of Denmark.

With regard to continuous simulation, the course only aims to give an introduction to the field. Thus, the introductory form of these notes. It should be strongly emphasized that the present edition is the first one. Criticisms are most welcome (and expected).

The author wishes to thank IMSOR, which by allocating two months wages and the secretarial assistance of Mrs. B. Pettersson made the work possible.

Lyngby, March 1978

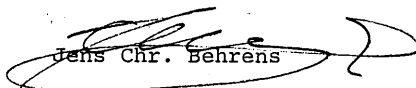


Jens Chr. Behrens

PREFACE to 2nd edition

The only major revision consists of the addition of section 3.2 (COLTS (Continuous Long-Term Simulation)) along with appendix B (JCL for COLTS-Syntax and examples). Furthermore, a number of corrections have been made. For corrections and criticisms the author wishes to credit H. Meyer, P.G. Thomsen, J. la.C. Jansen, M. Pedersen, A. Malmgren and J. Bueller.

Lyngby, May 1979



Jens Chr. Behrens

<u>CONTENTS</u>	Page
<u>0. Introduction</u>	5
0.1 Introduction to the Master Example	7
0.2 Systems	9
0.3 Models and a Taxonomy of Models	11
0.4 Introduction to the Modeling Process	15
0.5 Definition of Modeling Goals	17
<u>1. Systems Analysis</u>	19
1.1 Master example	20
<u>2. System Synthesis</u>	25
2.1 System Dynamics Flowcharts	26
2.2 Introduction to DYNAMO	31
2.2.1 Character set, Numerical values, Card format and Quantity names	31
2.2.2 Card Types	33
2.2.3 Time Sequencing and Equation writing	34
2.2.4 Arithmetic Operations, Functions and Tables	35
2.2.5 Master example, Equation writing	38
2.2.6 Direction cards	39
2.2.7 Master Example, Complete program	43
2.2.8 Further Features of DYNAMO	47
<u>3. Verification</u>	49
3.1 Introduction to Errors in Numerical Integration	50
3.1.1 DYNAMO integration - Euler's Method	50
3.1.2 Phase Error	51
3.1.3 Error Accumulation	54
3.1.4 Rounding Errors	60
3.1.5 Stability Problems	63

3.2	COLTS (<u>C</u> ontinuous <u>L</u> ong- <u>T</u> erm <u>S</u> imulation)	68
3.2.1	Specifying an integration routine	70
3.2.2	PHASE-plots	74
3.2.3	Further Features of COLTS	77
3.3	Model Behavior	77
3.4	Sensitivity Analysis	80
3.4.1	Analytical Approach	80
3.4.2	Master Example, Sensitivity	84
<u>4.</u>	<u>Validation</u>	93
4.1	Calibration	94
4.2	Goodness of Fit	97
<u>5.</u>	<u>Inference</u>	101
5.1	Master Example	102
	<u>REFERENCES</u>	115
	<u>APPENDIX A</u>	118
	JCL for DYNAMO at NEUCC	118
	<u>APPENDIX B</u>	119
	JCL for COLTS-syntax and Examples	119
	<u>INDEX</u>	125

0. INTRODUCTION

The branch of science occupied with computer simulation of continuous real-life events is a fairly young one. Furthermore, the branch has its background in several disciplines such as general systems theory, cybernetics, computer science, numerical analysis and mathematical analysis. Consequently, no final terminology for the branch has been settled. This leads to a choice when introducing continuous simulation. The present note, the terminology of Svend Clausen [1] was chosen for systems, while Mihram's terminology [2] is applied to models and to the modeling process.

Continuous simulation involves a number of fairly abstract concepts. In an attempt to ensure the readability, the descriptions are backed by a master example which will be used to illustrate the presented concepts throughout the entire note.

0.1 INTRODUCTION TO THE MASTER EXAMPLE

On the banks of a little idyllic lake lies Dump City. The little lake has a narrowing in the middle (see Fig. 1) and at what the locals calls "upper lake" a small paper industry, Wasters Inc., is situated. When the old folks at Dump City were young, they used to go bathing in the lake, but nobody does that anymore, since the lake waters have turned to a nasty intransparent green and is starting to smell rather bad on occasions. Both the city sewers and the waste water from Waster's Inc. are lead directly into the lake so nobody has any doubts that it is the pollution which is causing the wrecking of the lake by giving way to good living conditions for the algae.

On a nice spring day just after the ice has melted of the lake, the stench is unbearable in Dump City and at once a council meeting is held. Everybody agrees that something must be done about the pollution of the lake, but at once get into a fight about who, where and how much. The argument continues for several hours until finally a bright guy says: "Why don't we ask an Operations Researcher to do some deterministic continuous simulation of the problem so that we'll know?"

And so it all started.

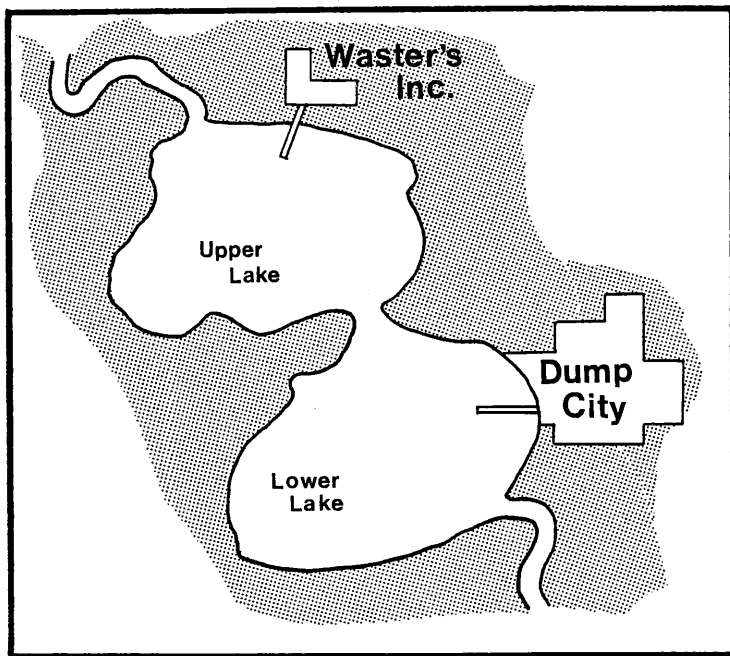


FIGURE 1: Map of Dump City and vicinity.

0.2 SYSTEMS

Several definitions of the concepts linked to systems have been given. The definitions adapted in this note are the ones of Svend Clausen [1].

Clausen states:

"A system is a 'part of reality' which in one way or another is detached (seems different) from the 'rest of reality'. This 'rest' is the environment of the system. Evidently, the 'rest' is also a system. These two systems (the system and its environment) are characterized by their individual structure*. But they are also interdependent. There is said to be a communication between them".

Since the concept 'system' is an important one, a rephrasing of Clausen's definition may be useful.

R. Buckminster Fuller [3] bases his definition on "universe":

"The universe is the aggregate of all of humanity's consciously-apprehended and communicated experience with the nonsimultaneous, nonidentical, and only partially overlapping, always complementary, weighable and unweighable, ever omni-transforming, event sequences."

From this, the following definition emerges:

"A system subdivides universe into all the universe outside the system (macrocosm) and all the rest of the universe which is inside the system (microcosm) with the exception of the minor fraction of universe which constitutes the system itself."

* The structure of a system is the characteristic laws pertaining to the system (authors note).

Any system S_2 is always part of at least one supersystem S_1 , if none other, then always the universe. Furthermore, S_2 may be experienced as a supersystem for yet other subsystems S_3 , S_4 etc.. This characteristic of systems is termed recursiveness of systems.

MASTER EXAMPLE

The task of defining the system with which we are going to be working, is that of isolating the part of reality (or universe) with which we are concerned.

The good citizens of Dump City were concerned with the pollution of their lake. Thus, our system must include the lake. We are not interested in the pollution outside the lake so this sets a geographical boundary to our system. Since the algae were the pest in the lake, the algae must be included. The algae needs food etc. in order to exist, but since we are not yet sure of all the factors we must take into account, we include the rather fuzzy boundary: the biological elements necessary for the existence of algae.

Summarizing, we state:

Our system is constituted by the lake water, the algae and the biological elements necessary for the existence of algae.

By defining our system, we have also defined its environment. We have included Wasters Inc., Dump City, the in- and out flowing river, the adjacent air and the sun in this environment. Thus, we have communications between our system and environment. Waster's Inc. and Dump City communicate with the system through sending in pollutants, the river through water and material flows and the sun and air through energy transfers.

We have now set the boundary and identified the communications, our system is defined.

0.3 MODELS AND A TAXONOMY OF MODELS

Clausen gives the following definition of the concept model:

"A model is a system showing resemblance to another system".

In consequence of this definition a statue of a person is a model of the person.

As it is most clearly stated in Fuller's universe-definition, immaterial communicated experiences are also part of the universe. Thus, the system constituted by Newton's law is a model of the interactions between two physical objects.

By presenting these two examples we have lead up to the first catagorization of models according to Mihram:

"1) Material models transformations of original physical objects, the representation of a complex system by another physical system assumed to be simpler than, yet similar to, the original system. 2) Formal models symbolic assertions in logical terms of an idealized relatively simple situation, the assertions representing the structural properties of the original factual system."

Another catagorization of models is focused on the role played by time. If the features or symbols of a model alters with time, the model is said to be dynamic, otherwise it is static.

Also randomness gives rise to a catagorization. If a model starting in one state invariably ends up in one and only one other state, the model is said to be deterministic. Does it only with a certain probability end up in one of several states, the model is stochastic.

A further refinement of the classification of material models is proposed by Mihram.

"Replicas spatial transformations of original physical objects in which the dimensionality of the modeled is retained in the replica; 2) quasi-replicas physical models in which one or more of the dimensions of the modeled object is missing; and 3) analogues physical models which bear no direct resemblance to the modeled phenomena, yet whose essential properties may be placed in a one-to-one correspondence with pertinent properties of the modeled."

Also the formal or symbolic model classification may be refined:

- 1) Descriptive models expressed in terms of one of man's natural languages subjected to manipulation and transformation only by means of the accepted rules of grammar.
- 2) Simular models expressed in a mixture of natural language and formal mathematics and not entirely manipulated by mathematical logic (e.g. algorithms).
- 3) Formalized-symbolic models which are manipulated entirely by operations of a well-formed mathematical discipline.

These different categorizations will be more meaningful when backed by actual examples. Fig. 2 shows a selection of exemplary systems and corresponding models. The figure is adapted from Mihram.

This note is devoted to the class of symbolic dynamic deterministic formalized models.

		MATERIAL				SYMBOLIC		
		REPLICA	QUASI-REPLICA	ANALOGUE	DESCRIPTIVE	SIMILAR	FORMALIZED	
STATIC	DETERMINISTIC	S:shape of Earth surface	S:Road network	S:Person	S:Human social association	S:Decision	S:Electrical currents	
		M:Earthen relief map	M:Road map	M:Statue	M:Ten commandments	M:Decision Logic table	M:Ohm's law	
	STOCHASTIC	S:Critical Dosage	S:Weather	S:Russian Roulette	S:Weather	S:Chess-player	S:Queue	
		M:Critical Dosage test	M:Weather map	M:Die toss	M:Weather report	M:Non-adaptive random chess playing program	M:Equilibrium Queue Length	
DYNAMIC	DETERMINISTIC	S:Railroad	S:The stars	S:Consumer prices	S:Human social association	S:Vehicle Scheduling	S:Prey/predator	
		M:Model train set	M:Planetarium show	M:Analog computer circuit for $\dot{Y}=Y$	M:Constitution	M:Savings algorithm	M:Lotka-Volterra equations	
	STOCHASTIC	S:Evolution	S:Endurance	S:Traffic noise	S:Evolution	S:Chess player	S:Prey/predator	
		M:Drosophila genetic experiment	M:Endurance test	M:White noise generator	M:Text on Darwinian evolution	M:Adaptive random chess playing program	M:Stochastic differential equation	

FIGURE 2: Exemplary Systems (S:) and models (M:)

0.4 INTRODUCTION TO THE MODELING PROCESS

The modeling process is a translation of the normal scientific method of reaching from observation to comprehension into a form suitable for dealing with modeling. This note follows the modeling process as suggested by Mihram by devoting one chapter to each of the steps of the process.

Mihram suggests the following procedure:

- 0) Definition of modeling goals:
Before commencing the actual modeling, the modeller must ascertain the goals of the modeling effort.
- 1) Systems Analysis: This is the initial stage of a model's development, during which the salient components, interactions, relationships, and dynamic behavior mechanisms of a system are isolated.
- 2) System Synthesis: This describes that stage of a model's development during which the model of the system's behavior is organized in accordance with the findings of the preceding Systems Analysis stage.
- 3) Verification: The third stage of a model's development is that in which the model's responses are compared with those which would be anticipated if indeed the model's structure was prepared as intended.
- 4) Validation: This is that stage of a model's development during which the responses emanating from the verified model are compared with corresponding observations of, and measurements from, the actual system in order to establish the verisimilitude of the model and the modeled.
- 5) Inference: The final stage of a model's development is concerned with the definition of experiments with and comparisons of responses from, the verified and validated model.

The modeling process is not necessarily a straight sequential process. Occurrences in the process may demand that the modeler returns to reconsider an earlier stage in the process. Fig. 3 from Mihram shows the possible transitions, which will be elaborated in the more detailed description of the process.

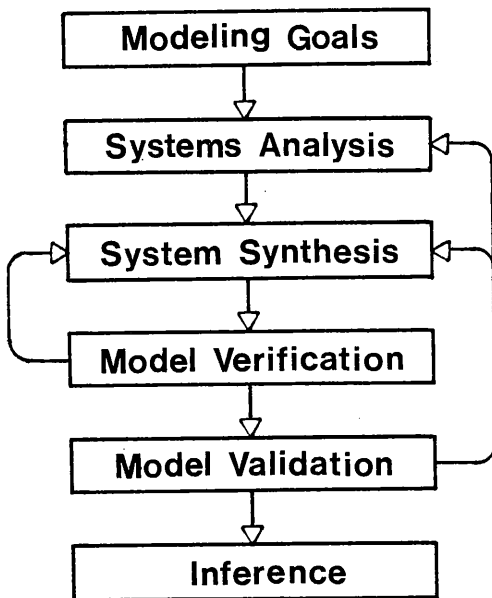


FIGURE 3: The modeling process.

0.5 DEFINITION OF MODELING GOALS

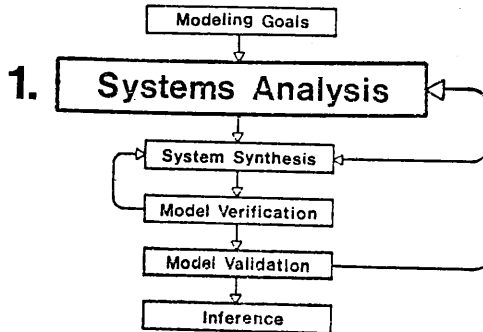
Prior to commencing the modeling process, the modeler has set a superior goal of the modeling in terms of solving a specific problem. In the case of our master example we have the superior goal of wanting to advise the good citizens of Dump City in their pollution troubles. But the superior goal does not necessarily lead to choosing modeling as a tool. Thus, the question one should ask with regard to the modeling process is not that of the superior aim, but the one pertaining to the use of models: "What would I do if I had finished the model which I may be about to build?"

Returning to the master example we ask the first subquestion: "Is there anything that we will be able to do if we had a finished model of the lake that we wouldn't be able to do without the model?"

Asking a biologist, he tells us that he is fully capable of giving a qualitative statement about the effects of different wastewater treatment plans, but nothing quantitative, so here lies the justification for trying to model the lake.

So, "What do we want to use the model for?" We want to be able to give a quantitative comparison of the effects of different wastewater treatment plans.

By stating this goal of the modeling effort, we have at the same time defined that our model must be build in such a way that we may perform this comparison.



At a given time, a system is distinguished by certain characteristics, it is said to be in a certain state. The state is indicated by a number of quantitative variables, the state variables. Due to certain properties of the system, the value of the state variables change with time, i.e. they undergo processes. The processes often depend on the value of state variables. Where this is the case, there is said to be a feedback from the state variable to the processes et vica versa. Previously, we have defined the term communication. In mathematical modeling, a one-way communication from the environment to the system is often termed a forcing function.

With these definitions at hand, the work to be carried out in the system analysis stage may be summarized as the identification of the following system characteristics:

- 1) System boundary
- 2) State variables of the system

- 3) Processes and feedback mechanisms.
- 4) Forcing functions and other communications.

1.1 MASTER EXAMPLE

In the section on systems we have defined our system as being constituted by the lake waters, the algae and the biological elements necessary for the existence of algae. The role played by the lake water is solely that of transportation. This leads to a subdivision of our system into two subsystems:

- a) a transportation subsystem constituted by the lake water
- b) a biological subsystem constituted by the algae and the biological element necessary for the existence of algae.

Postponing the treatment of the transportation subsystem, we exemplify the system analysis with the biological subsystem.

We find that we need the help of a biologist in order to analyze the subsystem. He comes up with the following information:

The algae may be measured as algal biomass. The algal biomass grows through photosynthesis in which inorganic nutrients in the water phase are synthesized into organic matter utilizing solar energy. The photosynthetic activity is observed to be proportional to the algal biomass if energy and nutrients are in surplus. When an algae dies, it is sedimented. In the sediment layers mineralization processes take place decomposing dead organic matter to free inorganic nutrients which are released to the waterphase. The external forcing on the photosynthesis from daylength and temperature has experimentally been shown to vary sinusoidally over the year with a maximum inhibition around December 20th. Furthermore, the photosynthesis may be limited by lack of nutrients. Experiments have shown a hyperbolic dependance of photosynthesis on the availability of nutrients in the waterphase. The sedimentation has been observed to be proportional to the algal

biomass. The mineralization has experimentally been determined to be proportional to the amount of organic bound nutrients available in the sediment.

From the biologist's description of the lifecycle of algae we may derive, that if we know the algal biomass, the amount of inorganic nutrients in the waterphase and the amount of organic bound nutrients in the sediment, we know the state of the system. Thus, our biological subsystem has the following statevariables:

1. Algal biomass (A)
2. Nutrients (N) in the waterphase
3. Organic Nutrient in Sediment (ONS).

From the description we may also isolate the following processes:

- 1: Photosynthesis or Primary Production (PR)
- 2: Sedimentation of Algae (SEDA)
- 3: Mineralization (MINS)

In the description we may also detect several feedback mechanisms, e.g. the proportionality between sedimentation and algae. A convenient way of representing feedback mechanisms is in a causal diagram. In a causal diagram feedbacks are shown as arrows. These arrows are often signed so that a positive influence (a positive feedback) is marked with a "+" while a negative is marked "-". Fig. 4 shows the causal diagram which may be derived from the biologist's description.

In separating the transportation subsystem from the biological subsystem, we have made the transportation subsystem part of the environment of the biological subsystem. Thus, we have the following forcing functions on the biological subsystem:

1. Transportation.
2. Nutrient input with wastewater

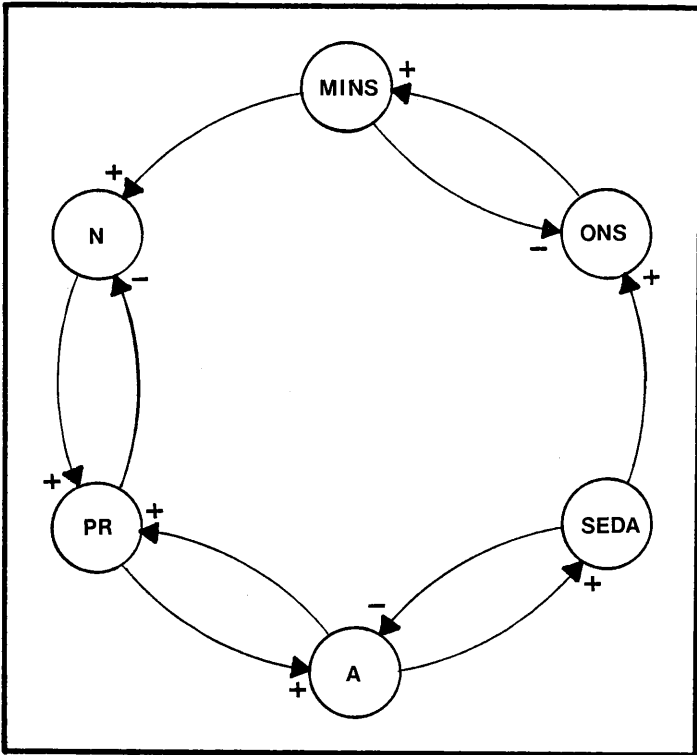
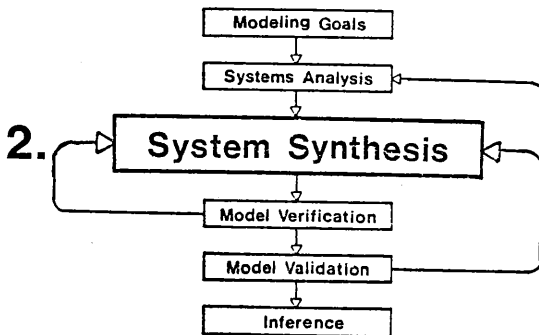


FIGURE 4: Causal diagram showing the feedbacks of the biological subsystem. In determining whether you have a positive or a negative feedback, pose the question: "Will an increase in the entity, from which the feedback originates, cause an increase in the entity at which the feedback terminates?" If "yes", then the feedback is positive, otherwise, it is negative.

3. Forcing on photosynthesis from daylength and temperature.

We have now completed the system analysis of the biological subsystem.



This stage of the models development is occupied with the actual construction of the model. In the stage, the first choice to be made concerns the type of model to be constructed. In our case, the behavior of the system is definitely time-dependant leading to the need for a dynamic model. The choice in our case between a material and a symbolic model is less clear. Principally, there would be nothing stopping us from using analogue circuits for our modeling. However, analog computation is outside the scope of this book. Jackson [4] gives an introduction to analogue computation without prerequesting a knowledge of electronics. Here we choose to build a symbolic model. From the biologist's description we do not expect any randomness in the behavior of our system, so we choose a deterministic model. Summarizing, we choose to build a symbolic dynamic deterministic formalized model of the biological subsystem.

Having made this choice, we have a wide variety of methods to choose from in the actual modeling. Here we choose to apply an approach based on System Dynamics and the simulation language DYNAMO.

System Dynamics was developed over the last thirty years at the Massachusetts Institute of Technology. From a mathematical point of view, System Dynamics is a pictorial representation of a restricted class of deterministic models, i.e. models which can be described analytically by a finite number of coupled first order differential equations.

2.1 SYSTEM DYNAMICS FLOWCHARTS

Fig. 5 shows the System Dynamic flowchart symbols. Apart from the symbols of the figure, a delay-symbol exists but has been left out, since it is not used in the note.

For the use of the delay-symbol, see e.g. Pugh [5].

The use of the symbols can be exemplified with the following:

Consider the experimental set-up of fig. 6.

The object of the experiment is to fill the sink to a black ring left by the last user. The state of the system is indicated by the quantity of water in the sink, i.e. the water quantity is the statevariable (or level) of the system. The physical flow to be conserved within the system boundary is water. We are not interested in the quantity of water of the waterworks or in the sewers. We may now draw the material flow in the system: From the waterworks (w), which is a level of no interest to us (a source) the water flows into the sink (SI) and from the sink out to the sewers (S), another level of no interest to us (a drain). The flow into the sink is regulated by the tapping process (or rate) T. Depending on whether the water level in the sink is above or below the black ring, the experimenter will open or close the tap, i.e. there is an

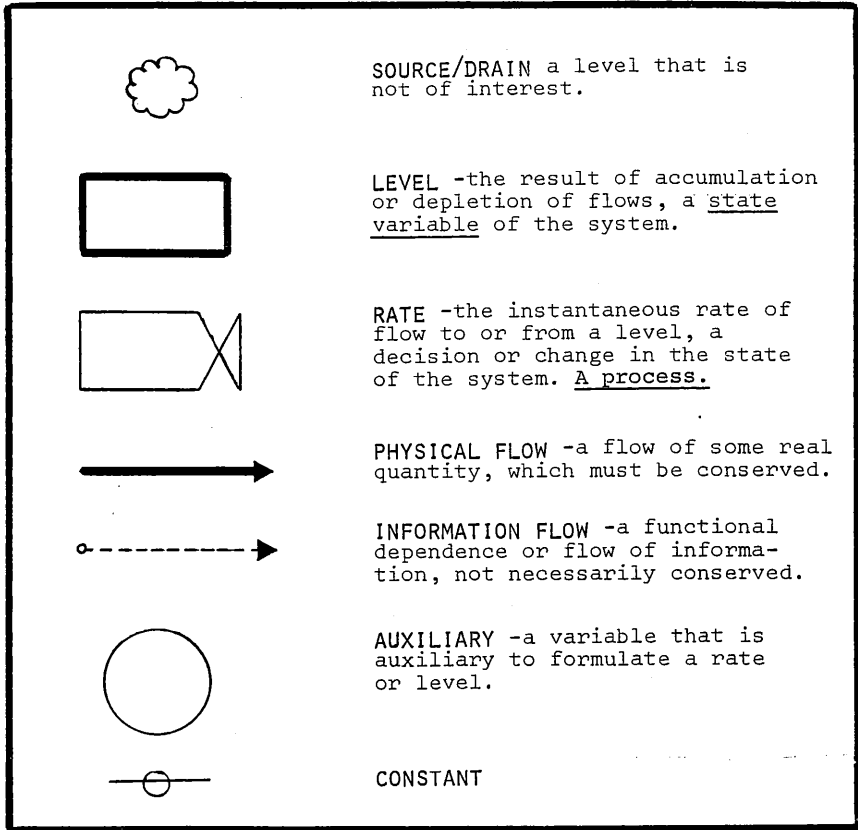


FIGURE 5: System Dynamic flow chart symbols.

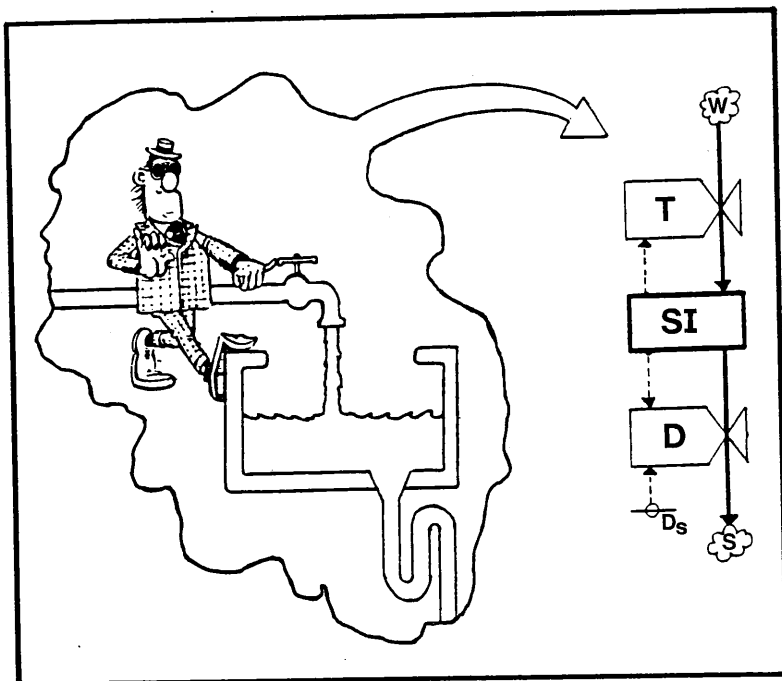


FIGURE 6: Example of the application of System Dynamic flow-chart symbols. Courtesy to D. Martin.

information flow from SI to T indicated by the dotted line. The flow out of the sink is regulated by the draining process (or rate) D. The draining rate depends on the size of the drain which is a constant D_s , i.e. there is an information flow from the constant D_s to D. Furthermore, the water level will determine the draining rate (no water in sink, no draining) so an information flow from SI to D must be drawn.

It should be noted, that the System Dynamics flowchart only shows the information flows, not the functional dependence, e.g. not whether the draining is directly proportional to the size of the drain or not.

MASTER EXAMPLE:

Fig. 7. shows a System Dynamic flowchart for the isolated biological subsystem.

We have chosen to measure the Algal biomass (A) and the Organic Nutrient in Sediment (ONS) by their content of Nutrients (N). Thus, the material flow in the biological subsystem is a flow of Nutrients. From the biologist's description we know that there is a flow from A to ONS in the form of the process Sedimentation of algae (SEDA). The mineralization (MINS) causes a flow from the sediment to the free nutrients N. Through the photosynthesis (primary production PR) the nutrients flow to the algae. We have now drawn the levels, rates and material flows of the biological subsystem. Turning to the information flows we remember that the mineralization was proportional to the sediment. Thus, an information flow from the ONS to MINS must be indicated.

Furthermore, we need information on the size of the proportionality constant, K_1 , leading to the information flow from K_1 to MINS. The sedimentation of algae is proportional to the algal biomass and we may, therefore, draw the information flows to SEDA parallel to those of MINS. The primary production PR is proportional to the algal biomass leading to the information

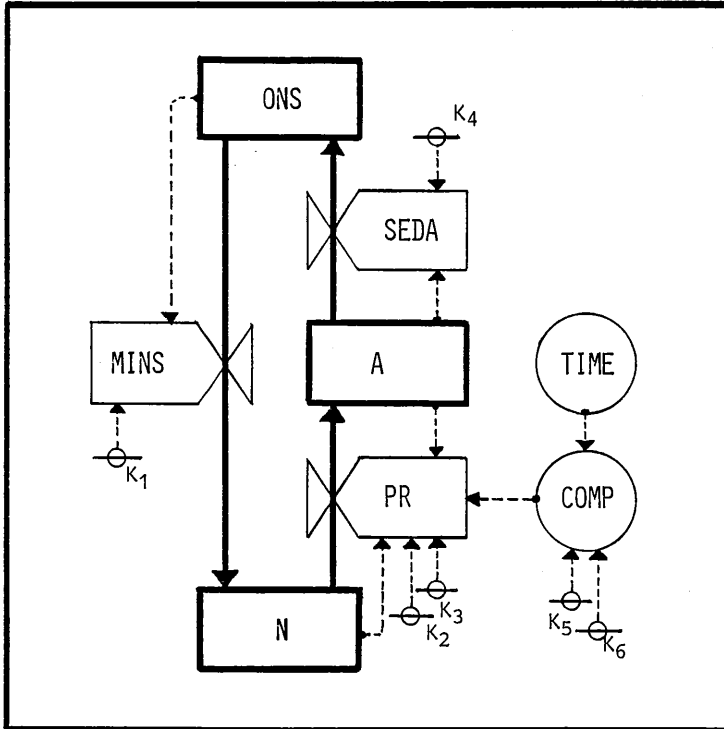


FIGURE 7: System Dynamic flowchart for the isolated biological subsystem.

flows from A and K_2 to PR. PR has a hyperbolic dependence on N with the resulting information flows from N and K_3 to PR. Furthermore, the external compulsion or forcing on PR varies sinusoidally over the year. We use an auxiliary, COMP, to calculate this forcing. The forcing depends on the time of the year and on two constants K_5 and K_6 with the resulting information flows from TIME, K_5 and K_6 to COMP. Since PR depends on the external compulsion, we have the information flow from COMP to PR.

2.2 INTRODUCTION TO DYNAMO

The purpose of the Simulation language DYNAMO is to give a mean of translating models formulated as System Dynamics flow-charts into a machine readable form.

The purpose of this section is to give an introduction to DYNAMO enabling the reader to formulate small models in the language. The object is not to give a full description of DYNAMO. The reason for this limitation is the existence of an easily readable manual, "DYNAMO II USER'S MANUAL" by Pugh [5] prerequesting no knowledge of computers what so ever. Readers wishing to formulate more complex models in DYNAMO are referred to this manual. The section "2.2.8 Further Features of DYNAMO" gives a short description of the DYNAMO-facilities not treated in this note.

2.2.1 Character set, numerical values, card format and quantity names.

The following character set may be used in input to DYNAMO:

upper case alphabetic letters: AB...Z
underscore: _
numbers: 0,1...9
arithmetic operators: =+*/*(),
decimal point or period: .
dollar sign: \$
space:

Numerical values are expressed in up to 8 digits in addition to sign and decimal point. Decimal point to the right of a number may be dropped. 10 to the power n is expressed as the letter E followed by the number n . Examples of numerical values:

13. +999.1 -0.003

5E+3(or 5E3) 2.1E-4

The input to DYNAMO is punchcard oriented, i.e. all input statements take up one punch card (80 characters). Each card is divided into 4 fields:

- 1) starting in column 1 is the card type specification field
- 2) separated from field 1) by one or more blanks is the equation or direction field
- 3) separated from field 2) by one or more blanks is the comment field
- 4) columns 73 to 80 may be used for numbering the cards.

Quantity names consists of from 1 to 7 alphabetic, numerical characters or underscores, the first of which must be alphabetic.

Examples:

X
VAR12
STA_VAR
Y14B3

The following quantity names have a special significance in DYNAMO and should not be used for user defined quantities: DT, LENGTH, PRTPER, PLTPER and TIME.

2.2.2 Card types.

DYNAMO was written with the intention of enabling the translation of System Dynamic flow chart to a machine readable form. Thus, a number of card types indicate a System Dynamic flow-chart symbol:

card type	indicates
L	Level
R	Rate
A	Auxiliary
C	Constant.

Furthermore, constants in the form of tables are allowed

T indicates a Table.

Differential equations require an initial condition in order to be defined.

N indicates iNitial condition

When performing a computer simulation you need to give specifications concerning the simulation, e.g. what time interval do you want to simulate, what the output should look like and so on. In DYNAMO, certain card types indicates this type of information:

PRINT defines the values to be printed in output

PLOT defines the quantities to be plotted

NOTE indicates that the entire card is a comment

SPEC defines the length of the simulation, printout intervals, plot intervals and the steplength to be used in the simulation

* defines a title to be printed on top of each page of the output

- RUN specifies that the entire inputdeck has been given and that the simulation may start
- X indicates that the card is a continuation of the preceding card. The preceding card may be broken after a quantity name. The continuation must start in columns 3 to 9.

2.2.3 Time sequencing and equation writing

The DYNAMO variable TIME contains the time reached in the simulation.

DYNAMO operates with 3 points in time, see fig. 8.

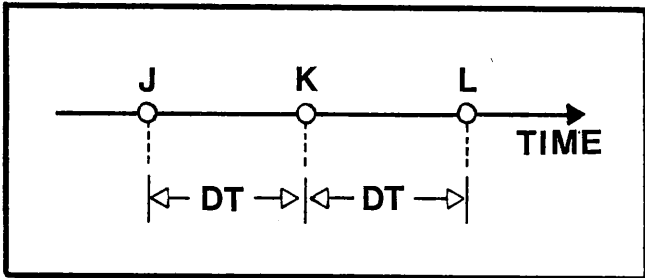


FIGURE 8: DYNAMO time sequencing.

A subscript is used to indicate the time in the equation writing. L.J indicates quantity L at time J, R.JK indicates quantity R in the time interval from J to K.

DYNAMO performs the integration of the level equations according to Euler's method, i.e. through a tangential projection. Denoting the timestep (the length of the projection interval) by DT, the level equation according to Euler may be expressed as

$$\text{LEVEL.K} = \text{LEVEL.J} + \text{DT} * (\text{Rate of change in LEVEL}).\text{JK}$$

The computational sequence is level, auxiliary, rate. Once the rate has been computed, the time is advanced DT units. Quantities computed at time K are then regarded as being quantities at time J and so on. The time-administration along with the computational sequence leads to that only some timesubscripts are allowed on left and right hand sides of the equal-sign respectively. The rules are summarized in fig. 9.

Quantity Type on Left of Equation	Subscript on Left	Subscripts on Quantities on Right if Quantity is				
		L	A	R	C	N
L Level	K	J	J	JK	none	none
A Auxiliary	K	K	K	JK	none	none
R Rate	KL	K	K	JK	none	none
C Constant	none	np	np	np	np	np
N iNitial value or computed constant	none	none	none	none	none	none

np = not permitted

FIGURE 9: DYNAMO timesubscripts.

2.2.4 Arithmetic Operations, functions and tables

The following arithmetic operations are allowed in DYNAMO:

+ - * /

Signifying addition, subtraction, multiplication and division.

Examples:

$$A*-B = A*(-B)$$

$$A/B+C = (A/B)+C$$

The following functions are available

DYNAMO SPELLING	FUNCTION	RESTRICTION ON ARGUMENT
EXP(A)	e^A	$A \leq 174$
LOGN(A)	$\log_e(A)$	$A > 0$
SQRT(A)	\sqrt{A}	$A \geq 0$
SIN(A)	sine(A)	$A < 823.000$
COS(A)	cosine(A)	$A < 823.000$

A to the power B may be computed as

$$\text{EXP}(B*\text{LOGN}(A))$$

It should be noted that DYNAMO uses the exact equality $2\pi = 6.283$ and not the correct value of 2π .

Other functions:

CLIP(P,Q,R,S) or FIFGE(P,Q,R,S)

CLIP = P if $R \geq S$
FIFGE

CLIP = Q if $R < S$
FIFGE

MIN(P,Q)

MIN = P if $P < Q$

MIN = Q if $P \geq Q$

MAX(P,Q)

MAX = P if $P \geq Q$

MAX = Q if $P < Q$

NOISE() gives a uniformly distributed random number
between -.5 and +.5

STEP (HGHT,STTM)

STEP = 0 if TIME < STTM
STEP = HGHT if TIME \geq STTM

SWITCH(P,Q,R) or FIFZE(P,Q,R)

SWITCH
FIFZE = P if R = 0

SWITCH
FIFZE = Q if R \neq 0

Timerelated events, such as the STEP-function follows the following rule:

An action will occur if TIME is greater than the action time or if the two agree within $DT/2$.

Entries to a DYNAMO table must be given at equidistant values of the independent variable.

Example from [5]:

Figure 10 shows the graph to be tabulated.

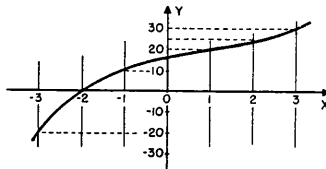


FIGURE 10: Table example.

From Fig. 10 we get the following corresponding values:

X	-3	-2	-1	0	+1	+2	+3
Y	-20	0	10	16	20	24	30

we may now form a table for Y under the name

YTAB

T YTAB = -20/0/10/16/20/24/30

To extract information from a table, DYNAMO provides a table-look-up-function:

TABLE(TAB,X,XLOW,XHIGH,XINCR)

TAB - name of the table
X - independent variable
XLOW - lowest value of range of independent variable
XHIGH - highest value of range of independent variable
XINCR - increment of independent variable

supposing that Y on fig. 10 is an auxiliary, we may find its value in the following statement

A Y.K = TABLE(YTAB,X.K,-3,3,1)

TABLE uses a linear interpolation.

2.2.5 Master Example, equation writing

With the basic information on equation writing at hand, we are ready to write the DYNAMO statements for the biological subsystem.

Starting with the nutrients N we turn to figure 7. On the figure we see, that the flow into N is the mineralization MINS and the flow out of N is the primary production PR. Thus, the rate of change in N must be MINS-PR. Since N is a level, we get the following equation:

$$L N.K = N.J+DT*(MINS.JK-PR.JK)$$

Inspecting the in- and outgoing flows in connection with the two other levels we get

$$L A.K = A.J+DT*(PR.JK-SEDA.JK)$$

$$L ONS.K = ONS.J+DT*(SEDA.JK-MINS.JK)$$

We now turn to formulate the Rate-equations.

The mineralization is proportional to the amount of nutrients in the sediment with proportionality constant K_1 . Thus, (remembering the time-indexing of figure 9.)

$$R MINS.KL = K1*ONS.K$$

The sedimentation is proportional to the algal biomass with proportionality constant K_4 , i.e.

$$R SEDA.KL = K4*A.K$$

Asking the biologist again, he provides the information that the inhibition on the photosynthesis by daylength and temperature has been observed to vary according to

$$.5 \cdot \left[1 + 1 \cdot \sin\left(\frac{2\pi}{365} * (T-80)\right) \right]$$

where T is the number of the day in the year. Defining $K_5 = 365$ and $K_6 = 1$ we may write the following auxiliary equation in order to compute the external compulsion (COMP)

$$A COMP.K = .5 * \left(1 + K6 * \sin\left(\frac{6.283}{K5} * (TIME.K-80)\right) \right)$$

Remembering that the photosynthesis was proportional to the algal biomass with proportionality constant K_2 and had a hyperbolic dependence on the availability of nutrients we get the following rate equation

$$R \text{ PR.KL} = K2 * A.K * \left(\frac{N.K}{N.K + K3} \right) * \text{COMP.K}$$

In our definition of the rates we have used a number of constants. These have to be defined:

C K1 = .001
C K2 = .07
C K3 = .12
C K4 = .01
C K5 = 365.
C K6 = 1.

A part of the definition of a set of differential equations is the definition of the initial conditions. In DYNAMO, all level equations must be initialized. Other quantities may be initialized if desired.

Initial conditions

N N = 1.4
N A = .9
N ONS = 12.8

We have now completed the equations for the biological subsystem and we may turn to the administration of the simulation.

2.2.6 Direction cards

The different direction cards have been mentioned in section 2.2.2 A little more detailed description is given here.

TITLE-card is marked with an asterix. The text separated from the asterix by one or more blanks will be listed on top of each page of output. The text may not be longer than 50 characters.

Example:

* MODEL OF THE BIOLOGICAL SUBSYSTEM

PLOT-card

It is possible to request a printerplot of quantities versus time. Automatic or selfchosen scaling can be applied.

Example:

In the master example we wish to plot the algal biomass with 0 as lower bound and 4 as upper bound. Furthermore, we wish Algal biomass to be marked with an "A" on the plot. We also want nutrients plotted as "N" and sediment as "0" on a scale chosen by DYNAMO.

PLOT A = A(0,4)/N=N/ONS=0

The interval between points in the plot is contained in the auxiliary

PLTPER (PLoTPERiode)

PRINT-card

The print-card causes a printout of the values listed on the card.

Example:

In the Master Example we wish a print out of the algal biomass, the nutrient, the sediment and the external compulsion

PRINT A,N,ONS,COMP

The time-interval between printouts is contained in the auxiliary

PRTPER (PRinTPERiode)

SPEC-card

Four variables have a special significance in the running of a simulation:

DT: The timestep in the integration

LENGTH: The time at which the integration is terminated.

PRTPER: Time interval between printout. PRTPER = 0 suppresses printing.

PLTPER: Time interval between plot-points. PLTPER = 0 suppresses plotting.

These four variables are all auxiliary and may be defined in an auxiliary statement. Furthermore, they may be given on a SPEC-card.

To the setting of the time step DT, the following common rule may be helpful:

Using a given stepsize h a solution is obtained. This solution is compared to another solution obtained using a stepsize $h/2$. If the two agree, it is reasonable to assume that a usable stepsize has been found. Otherwise, keep halving the stepsize until agreement.

Example:

In the master example, we wish to try a stepsize of .25 . We wish to run the model for 2 years (730 days) and we want a print out every 10 days. Furthermore we want a plot of every 10th day of the second year (day 365 to day 730).

SPEC DT = .25/LENGTH = 730/PRTPER = 10

A PLTPER.K = STEP(10,365)

RUN-card

The RUN-card indicates that all statement necessary for the simulation have been given and that execution may start. It is possible to mark the run with a title which will be printed on top of the output pages following the text from the asterix card. The title may contain up to 50 characters.

Example:

RUN FIRST MODEL TEST

2.2.7 Master Example, Complete program

Using the direction cards specified in section 2.2.6 the following complete program for the isolated biological subsystem emerges:

```
* MODEL OF BIOLOGICAL SUBSYSTEM
NOTE
NOTE MASTER EXAMPLE
NOTE
L N.K=N.J+DT*(MINS.JK-PR.JK) NUTRIENTS
L A.K=A.J+DT*(PR.JK-SEDA.JK) ALGAL BIOMASS
L ONS.K=ONS.J+DT*(SEDA.JK-MINS.JK) SEDIMENT
NOTE
R MINS.KL=K1*ONS.K
R PR.KL=K2*A.K*(N.K/(N.K+K3))*COMP.K
R SED.A.KL=K4*A.K
NOTE
A COMP.K=.5*(1+K6*SIN((6.283/K5)*(TIME.K-80)))
NOTE
N N=1.4
N A=.9
N ONS=12.8
NOTE
C K1=.001
C K2=.07
C K3=.12
C K4=.01
C K5=365.
C K6=1.
```

NOTE
NOTE RUN ADMINISTRATION
NOTE
A PLTPER.K=STEP(10,365)
PRINT A,N,ONS,COMP
PLOT A=A(0,4)/N=N/ONS=0
SPEC DT=.25/LENGTH=730/PRTPER=10
RUN FIRST MODEL TEST

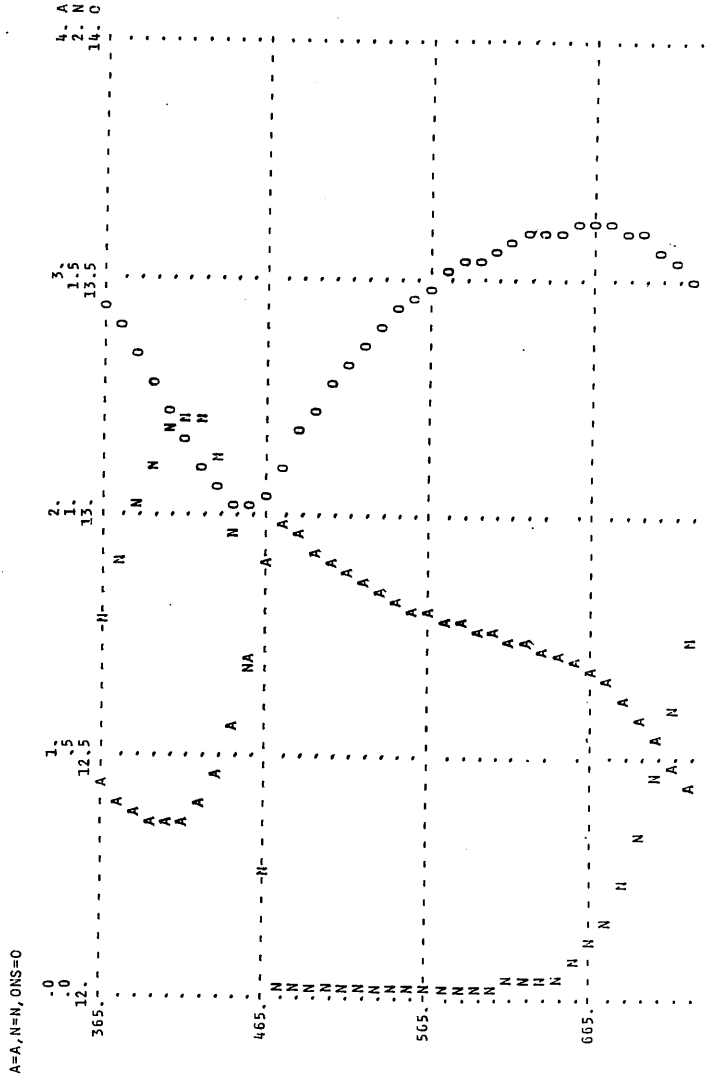
The JCL-statements (Job Control Language) necessary to run the job at NEUCC, the Technical University of Denmark, are given in appendix A.

The following output is generated by DYNAMO and requested by PRINT A,N,ONS,COMP with PRTPER = 10 :

TIME	A	N	ONS	COMP
E+00	E+00	E+00	E+00	E+00
.0	.9000	1.4000	12.800	.00935
10.	.8247	1.5169	12.758	.03309
20.	.7710	1.6183	12.711	.07063
30.	.7418	1.6988	12.659	.12086
40.	.7405	1.7526	12.607	.18229
50.	.7719	1.7717	12.556	.25312
60.	.8450	1.7436	12.511	.33124
70.	.9745	1.6487	12.477	.41436
80.	1.1854	1.4555	12.459	.50000
90.	1.5165	1.1155	12.468	.58564
100.	2.0116	.5710	12.517	.66876
110.	2.4526	.0282	12.619	.74688
120.	2.3541	.0126	12.733	.81771
130.	2.2519	.0123	12.835	.87914
140.	2.1601	.0122	12.927	.92937
150.	2.0778	.0122	13.009	.96691
160.	2.0039	.0124	13.083	.99065
170.	1.9376	.0128	13.149	.99988
180.	1.8780	.0134	13.208	.99434
190.	1.8244	.0142	13.261	.97419
200.	1.7761	.0153	13.308	.94002
210.	1.7325	.0166	13.350	.89285
220.	1.6937	.0185	13.385	.84449
230.	1.6600	.0211	13.415	.79451
240.	1.6318	.0244	13.440	.74349
250.	1.6080	.0284	13.460	.69186
260.	1.5888	.0331	13.475	.64000
270.	1.5738	.0384	13.485	.58833
280.	1.5624	.0443	13.490	.53716
290.	1.5541	.0508	13.490	.48668
300.	1.5484	.0579	13.485	.43700
310.	1.5447	.0656	13.475	.38833
320.	1.5424	.0739	13.460	.34086
330.	1.5411	.0828	13.440	.29468
340.	1.5405	.0922	13.415	.24986
350.	1.5405	.1022	13.385	.20668
360.	1.5411	.1128	13.350	.16468
370.	1.5424	.1239	13.310	.12416
380.	1.5441	.1356	13.265	.08568
390.	1.5464	.1479	13.215	.04968
400.	1.5493	.1608	13.160	.01668
410.	1.5528	.1743	13.100	.00000
420.	1.5568	.1884	13.035	.00000
430.	1.5614	.2031	12.965	.00000
440.	1.5664	.2184	12.890	.00000
450.	1.5719	.2343	12.810	.00000
460.	1.5779	.2508	12.725	.00000
470.	1.5844	.2679	12.635	.00000
480.	1.5914	.2856	12.540	.00000
490.	1.5989	.3039	12.440	.00000
500.	1.6068	.3228	12.335	.00000
510.	1.6151	.3423	12.225	.00000
520.	1.6238	.3624	12.110	.00000
530.	1.6329	.3831	11.990	.00000
540.	1.6424	.4044	11.865	.00000
550.	1.6523	.4263	11.735	.00000
560.	1.6626	.4488	11.600	.00000
570.	1.6733	.4719	11.460	.00000
580.	1.6844	.4956	11.315	.00000
590.	1.6959	.5200	11.165	.00000
600.	1.7078	.5451	11.010	.00000
610.	1.7201	.5708	10.850	.00000
620.	1.7328	.5971	10.685	.00000
630.	1.7459	.6240	10.515	.00000
640.	1.7594	.6516	10.340	.00000
650.	1.7733	.6800	10.160	.00000
660.	1.7876	.7091	9.975	.00000
670.	1.8023	.7389	9.785	.00000
680.	1.8174	.7694	9.590	.00000
690.	1.8329	.8006	9.390	.00000
700.	1.8488	.8325	9.185	.00000
710.	1.8651	.8651	8.975	.00000
720.	1.8818	.8984	8.760	.00000
730.	1.8989	.9324	8.540	.00000

OUTPUT requested by PLOT A = A(0,4)/N = N/ONS = 0/ with PLTPER.K = STEP(10,365)

PAGE 4 MODEL OF BIOLOGICAL SUBSYSTEM 3/29/78 FIRST MODEL TEST



HEUC EXPRES --- EXPRV510

EXPROO1 EXPRESS AJOB DYNAMO ALS9093
 EXPROO0 DYNAMO FORBRUG: 4 SEKUNDE REEL TID, 1.02 CPU-SEKUNDE OC 69 DISK I/O

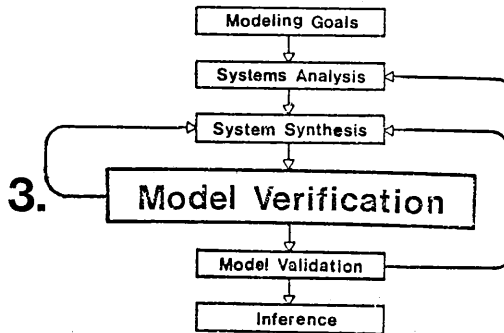
In some instances DYNAMO uses a letter behind a number to indicate the scaling on output. The letters have the following significance:

Letter	Multiply Printed Results by
A . . .	10^{-3}
B . . .	10^9
C . . .	10^{27}
D . . .	10^{33}
E . . .	10^{-6}
F . . .	10^{-9}
G . . .	10^{-12}
H . . .	10^{-15}
J . . .	10^{-18}
K . . . <	10^{-30}
L . . .	10^{-21}
M . . .	10^6
N . . .	10^{30}
P . . .	10^{24}
Q . . .	10^{15}
R . . .	10^{12}
S . . .	10^{21}
T . . .	10^3
U . . .	10^{-24}
V . . .	10^{18}
W . . .	10^{-27}
X . . .	1
Y . . .	10^{-30}
Z . . . >	10^{33}

2.2.8 Further features of DYNAMO

This section gives a reference to DYNAMO facilities not discussed in this note. The reference is made to [5] with the corresponding page number.

Information and material delays	[5, p. 29]
Random number normally distributed	[5, p. 30]
Pulse, ramp and sample functions	[5, p. 31]
Smoothing	[5, p. 32]
Table look-up with horizontal asymptotes	[5, p. 34]
A facility for user-definition of functions (Macros)	[5, p. 36]
Changing the seed of random number generator	[5, p. 38]
Scaling of printing	[5, p. 41]
A rerun facility allowing the user to change constant and tables without recompile ling the program	[5, p. 43]
Comparing plots from two runs	[5, p. 85]
Most error messages from DYNAMO are self-ex- planatory, error messages	[5, p. 44-60]



The verification stage is occupied with the determination whether the model was actually synthesized as intended in the system Synthesis stage, i.e. the purpose of the verification stage is to verify the internal logic of the model.

In our synthetization of the model we have chosen an integration scheme for solving the model equations. A first demand to the internal logic of a model must be that the integration scheme actually yields a solution reasonably close to the true solution of the model equations. In order to help the reader in passing this judgement, this section starts with an exemplified introduction to errors in numerical integration. Following this introduction a simulation package, COLTS, with input similar to DYNAMO but with better integration methods is introduced in section 3.2.

Another part of the verification is constituted by a determination of whether model responses are as intended by the modeller. This calls for experiments with the model in order to determine the model behavior.

Mihram [2] states

"One should note that verification tests are not conducted via comparisons of model responses with those of the actual modeled system; rather, comparisons between observed model responses and theoretically anticipated results are made in as many cases as possible ..."

Section 3.3 exemplifies the determination of model behavior.

The internal logic of a model formulated as coupled differential equations is severely challenged in cases where the value of a not well determined parameter ("constant") greatly affects the solution of the model. If this is the case, the solution is said to be highly sensitive to the parameter. Since it is the internal logic which is challenged, the verification stage should also include a sensitivity analysis. Sensitivity analysis is treated in section 3.4.

As previously stated, the verification stage is occupied with the validity of the system synthesis stage. A failure of verification thus indicates that the model must be re-synthesized, i.e. that we must return to the system synthesis stage in the modeling process.

3.1 INTRODUCTION TO ERRORS IN NUMERICAL INTEGRATION

This section exemplifies the different types of errors in numerical integration. The section is based on [6].

3.1.1 DYNAMO integration - Euler's method

The integration in DYNAMO is performed by Euler's method. The method is a first order method projecting the solution along the tangent, i.e.

$$Y_{n+1} = Y_n + h \cdot Y'(t_n, Y_n)$$

where Y_n is the solution at time t_n , Y_{n+1} at time t_{n+1} and h is the steplength. The method will have a tendency of overestimating the solution when the true solution is downward hollow, and of underestimating when upward hollow. See Fig. 11. Furthermore, the error is accumulative.

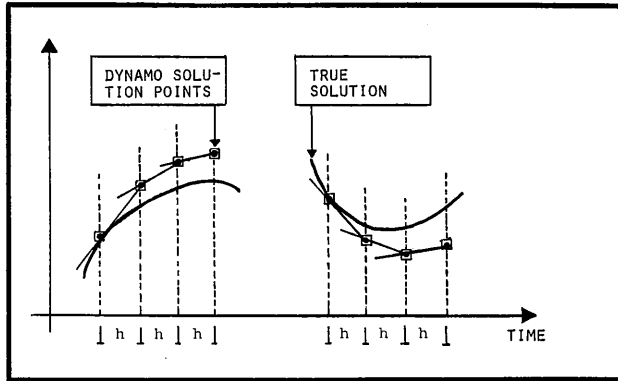


FIGURE 11: The effect of Eulerian projection along the tangent.

The simple integration method and its DYNAMO implementation almost invariably leads to erroneous results, especially when a long time-horizon is considered. The following sections exemplify some types of errors caused by the DYNAMO integration.

3.1.2 Phase Error

We now consider the simple problem depicted on Fig. 12. In mathematical notation the system may represent the following problem:

$$\begin{aligned}
 \frac{dy_1}{dt} &= y_2 & y_1 &= c_1 \sin(t) + c_2 \cos(t) \\
 \frac{dy_2}{dt} &= -y_1 & y_2 &= c_1 \cos(t) + c_2 \sin(t)
 \end{aligned}
 \tag{1}$$

where the initial conditions

$$\left. \begin{array}{l} Y1(0) = 0 \\ Y2(0) = 1 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} c_1 = 1 \\ c_2 = 0 \end{array} \right.$$

With support in Fig. 12 it is an easy task to formulate the problem in DYNAMO.

The following program emerges:

```
*      SIN/COS
NOTE
NOTE  EXAMPLE
NOTE
L      Y1.K=Y1.J+DT*RT1.JK
R      RT1.KL=Y2.K
NOTE
L      Y2.K=Y2.J+DT*RT2.JK
R      RT2.KL=-Y1.K
NOTE
N      Y1=0
N      Y2=1
NOTE
A      A1.K=SIN(TIME.K)
A      A2.K=COS(TIME.K)
NOTE
NOTE  RUN ADMINISTRATION
NOTE
A      PLTPER.K=.1-STEP(.1,6.283)+STEP(.1,125.66)
PLOT  Y1=1(-2,2)/Y2=2(-2,2)/A1=S(-2,2)/A2=C(-2,2)
SPEC  LENGTH=131.943/DT=.001
RUN   TEST
```

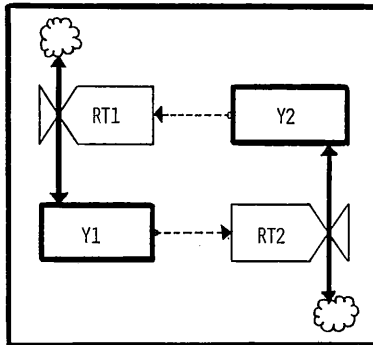


FIGURE 12: System Dynamics flow diagram representing a system of differential equations rendering the solution SIN and COS.

On the printerplot from DYNAMO the SIN and COS of the DYNAMO package were plotted along with the solution Y1,Y2 of (1), and the steplength of DYNAMO was lowered until there was no difference in the plot of the built-in functions and the solution. These test-runs were made on the first period of SIN and COS ($[0,2\pi]$). The resulting steplength was .001.

The program was now run over 21 periods with steplength .001 and the first and 21st periods were plotted. The resulting plot is shown on Fig. 13.

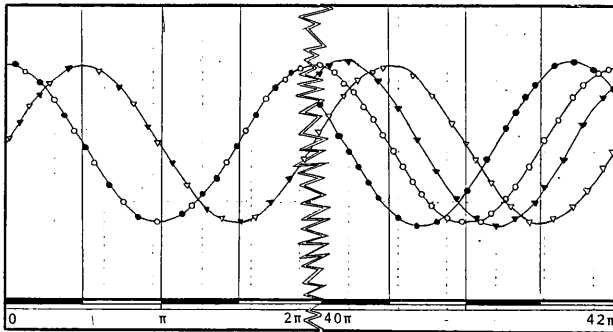


FIGURE 13: Plot of the SIN/COS test showing the first and the 21st period.

SIGNATURE: ∇ (\circ) represents the built-in SIN(COS) of DYNAMO

\blacktriangledown (\bullet) represents the solution Y1 (Y2) of (1) obtained by DYNAMO.

Due to the DYNAMO under/over estimation when the true solution is upward/downward hollow, the sign-change in the derivatives occur too early, resulting in a phase error. Fig. 13 shows a

considerable accumulation of phase error in the 21st period of the DYNAMO solution.

It has been said about DYNAMO, that it should be used to determine the general behavior of a system and not to render the exact solution of the system. Thus, a simple integration technique as Euler's method, should suffice.

Consider a simple system controlled by two periodical functions obtained as part of the system. If one of these control functions had a period half as long as the other and perhaps a different amplitude, the phase error would build up differently in the two. Simulating a period of time with DYNAMO you would not experience the behavior of your system, but of a system based on accumulated phase errors. In this connection, it is worth keeping in mind that most real life events hold a certain amount of periodicity.

3.1.3 Error Accumulation

As an example for our discussion of the problem in connection with accumulation of errors, we apply the prey/predator system:

In a closed reservation (no e- or immigration) lives a population of predators (N_2 individuals) and their prey (N_1 individuals). The birthrate of the prey is α_1 and the only deathcause for the prey is being devoured by the predators. Thus, the number of deaths occuring among the prey depends on the number of encounters prey/predator. The number of encounters is considered proportional to the number of individuals of the two species (Factor β_1). The predators have a deathrate α_2 . The more prey the predators devour, the better their living conditions and, thus, more predators are born (Factor β_2).

The prey/predator system may now be formulated as the following set of differential equations:

$$\frac{dN_1}{dt} = (\alpha_1 - \beta_1 N_2) N_1$$

(2)

$$\frac{dN_2}{dt} = (-\alpha_2 + \beta_2 N_1) N_2$$

(2): The Lotka-Volteira equations, where:

$N_1 = N_1(t)$ is the number of prey

$N_2 = N_2(t)$ is the number of predators

α_1 birthrate of prey

β_1 the rate at which prey is devoured

α_2 deathrate of predators

β_2 predator benefit rate

We now proceed to find a kinematic solution to (2).

$$(N_1 N_2 \neq 0)$$

$$= \left\{ \begin{array}{l} \frac{dN_1}{dt} = N_1 (\alpha_1 - \beta_1 N_2) \quad \left[= 0 \Rightarrow N_2 = \frac{\alpha_1}{\beta_1} \right] \\ \frac{dN_2}{dt} = N_2 (\beta_2 N_1 - \alpha_2) \quad \left[= 0 \Rightarrow N_1 = \frac{\alpha_2}{\beta_2} \right] \end{array} \right.$$

$$\frac{dN_1}{dN_2} = \frac{N_1 (\alpha_1 - \beta_1 N_2)}{N_2 (\beta_2 N_1 - \alpha_2)} = \frac{\alpha_1 N_1 - \beta_1 N_1 N_2}{\beta_2 N_1 N_2 - \alpha_2 N_2} \quad \rightarrow$$

$$\frac{dN_1}{dN_2} = \frac{\frac{\alpha_1}{N_2} - \beta_1}{\beta_2 - \frac{\alpha_2}{N_1}} \quad \rightarrow$$

$$(3) \quad (\beta_2 - \frac{\alpha_2}{N_1}) dN_1 = (\frac{\alpha_1}{N_2} - \beta_1) dN_2$$

Integrating (3) we obtain

$$(4) \quad \beta_2 N_1 - \alpha_2 \text{LOG}(N_1) = \alpha_1 \text{LOG}(N_2) - \beta_1 N_2 + c$$

into which we insert

$$N_1 = x + \frac{\alpha_2}{\beta_2} = x + p$$

$$N_2 = y + \frac{\alpha_1}{\beta_1} = y + q$$

Thus, obtaining

$$(5) \quad \boxed{\alpha_2 \text{LOG}(x+p) + \alpha_1 \text{LOG}(y+q) - \beta_1 y - \beta_2 x = c'}$$

By inserting different values for c' in (5) we obtain the set of phase-plots indicated on Fig. 14.

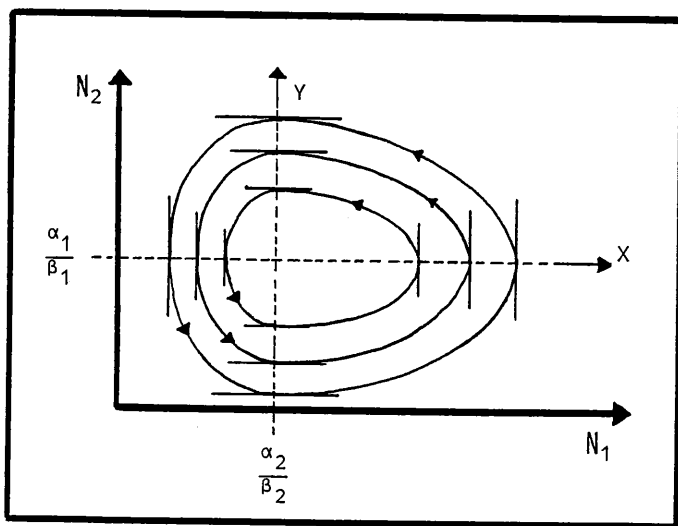


FIGURE 14: Phase-plot of the prey/predator system.

From our kinematic investigation of the system we may conclude that the solution $N_1(t), N_2(t)$ will be a set of periodical functions with constant amplitude.

We now turn to the problem of solving the system using DYNAMO. The first step is to make the System Dynamics flowchart on Fig. 15.

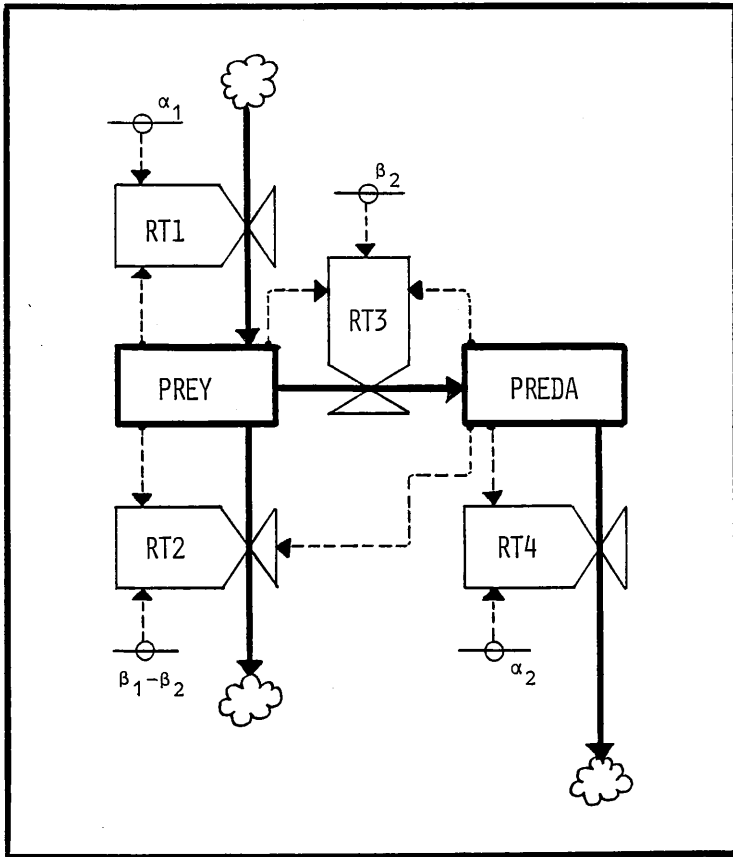


FIGURE 15: System Dynamics Flowchart representing the prey/predator system.

With support in Fig. 15 the following program may be written:

```
*      PREY/PREDATOR SYSTEM
NOTE
NOTE  EXAMPLE
NOTE
L      PREY.K=PREY.J+DT.J*(RT1.JK-RT2.JK-RT3.JK)
L      PREDA.K=PREDA.J+DT.J*(RT3.JK-RT4.JK)
NOTE
R      RT1.KL=ALFA1*PREY.K
R      RT2.KL=(BETA1-BETA2)*PREY.K*PREDA.K
R      RT3.KL=BETA2*PREY.K*PREDA.K
R      RT4.KL=ALFA2*PREDA.K
NOTE
C      ALFA1=2.5
C      ALFA2=1.0
C      BETA1=.02
C      BETA2=.01
NOTE
N      PREY=100
N      PREDA=124
NOTE
NOTE  RUN ADMINISTRATION
NOTE
A      DT.K=.005
A      PLTPER.K=.5-STEP(.5,17)+STEP(.5,383)
PLOT  PREY=*/PREDA=+
SPEC  LENGTH=400
RUN   TEST
```

Running the problem we obtain the solution plotted on Fig. 16.

Due to accumulated errors, the DYNAMO solution leaves the true solution and proceeds along a spiral in the phase-plane. This development is gradual, i.e. with no sudden jumps. Had our kinematic investigation not given us the knowledge that the solution to our problem has a closed path in the phase-plane, we would probably not have stated that the obtained solution is erroneous with respect to system behavior.

The example underlines the necessity of utter caution when deriving conclusions concerning system behavior from a simple integration method like Euler's. It is hard to determine whether your conclusions are based on the actual behavior of the system or on accumulated integration errors.

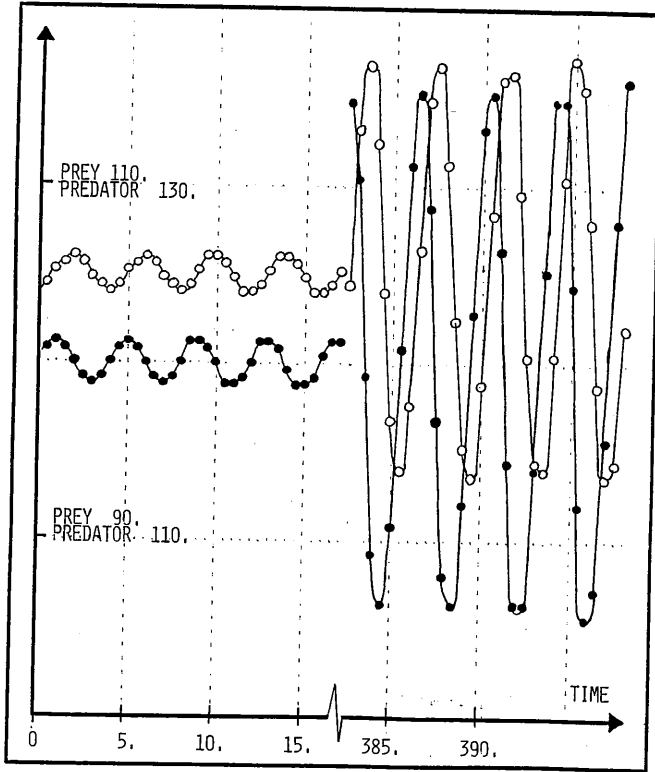


FIGURE 16: DYNAMO solution of the prey/predator system.

SIGNATURES: —●—●— Prey
—○—○— Predator

3.1.4 Rounding errors

In our discussion of rounding errors we apply the model of the isolated biological subsystem from the master example.

On Fig. 7. p. 30 we see that the material flows between N, A and ONS form a closed loop. We may now test whether this loop is kept closed throughout the integration by defining

$$\text{CHECK} = A + \text{ONS} + N$$

and plotting this control-variable along with the results. The resulting plot showing the 2nd year of the solution is shown on Fig.17.

It appears from Fig. 17 that DYNAMO is unable to keep the loop closed and, thus, is constantly creating new transients for the system, thus, making a static solution impossible.

From fig. 17 we notice that the decrease in CHECK seems linear with time, i.e. with the number of steps. Experiments with different steplengths were, therefore, carried out. The results appear on fig. 18 marked n-DYNAMO, where n gives the time for the curve. In connection with Fig. 18. it should be noted, that a constant CHECK value does not give any information on the quality of the integration. Only when CHECK is changing may we deduce that something is wrong.

The linear double logarithmic dependance on Fig. 18. indicates the trouble. A halving of the step-size doubles the CHECK deviation, i.e. the same error seems to occur at all solution points regardless of the stepsize meaning that we have a rounding error. To verify this hypothesis a double precision program was written in ALGOL simulating the rounding error by changing the integration results to single precision before forming the CHECK time derivative

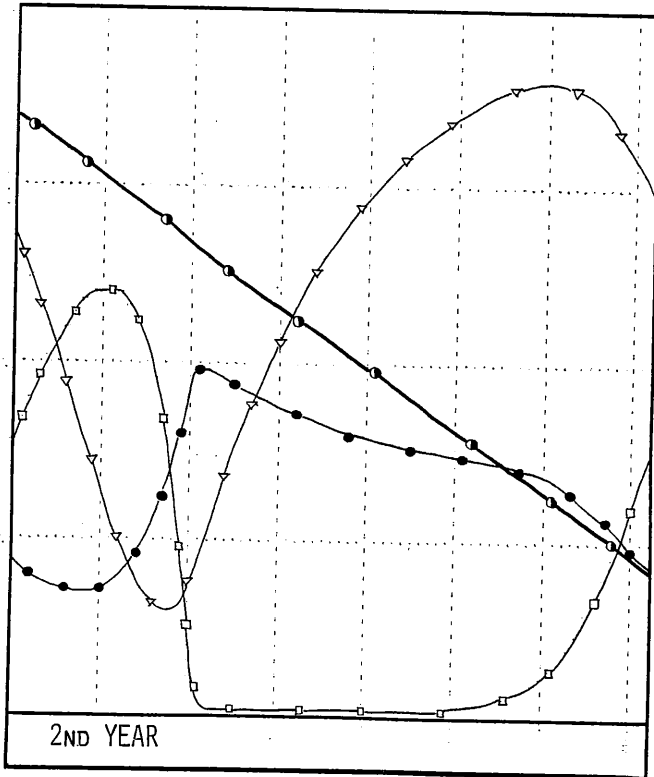


FIGURE 17: DYNAMO solution of the biological subsystem model:
The figure shows 2nd year of the solution.

SIGNATURES: \diamond — \diamond — y_{lo} : Lower bound of figure
 (y_{lo}, y_{hi}) y_{hi} : Upper bound
 \bullet — \bullet — Algal biomass
 $(0, 4)$
 \square — \square — Nutrients in waterphase
 $(0, 2)$ sediment
 ∇ — ∇ —
 $(12.9, 13.7)$ 5.CHECK
 \bullet — \bullet —
 $(75.48, 75.495)$

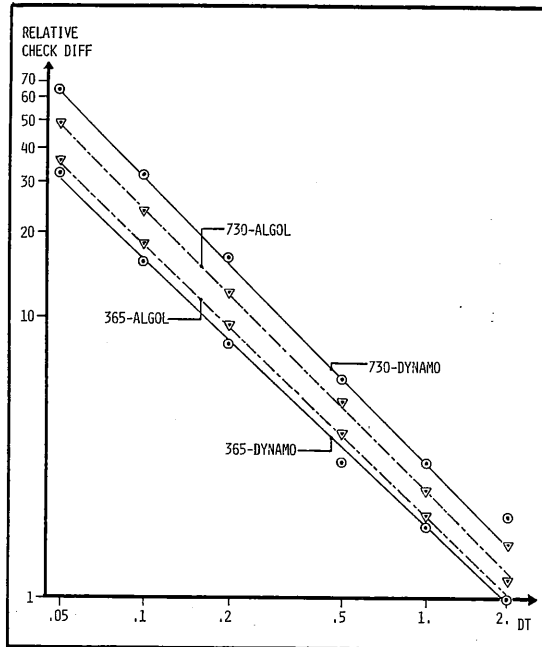


FIGURE 18: CHECK decrease relative ($5000 \cdot \text{CHECK}$) versus step-length. n-DYNAMO indicates the value obtained by DYNAMO at time n, n-ALGOL the same value obtained by an ALGOL simulation of rounding errors.

$$(6) \quad \text{CHECK}(t) = A(t) + \text{ONS}(t) + N(t)$$

The double precision calculated CHECK was constant. The accumulated rounding error derived from (6) is shown on Fig. 18. marked n-ALGOL, where n gives the time for the curve.

Generally, methods like Euler's have error contributions depending on the stepsize as indicated on Fig. 19.

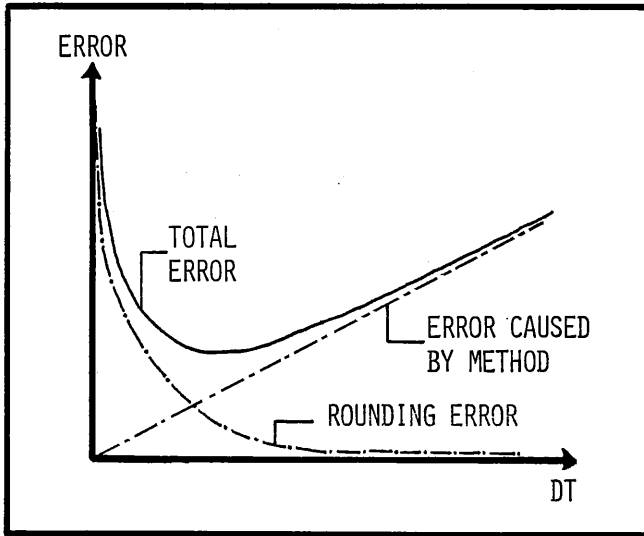


FIGURE 19: The error-contribution for Runge-Kutta methods depending on stepsize.

3.15. Stability problems

For our discussion of stability problems we apply Van Der Pool's

oscillator*) represented by

$$(7) \quad \frac{d^2x}{dt^2} - \varepsilon \cdot (1-x^2) \frac{dx}{dt} + x = 0$$

For increasing ε the oscillator approaches a square pulsator with amplitude 2 and period increasing with ε . We may follow this approach to the pulsator by defining

$$(8) \quad \varepsilon(t) = \varepsilon_0 \cdot 10^{t/\text{PER}}, \quad \text{PER} = \text{constant}$$

i.e. by letting ε increase with time. Defining

$$y_1 = x \text{ and } y_2 = \frac{dx}{dt}$$

we may reformulate the combination of (7) and (8) to a set of first order differential equations:

$$(9) \quad \begin{aligned} \frac{dY_1}{dt} &= Y_2 \\ \frac{dY_2}{dt} &= \varepsilon(t) \cdot (1-Y_1^2) \cdot Y_2 - Y_1 \end{aligned}$$

We can now solve Van Der Pool's oscillator using DYNAMO. With support in the System Dynamics flow-chart of Fig. 20. the DYNAMO-program listed is easily written.

Fig. 21 shows part of the solution of (9) obtained by the DYNAMO-program.

After some time of simulation we rather unexpectedly encounter an exponent overflow in y_2 . Fig. 22 gives the solution obtained by DYNAMO right before the overflow is encountered.

*) This example was suggested by P.G. Thomsen of the Institute for Numerical Analysis, the Technical University of Denmark.

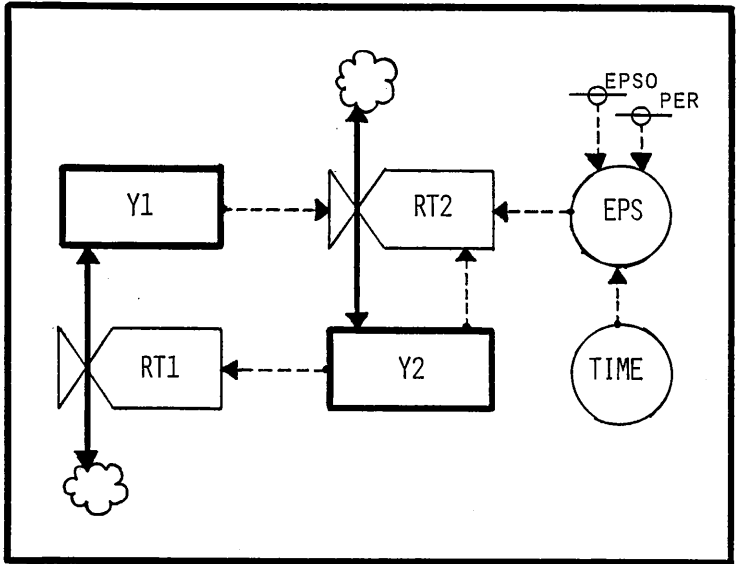


FIGURE 20: System Dynamics flowchart representing Van Der Pool's oscillator.

```

* VAN DER POOL'S OSCILLATOR
NOTE
NOTE EXAMPLE
NOTE
L Y1.K=Y1.J+DT*RT1.JK
R RT1.KL=Y2.K
NOTE
L Y2.K=Y2.J+DT*RT2.JK
R RT2.KL=EPS.K*(1-Y1.K*Y1.K)*Y2.K-Y1.K
NOTE
A EPS.K=EPS0*EXP((TIME.K/PER)*LOGN(10))
NOTE
N Y1=2
N Y2=0
NOTE
C EPS0=.1
C PER=30.
NOTE
NOTE RUN ADMINISTRATION
NOTE
PLOT Y1=1/Y2=2
SPEC DT=.005/PLTPER=.5/LENGTH=150
RUN TEST
    
```

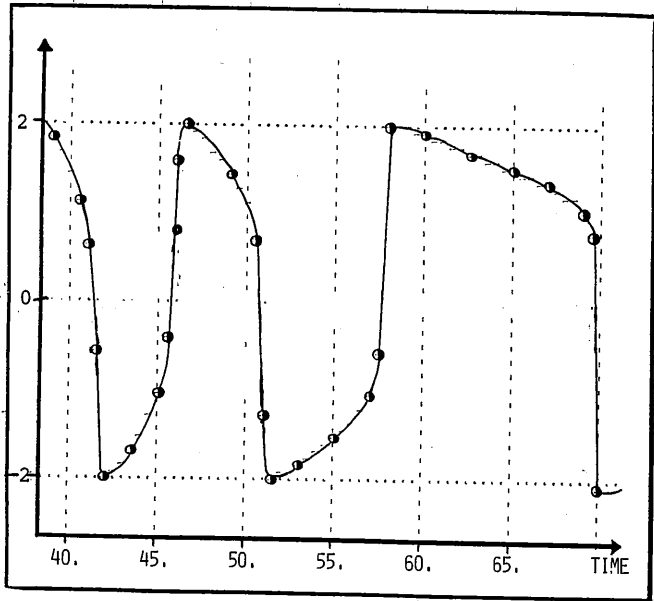


FIGURE 21: DYNAMO solution of Van Der Pool's oscillator.

TIME	Y1	Y2	EPS
102.280	-1.65	3.2	256.61
102.285	-1.63	-3.9	256.70
102.290	-1.65	4.4	256.80
102.295	-1.63	-5.4	256.90
102.300	-1.66	6.1	257.00
102.305	-1.63	-7.6	257.10
102.310	-1.66	8.5	257.20
102.315	-1.62	-10.8	257.30
102.320	-1.68	11.9	257.39
102.325	-1.62	-15.8	257.49
102.330	-1.70	17.1	257.59
102.335	-1.61	-24.1	257.69
102.340	-1.73	25.4	257.79
102.345	-1.60	-40.0	257.89
102.350	-1.80	41.2	257.99
102.355	-1.60	-78.6	258.08
102.360	-1.99	79.1	258.18
102.365	-1.60	$-2.24 \cdot 10^2$	258.28
102.370	-2.71	$2.22 \cdot 10^2$	258.38
102.375	-1.60	$-1.61 \cdot 10^3$	258.48
102.380	-9.65	$1.63 \cdot 10^3$	258.58
102.385	-1.51	$-1.92 \cdot 10^2$	258.68
102.390	$-9.61 \cdot 10^2$	$1.29 \cdot 10^2$	258.78
102.395	$-3.15 \cdot 10^2$	$-1.54 \cdot 10^9$	258.88
102.400	$-7.72 \cdot 10^6$	$1.98 \cdot 10^{15}$	258.98
102.405	$9.90 \cdot 10^{12}$	$-1.53 \cdot 10^{33}$	259.07
102.410	EXPONENT OVERFLOW		

FIGURE 22: DYNAMO solution of the Van Der Pool oscillator prior to exponent overflow.

Fig. 22 shows that the integration error has suddenly exploded.

This indicates that the numerical method has become unstable. The remedy for this instability is to use a smaller stepsize, resulting in an increase in computational cost.

Some systems of differential equations, including (9), has a solution in which the different components can vary in time constants. Some components will decay or increase very slowly while others may only influence the solution in a short transient phase. The slowly varying components form what is sometimes called "the set of smooth solutions". The fast ones are referred to as "stiff" components.

The increase in ϵ increases the stiffness of the system finally resulting in the error explosion demonstrated in fig. 22.

When the transient phase is over, the smooth components carry all the information about the solution while the stiff components have died out. The goal of our integration is to find the smooth solution. However, it can be shown that the stiff components control the error of methods like Euler's and Runge-Kutta, and that it is because of their presence in the solution that the methods become unstable.

Special integration methods that are able to cope with stiff systems of differential equations have been developed. The Gear predictor-corrector method [7,8] and the semi-implicit Runge-Kutta methods [22] are of this class.

3.2 COLTS (Continuous Long-Term Simulation)

The COLTS package has been constructed with the primary aim of offering more sophisticated methods of integration than the one used in DYNAMO and, at the same time, preserving the simple input-language from DYNAMO. COLTS may be viewed as a continuous version of the basically discrete DYNAMO compiler.

The DYNAMO input-language is a subset of the COLTS-language. This means that a DYNAMO program may be run under the COLTS package. However, the COLTS language has a few additional features. In this section, which is adapted from [21], these additional features are introduced.

In DYNAMO the three timeindices J, K and L are meaningful since a model formulated in DYNAMO is interpreted as a set of difference equations. A lowering of the distance DT between J, K and L is then experienced as a transition from difference to differential equations.

In COLTS, the timeindices are only used as a means of notation and are, therefore, not meaningful. In COLTS, the level-equation

$$L \quad L.K = L.J + DT*RT.JK$$

is interpreted as a differential equation:

$$\frac{dL}{dt} = RT$$

This means, that a model formulated in COLTS is a set of coupled first order differential equations. As a consequence of this interpretation it is possible to solve the model using more sophisticated integration methods than the tangential projection applied in DYNAMO. However, this interpretation also makes it impossible to solve DYNAMO-models that are truly difference equation models, i.e. models which in their structure make logical use of the displacement of the timeindices.

COLTS is a PL/I based precompiler to FORTRAN. In addition to the DYNAMO facilities, COLTS version 1.0 offers

- a) Four additional integration routines
 - 1) Runge-Kutta Fehlberg 4th order
 - 2) Bulirsch - Stoer
 - 3) Gear Predictor-Corrector
 - 4) Semi Implicit Runge-Kutta 2nd order

Furthermore, the user may supply his/her own integration routines.

- b) The definition and use of FORTRAN FUNCTIONS in a COLTS program. The functions may also be supplied as load modules.

The use of functions opens the possibility of reading table-values from data cards, easy programming of decisions, terminating the simulation at preset conditions a.s.o.

- c) PHASE plot, i.e. plots of solution versus solution.

3.2.1 Specifying an integration method

It is outside the aim of these notes to introduce different methods in numerical analysis. The integration methods are handled by way of reference. For an introduction to numerical analysis, see [23,24].

In COLTS the name of the integration method is given on a special SIM-card. The card has the following form

SIM	NAME	=	$\left\{ \begin{array}{l} \text{EULR} \\ \text{RKF4} \\ \text{BUST} \\ \text{GEAR} \\ \text{SIRK} \end{array} \right\}$	/	INITDT	=	number/
	MAXDT	=	number/		MINDT	=	number/
	RELERR	=	number/		ABSERR	=	number/
	FLAG	=	number/				
	TIMOUT	=	$\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\}$				

where

NAME= Default: NAME=RKF4, see later for details

INITDT Specifies the initial stepsize. During the simulation INITDT contains the actual value of the stepsize.
Default: LENGTH/100

MAXDT The maximum stepsize to be used in the integration.
Default: LENGTH/50

MINDT Minimal stepsize to be used.
Default: INITDT/1000

RELERR The relative local error tolerance to be used. The integration methods have an automatic stepsize control. The stepsize will be set to meet certain error tolerances $f(Y, RELERR, ABSERR)$. f differs slightly from method to method, see under the specific routine for details.
Default: 10^{-3}

ABSERR The absolute local error tolerance.
Default: 10^{-5}

TIMOUT TIMOUT=1 will give output of the simulation time after each step in the integration.
Default:0

FLAG will indicate the version of the integration routine to be used. See under the specific routine for details.
Default: FLAG=0

NAME=EULR

Method Eulers method.
(Tangential projection)

Notice

All other specifications on the sim-card are without meaning with this method. The step-size is DT which may be given on the SPEC-card or as an auxilliary.

Recommended_use

The use of NAME=EULR is not recommended for programs containing differential equations (i.e. LEVELS) since the method frequently leads to erroneous results as exemplified in section 3.1. However, the method will find its use when the program is purely computational and not differential equation solving.

NAME=RKF4

Method

Runge-Kutta Fehlberg 4th order, for details see [25,26]

Error_evaluation

LE(I): Local error for the i'th level Y(I)
LE(I) < ABS(Y(I)) * RELERR + ABSERR

$$\text{FLAG} = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^1$$

Normal operation

$$\text{FLAG} = \begin{Bmatrix} 2 \\ 3 \\ 4 \end{Bmatrix}^1$$

Special mode for effective passage of discontinuities where the position of a discontinuity is found by binary search.

Recommended_use

RKF4 is the default routine in COLTS. This is due to the fact that the code can handle most non-stiff problems without difficulty. If there is a chance that your system is stiff, COLTS will issue an error and recommend the use of GEAR with FLAG=2 or SIRK.

NAME=BUST

Method Bulirsch-Stoer, 6th order for details see [26]

Error_evaluation $LE(I) < ABS(Y(I)) * RELEERR + ABSERR$

FLAG=0 is the only option for this method

Recommended_use Bust will solve the same class of problems as RKF4. However, BUST can obtain a larger MAXDT than RKF4 meaning that BUST can be of interest if you are primarily concerned with the results at the end of the simulation

NAME=GEAR

Method Gear Predictor-Corrector, for details see [7,8]

Error_evaluation $LE < MAX(ABS(Y(I)) * RELEERR + ABSERR), I \in [1, N]$

FLAG=0 Adams predictor-corrector method

FLAG = $\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}^1$ Variable order method with numerical differencing for the Jacobian.

Recommended_use With FLAG=0 the method solves the same class of problems as RKF4. With FLAG=1 or 2 the method is suitable for stiff systems, especially if a low error tolerance is desired. For larger error tolerances ($RELEERR > 10^{-3}$) SIRK may be more economical for stiff systems.

NAME=SIRK

Method Semi Implicit Runge-Kutta 2nd order, for details, see [22].

Error_evaluation $LE < \text{MAX}(\text{ABS}(Y(I)) * \text{RELERR} + \text{ABSERR}), I \in [1, N]$

FLAG=0 is the only option for the method

Recommended_use SIRK is especially suited for stiff systems where a not too low error tolerance is desired. If a small error-tolerance ($\text{RELERR} < 10^{-3}$) is desired, GEAR with FLAG=2 may be more economical.

It should be noted that the various initial parameters for the integration methods (INITDT, MINDT a.s.o.) all have default values which are suitable for most problems. This means that the user in most cases only need to specify the name of the desired integration method.

Example:

 SIM NAME=RKF4

3.2.2 PHASE-plots

In COLTS it is possible to request a plot of solution versus solution (a phase-plot). Automatic or selfchosen scaling can be applied.

Example:

In the prey/predator system (see DYNAMO-program on p. 58) we wish to plot prey versus predator, i.e. get one of the curves on fig. 14, p. 56, with lower bound for prey on 65, upper bound 145, lower bound predator 90, upper bound 160.

 PHASE PREY(65,145)/PREDA(90,160)

The interval between points on the plot is contained in the auxiliary

PHSPER (PHaSePERiode)

PHSPER may also be given on the SPEC card.

Programming example:

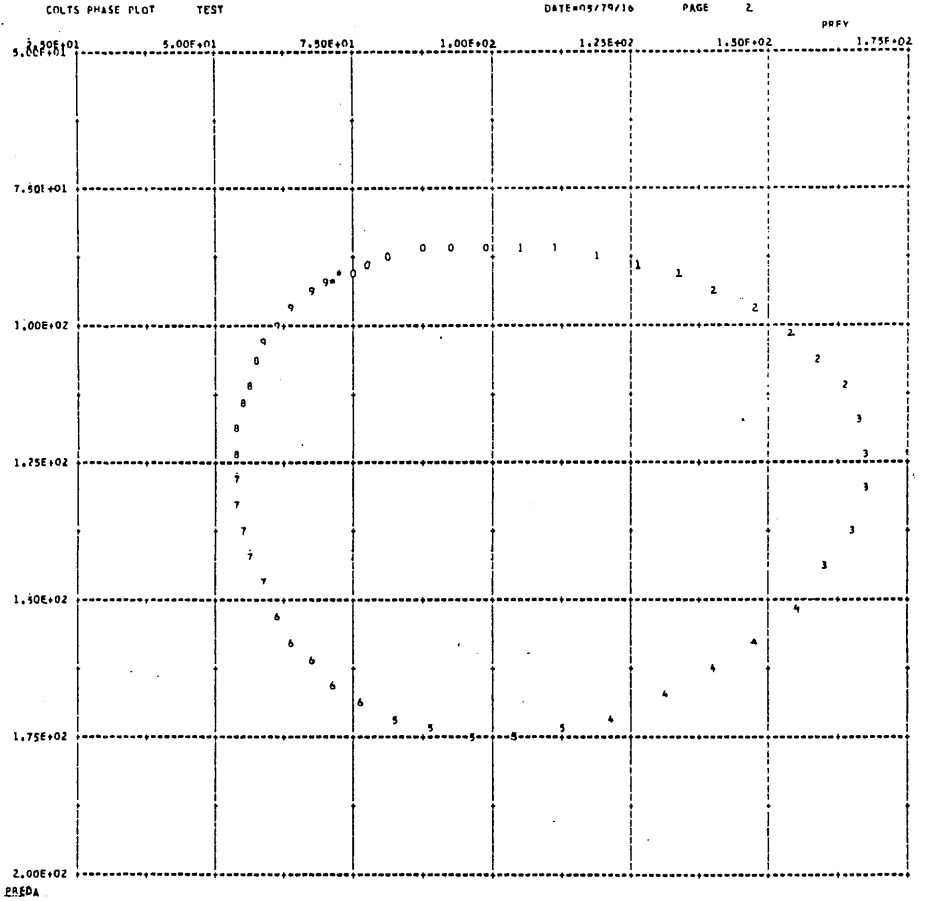
We wish to run the prey/predator program p. 58 under COLTS using Runge-Kutta Fehlberg 4th order and obtain a phase plot of prey versus predator.

```
// EXEC COLTS
//CSYSIN DD *

*      PREY PREDATOR SYSTEM
NOTE
NOTE  EXAMPLE:  COLTS PHASE-PLOT
NOTE
L      PREY.K=PREY.J+DT*(RT1.JK-RT2.JK-RT3.JK)
L      PREDA.K=PREDA.J+DT*(RT3.JK-RT4.JK)
R      RT1.KL=ALFA1*PREY.K
R      RT2.KL=(BETA1-BETA2)*PREY.K*PREDA.K
R      RT3.KL=BETA2*PREY.K*PREDA.K
R      RT4.KL=ALFA2*PREDA.K
C      ALFA1=2.5
C      ALFA2=1.0
C      BETA1=.02
C      BETA2=.01
N      PREY=75
N      PREDA=90
NOTE
NOTE  RUN ADMINISTRATION
NOTE
A      PHSPER.K=.001
PHASE  PREY(65,145)/PREDA(90,160)
SIM    NAME=RKF4
SPEC   LENGTH=4
RUN    TEST
```

The resulting output from PHASE is shown on the next page.

The syntax of the JCL for COLTS is given in appendix B along with some examples.



Output from PHASE

Notice that the phase-curve has been split in 10 equally long time intervals indicated by the digits 0-9. Star indicates 2 or more digits in the same position.

3.2.3 Further Features of COLTS

In addition to the normal DYNAMO II facilities (see [5]), COLTS has the following features:

COLTS supports involution
(x^5 is written as x^{**5})

Use of builtin FORTRAN
FUNCTIONS [21,p.30]

Use of user defined FORTRAN
FUNCTIONS [21,p.27]

User supplied integration
routines [21,p.41]

Dimensioning of tables
[21,p.16]

3.3 MODEL BEHAVIOR

The determination of model behavior is an exercise in deduction of cause-effect relationships. Having observed a phenomena in behavior, you consider which factors immediately influence the part of the model in which the phenomena has been observed. Visualizing the changes in these factors you may locate the factor immediately causing the phenomena. Then considering which factors influence the immediate cause, you may proceed in the same way until full understanding of the phenomena is achieved.

We exemplify the approach with the model of the biological subsystem. On Fig. 23 we notice that there is a spring maximum in the algal biomass. We want to determine what in the model causes this spring maximum. Turning to Fig. 7 we see, that the processes immediately influencing the algal biomass are sedimentation and primary production. The values of these two processes are plotted on Fig. 23. The sedimentation is merely proportional to the algal biomass and will, therefore, not be able to cause the maximum. We may then conclude that the spring maximum is caused by the fall in primary production.

The factors governing the primary production are algal biomass, external compulsion and nutrient. We isolate the effect of these three factors and make a plot of their values, see Fig. 23. The inhibition from the external compulsion is decreasing (COMP is increasing) at the time of the maximum so external compulsion may be disregarded as cause of the phenomena. Likewise, the algal biomass is increasing leaving us with the nutrients. On Fig. 23 we see that the lack of nutrients inhibits the primary production.

We may now conclude:

In the model, the spring maximum in the algae is caused by a shortage of nutrients.

Asking our biologist we get the answer, that the spring maximum due to nutrient shortage can be expected to occur in nature. Thus, we have established a comparison between an observed model response and a theoretically anticipated result for the model of the biological subsystem.

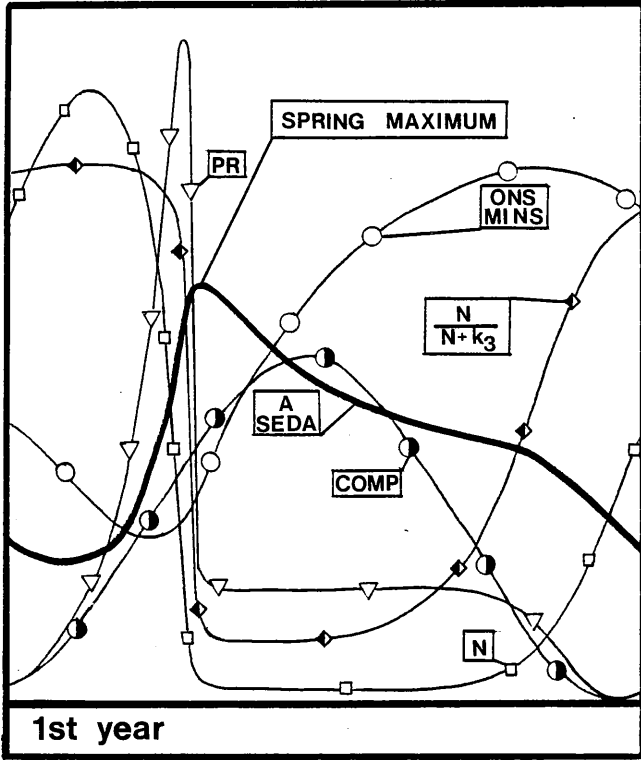


FIGURE 23: Plot of some of the variables in the model of the biological subsystem.

SIGNATURES:



(lo, up)

lo: Lower bound of figure

up: Upper bound of figure.

A ———
(0, 4)

SEDA ———
(0, .04)

N —□—□—
(0, 2)

$\frac{N}{N+K_3}$ —◆—◆—
(0, 1.2)

ONS —○—○—
(12, 14)

MINS —○—○—
(.012, .014)

COMP —●—●—
(0, 2)

PR —▽—▽—
(0, 80)

3. Sensitivity analysis

Sensitivity analysis is occupied with the determination of the influence of changes in one part of the system of differential equations on another. In this section we treat the changes in solution due to changes in parameter vector (the vector formed by the constants of the model), i.e. the sensitivity of solution to parameters. This is of interest in the verification stage since a high sensitivity to a not well determined parameter is crucial to the internal logic of a model.

The most commonly used approach to sensitivity to parameter-changes is that of changing one parameter at a time, run a simulation and then observe the resulting change in solution. However, as it will be shown in this section, parameter changes have an interaction effect, i.e. the changing of two parameters simultaneously does not lead to a change in solution equating the sum of changes from changing the same two parameters one at a time. In order to describe these interactions with the parameter-change approach all possible parameter combinations should be simulated. In the case of n parameters this would lead to 2^{n-1} simulations *) which is a prohibitively large number for any realistic model. Consequently, this note takes a semi-analytical approach to sensitivity instead of presenting the parameter-change approach. The description is based on [9].

The method operates through an analytical formulation of the first order time derivatives of the sensitivity. Using a numerical method which will estimate the higher order derivatives, the differential equations are solved. A sensitivity analysis using the parameter-change approach is then undertaken on the differential equations of the sensitivity yielding the parameter interactions as result. The necessary number of simulations thus equals the number of parameters.

3.4.1 Analytical Approach

Consider the system

*) $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n} = (1+1)^{n-1}$

$$(10) \quad \dot{\underline{Y}} = f(\underline{Y}, \underline{P}, t), \quad \underline{Y}_0 = \underline{C}_1$$

where

\underline{Y} is a statevector with n elements

\underline{P} is a parametervector with m elements

\underline{C}_1 is a constantvector containing initial conditions

t is time and $\dot{\underline{Y}} = \frac{d\underline{Y}}{dt}$

In an analytical solution of (10) \underline{Y} would be a function of \underline{P}, t and initial conditions. The solution may be developed using Taylor's formula

$$(11) \quad \underline{Y}_{n+1} = \underline{Y}_n + \Delta t \cdot f + \frac{\Delta t^2}{2!} f^{(1)} + \dots + \frac{\Delta t^{n-1}}{(n-1)!} f^{(n-2)} + \underline{R}(n)$$

where

\underline{Y}_n is the solution at time T

\underline{Y}_{n+1} the solution at time $T + \Delta t$

$f^{(1)} = \frac{df}{dt} + f \frac{df}{dy}$ etc. and

$$\underline{R}(n) = \frac{\Delta t^n}{n!} f^{(n-1)}(\underline{Y} + \Theta_1 \cdot \Delta \underline{Y}, \underline{P}, T + \Theta_2 \cdot \Delta t),$$

$\Theta_i \in [0, 1]$ Lagrange's remainder term

Defining the sensitivity as the change in solution due to change in the parametervector, sensitivity may be expressed as

$$(12) \quad \underline{S} = \frac{\partial \underline{Y}}{\partial \underline{P}}$$

where \underline{S} is the $n \times m$ sensitivity matrix.

An expression for sensitivity may be derived through a differentiation of (11)

$$\begin{aligned}
 (13) \quad & \frac{\partial \underline{Y}_{n+1}}{\partial \underline{P}} \cdot \frac{\partial \underline{P}}{\partial \underline{P}} + \frac{\partial \underline{Y}_{n+1}}{\partial t} \cdot \frac{\partial t}{\partial \underline{P}} = \frac{\partial \underline{Y}_n}{\partial \underline{P}} \cdot \frac{\partial \underline{P}}{\partial \underline{P}} + \frac{\partial \underline{Y}_n}{\partial t} \cdot \frac{\partial t}{\partial \underline{P}} \\
 & + \Delta t \left[\frac{\partial f}{\partial \underline{Y}} \cdot \frac{\partial \underline{Y}}{\partial \underline{P}} + \frac{\partial f}{\partial \underline{P}} \cdot \frac{\partial \underline{P}}{\partial \underline{P}} + \frac{\partial f \cdot \partial t}{\partial t \partial \underline{P}} \right] \\
 & + \frac{\Delta t^2}{2!} \left[\frac{\partial f^{(1)}}{\partial \underline{Y}} \cdot \frac{\partial \underline{Y}}{\partial \underline{P}} + \frac{\partial f^{(1)}}{\partial \underline{P}} \cdot \frac{\partial \underline{P}}{\partial \underline{P}} + \frac{\partial f^{(1)} \cdot \partial t}{\partial t \partial \underline{P}} \right] \\
 & + \vdots \\
 & + \vdots \\
 & + \frac{\Delta t^{n-1}}{(n-1)!} \left[\frac{\partial f^{(n-2)}}{\partial \underline{Y}} \cdot \frac{\partial \underline{Y}}{\partial \underline{P}} + \frac{\partial f^{(n-2)}}{\partial \underline{P}} \cdot \frac{\partial \underline{P}}{\partial \underline{P}} + \frac{\partial f^{(n-2)} \cdot \partial t}{\partial t \partial \underline{P}} \right] \\
 & + \frac{\partial \underline{R}(n)}{\partial \underline{P}}
 \end{aligned}$$

Using the substitution (12) and assuming \underline{P} to be a constant vector ($\Rightarrow \frac{\partial t}{\partial \underline{P}} = \underline{0}$) with $\frac{\partial \underline{P}}{\partial \underline{P}} = \underline{1}$, (13) may be reshaped as

$$\begin{aligned}
 (14) \quad \underline{S}_{n+1} &= \underline{S}_n + \Delta t \left[\frac{\partial f}{\partial \underline{Y}} \cdot \underline{S}_n + \frac{\partial f}{\partial \underline{P}} \right] \\
 &+ \frac{\Delta t^2}{2!} \left[\frac{\partial f^{(1)}}{\partial \underline{Y}} \cdot \underline{S}_n + \frac{\partial f^{(1)}}{\partial \underline{P}} \right] \\
 &+ \vdots \\
 &+ \vdots \\
 &+ \frac{\Delta t^{n-1}}{(n-1)!} \left[\frac{\partial f^{(n-2)}}{\partial \underline{Y}} \cdot \underline{S}_n + \frac{\partial f^{(n-2)}}{\partial \underline{P}} \right] \\
 &+ \frac{\partial \underline{R}(n)}{\partial \underline{P}}
 \end{aligned}$$

Since $\underline{Y}_0 = \underline{C}_1$ we have the boundary conditions

$$(15) \quad \underline{\underline{S}}_0 = \frac{\partial \underline{Y}_0}{\partial \underline{P}} = \underline{\underline{0}}$$

The expression (14) may be regarded as a Taylor development of the sensitivity. Thus,

$$(16) \quad \underline{\underline{\dot{S}}} = \frac{\partial f}{\partial \underline{Y}} \underline{\underline{S}} + \frac{\partial f}{\partial \underline{P}}$$

(16) may be viewed as a system of differential equations for the sensitivity. Using the initial conditions (15) the system may be solved yielding the local sensitivity, i.e. the sensitivity to small parameter-changes. The sensitivity changes, however, when we apply large parameter changes, i.e. the sensitivity is sensitive to parameter-changes.

In order to get an impression of the analytical implications of the sensitivity of the sensitivitiy, we take the simple approach of differentiating (14) leaving out all contributions of higher order than one. Since a matrix notation is no longer applicable, we reshape (14) to its element form omitting higher order derivatives. Hereby

$$(17) \quad S_{rs}^{n+1} = S_{rs}^n + \Delta t \cdot \frac{\partial f_r}{\partial Y_1} S_{1s}^n + \Delta t \frac{\partial f_r}{\partial Y_2} S_{2s}^n + \dots \\ + \Delta t \frac{\partial f_r}{\partial Y_m} S_{ms}^n + \Delta t \frac{\partial f_r}{\partial P_s}$$

differentiating (17) and reshaping the result we get

(18)

$$\begin{aligned}
 \frac{\partial S_{rs}^{n+1}}{\partial P_t} &= \frac{\partial S_{rs}^n}{\partial P_t} \\
 &+ \Delta t \left[\left(\frac{\partial f_r}{\partial Y_1} \cdot S_{1t}^n + \frac{\partial f_r}{\partial Y_2} S_{2t}^n + \dots + \frac{\partial f_r}{\partial Y_n} S_{nt}^n \right) \cdot S_{1s}^n + \frac{\partial f_r}{\partial Y_1} \frac{\partial S_{1s}^n}{\partial P_t} \right] \\
 &+ \Delta t \left[\left(\frac{\partial f_r}{\partial Y_1} \cdot S_{1t}^n + \frac{\partial f_r}{\partial Y_2} S_{2t}^n + \dots + \frac{\partial f_r}{\partial Y_n} S_{nt}^n \right) \cdot S_{2s}^n + \frac{\partial f_r}{\partial Y_2} \frac{\partial S_{2s}^n}{\partial P_t} \right] \\
 &+ \dots \\
 &+ \Delta t \left[\left(\frac{\partial f_r}{\partial Y_1^n} \cdot S_{1t}^n + \frac{\partial f_r}{\partial Y_2^n} S_{2t}^n + \dots + \frac{\partial f_r}{\partial Y_n^n} S_{nt}^n \right) \cdot S_{ns}^n + \frac{\partial f_r}{\partial Y_n} \frac{\partial S_{ns}^n}{\partial P_t} \right] \\
 &+ \Delta t \left[\frac{\partial f_r}{\partial Y_1^s} \cdot S_{1t}^n + \dots + \frac{\partial f_r}{\partial Y_n^s} S_{nt}^n \right]
 \end{aligned}$$

Even using a symbol manipulating compiler as FORMAC (18) would hardly hold any practical value for systems of realistic size.

This leads to the approach taken in this note, namely a sensitivity analysis of (16) through parameter-changes in order to establish the parameter interactions of the system.

3.4.2 Master Example, sensitivity

We decide to carry out a sensitivity analysis on the isolated biological subsystem. For the sake of simplicity we omit the constants K_5 and K_6 .

Our first task is to reshape the model equations to their analytical form parallel to that of (10):

$$(19) \quad \begin{aligned} \dot{N} &= K_1 * \text{ONS} - K_2 * A * \frac{N}{N+K_3} * \text{COMP} \\ \dot{A} &= K_2 * A * \frac{N}{N+K_3} * \text{COMP} - K_4 * A \\ \dot{\text{ONS}} &= K_4 * A - K_1 * \text{ONS} \end{aligned}$$

where

$$\text{COMP} = .5 * \left[1 + \text{SIN}\left(\frac{2\pi}{365} * (t-80)\right) \right]$$

Relating (19) to (10) we have

$$(20) \quad \begin{aligned} \underline{Y}_0 &= (1.4, .9, 12.8) \\ \underline{Y} &= (Y_1, Y_2, Y_3) = (N, A, \text{ONS}) \\ \underline{P} &= (P_1, P_2, P_3, P_4) = (K_1, K_2, K_3, K_4); \underline{P}_0 = (.001, .07, .12, .01) \end{aligned}$$

In order to use (16) we need the two matrices $\frac{\partial \dot{\underline{Y}}}{\partial \underline{Y}}$ and $\frac{\partial \dot{\underline{Y}}}{\partial \underline{P}}$

Substituting (20) into (19) and differentiating we get:

Non-zero elements of $\frac{\partial \dot{\underline{Y}}}{\partial \underline{Y}}$

$$(21) \quad \begin{aligned} \frac{\partial \dot{Y}_1}{\partial Y_1} &= -K_2 * Y_2 * \frac{K_3}{(Y_1+K_3)^2} * \text{COMP} \\ \frac{\partial \dot{Y}_1}{\partial Y_2} &= -K_2 * \frac{Y_1}{Y_1+K_3} * \text{COMP} \\ \frac{\partial \dot{Y}_1}{\partial Y_3} &= K_1 \end{aligned}$$

continued

$$(21) \quad \frac{\partial \dot{Y}_2}{\partial Y_1} = K_2 * Y_2 * \frac{K_3}{(K_3 + Y_1)^2} * \text{COMP}$$

$$\frac{\partial \dot{Y}_2}{\partial Y_2} = K_2 * \frac{Y_1}{Y_1 + K_3} * \text{COMP} - K_4$$

$$\frac{\partial \dot{Y}_3}{\partial Y_2} = K_4$$

$$\frac{\partial \dot{Y}_3}{\partial Y_3} = -K_1$$

and the non-zero elements of

$$\frac{\partial \dot{Y}}{\partial P}$$

$$(22) \quad \frac{\partial \dot{Y}_1}{\partial P_1} = Y_3$$

$$\frac{\partial \dot{Y}_1}{\partial P_2} = -Y_2 * \frac{Y_1}{Y_1 + K_3} * \text{COMP}$$

$$\frac{\partial \dot{Y}_1}{\partial P_3} = K_2 * Y_2 * \frac{Y_1}{(Y_1 + K_3)^2} * \text{COMP}$$

$$\frac{\partial \dot{Y}_2}{\partial P_2} = Y_2 * \frac{Y_1}{Y_1 + K_3} * \text{COMP}$$

$$\frac{\partial \dot{Y}_2}{\partial P_3} = -K_2 * Y_2 * \frac{Y_1}{(Y_1 + K_3)^2} * \text{COMP}$$

$$\frac{\partial \dot{Y}_2}{\partial P_4} = -Y_2$$

continued

$$(22) \quad \frac{\partial \dot{Y}_3}{\partial P_1} = - Y_3$$

$$\frac{\partial \dot{Y}_3}{\partial P_4} = Y_2$$

With (21) and (22) we can formulate the sensitivity equations (16). These equations with the initial condition (15) may then be solved using DYNAMO or a better method, e.g. a 4th order Runge-Kutta.

For small parameter changes we may assume that the change in solution is linear in the sensitivity and that parameter interactions may be regarded as negligible.

Thus, the prediction of change in solution, $\Delta \underline{Y}$, for a parameter change, $\Delta \underline{P}$, becomes

$$(23) \quad \Delta \underline{Y} = \Delta \underline{P} \cdot \underline{S}$$

Fig. 24 shows the effect of changing K_4 10% compared to the prediction (23).

The prediction on Fig. 24 is a 10% primary sensitivity chart for K_4 .

With increasing parameter changes, the assumption of linearity and of negligibility of interactions falls. Fig. 25 shows the result of changing K_2 and K_4 100% and using the prediction (23).

From Fig. 25 it is evident that we must take the interaction-effects into account.

Defining S_{ij}^k as the sensitivity of the i 'th statevariable to the j 'th parameter in a system where the k 'th parameter has been given a small change ΔP_k we have

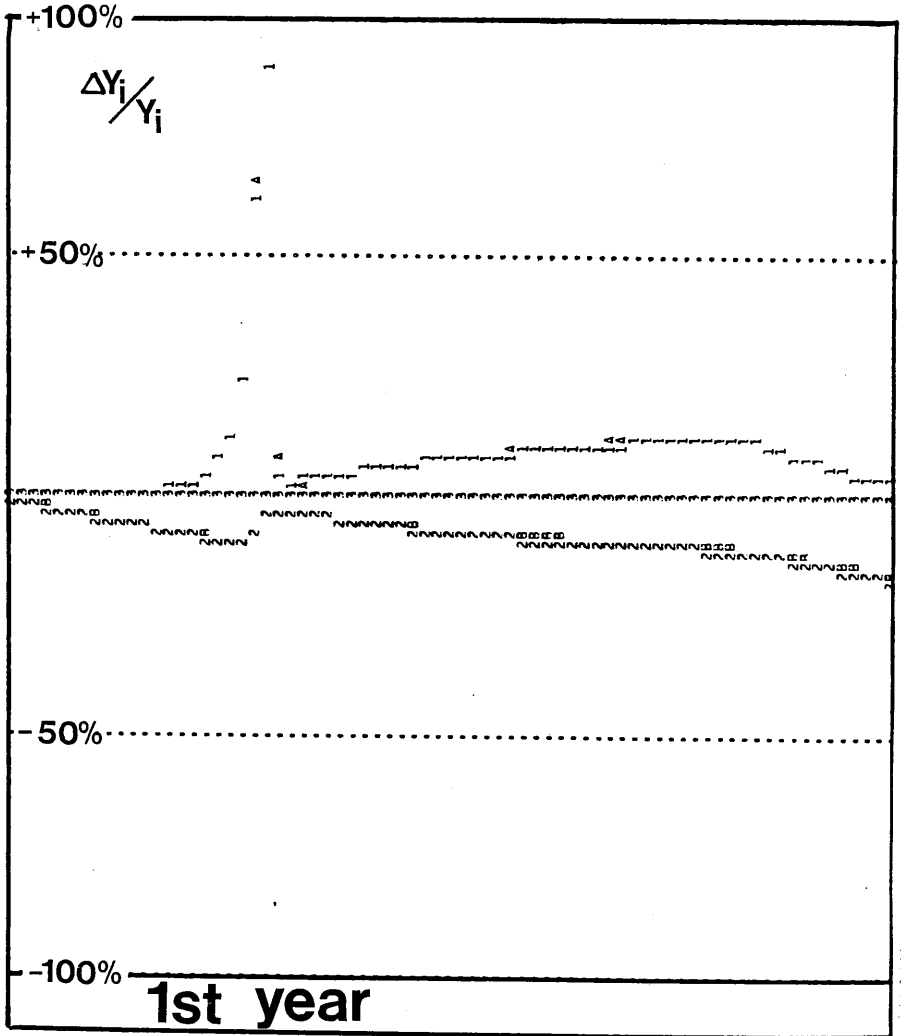


FIGURE 24: Predicted and actual percentage of change in solution due to the disturbance $\Delta P=(0\%,0\%,0\%,+10\%)$ changes exceeding $\pm 100\%$ are not shown. The prediction (23) was used.

A is actual value of Y_1 , 1 is predicted change
B is actual value of Y_2 , 2 is predicted change
C is actual value of Y_3 , 3 is predicted change
Where actual and predicted coincide, the predicted is marked

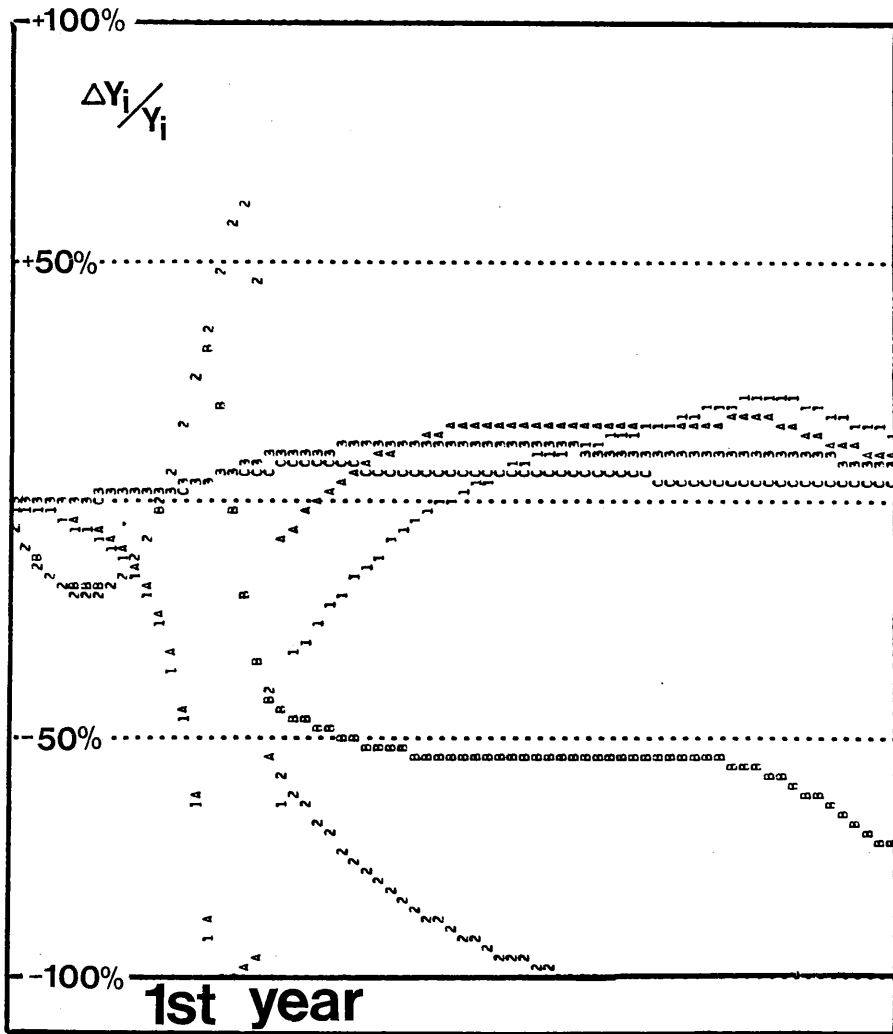


FIGURE 25: Predicted and actual percentage of change in solution due to the disturbance $\Delta P = (0\%, +100\%, 0\%, +100\%)$ changes exceeding $\pm 100\%$ are not shown. The prediction (23) was used.

A is actual value of Y_1 , 1 is predicted change

B is actual value of Y_2 , 2 is predicted change

C is actual value of Y_3 , 3 is predicted change

Where actual and predicted coincide, the predicted is marked.

$$(24) \quad \frac{\partial S_{ij}}{\partial P_k} \approx \frac{S_{ij}^k - S_{ij}}{\Delta P_k} = S_{ijk}^{(1)}$$

A first correction of the prediction (23) would then be with the meanvalue of (24), i.e.

$$(25) \quad \Delta Y_i = \sum_{j=1}^m \left[\Delta P_j \cdot \left(S_{ij} + \frac{1}{2} \sum_{k=1}^m \Delta P_k \cdot S_{ijk}^{(1)} \right) \right]$$

Fig. 26 shows the result of the approach and a reasonable agreement between the actual and predicted change in solution meaning that the secondary term in (25) expresses the effect of parameter interactions, i.e.

$$(26) \quad S_{i\Delta P_j \Delta P_k}^{*(1)} = \Delta P_j * \frac{1}{2} \Delta P_k * S_{ijk}^{(1)}$$

where $S_{i\Delta P_j \Delta P_k}^{*(1)}$ is the interaction effect on the i'th state variable of changing the i'th parameter ΔP_j and the k'te parameter ΔP_k simultaneously.

In our example we have 3x4x4 secondary effects. These 48 secondary effects change with time. Inspecting these effects once per time-unit would give 48x365=17.520 inspection values which is quite a bulk of information. A first sorting of this information can be carried out in a secondary sensitivity table. Chosing a parameter perturbation level, e.g. 50%, we may estimate the secondary effects according to (26). Testing at inspection points along the solution, e.g. an inspection point per time unit, the secondary effects in the classes below are counted:

x<-15%	-15%<x<-5%
+5%<x<+15%	+15%<x

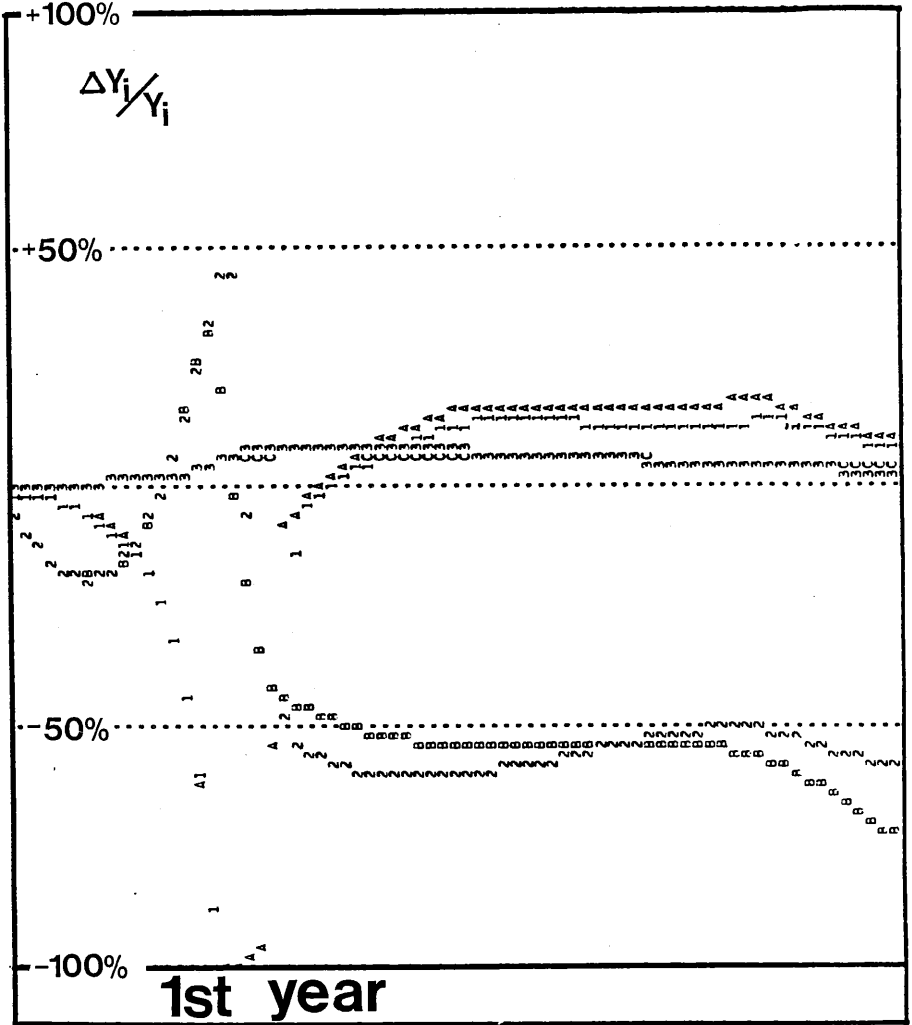


FIGURE 26: Predicted and actual percentage of change in solution due to the disturbance $\Delta P = (0\%, +100\%, 0\%, +100\%)$ changes exceeding $\pm 100\%$ are not shown.

A is actual value of Y_1 , 1 is predicted change

B is actual value of Y_2 , 2 is predicted change

C is actual value of Y_3 , 3 is predicted change

Where actual and predicted coincide, the predicted is marked.

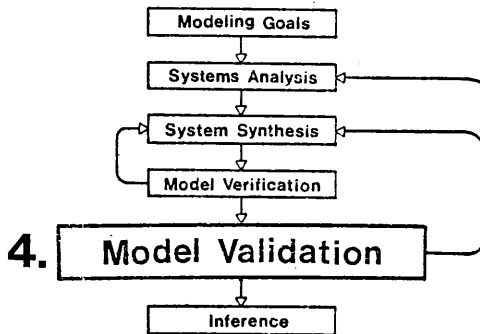
The prediction (25) was used with $S^{(1)}$ based on a 10% disturbance.

with such a table at hand, the interesting secondary effects may be identified.

The variation in time of the interesting secondary effects using (26) is then plotted analogously to Fig. 24 forming a secondary sensitivity chart.

Summarizing, a complete sensitivity analysis should include:

- (a) Relative boundaries in percent for all parameters to be included in the analysis
- (b) Primary sensitivity charts for all parameters
- (c) Secondary sensitivity tables for all parameters
- (d) Secondary sensitivity charts for important parameters selected on the basis of (c).



In the validation stage the responses from the verified model are compared to responses from the actual modeled system.

Preferably, experimentation should be carried out both with the model and with the modeled system. Whenever the modeled system is not available for experimentation, as it is the case with our master example, the risks of applying the model in the inference stage becomes greater, since the model has not been validated for the system-state in which it is applied.

The first stage in the validation is a calibration. Some of the parameters of the model are only determined to be within a certain interval. In the calibration, the actual value of the

parameter is chosen within the interval to give the best fit to measurements recorded on the actual modeled system.

Following the calibration, the actual validation is carried out through a comparison of a model using the parameterset found in the calibration to measurements recorded independently of the data used for the calibration.

In the verification stage we have checked the validity of the system synthesis stage. Consequently, a failure in the validation stage means that we should reconsider our analysis of the system, i.e. that we should return to the systems analysis stage in the modeling process. However, a failure in the calibration may sometime occur due to the way the model has been synthesized. In the instances where synthesis-failure is evident the return may be made to the system synthesis stage.

4.1 CALIBRATION

The problem faced in the calibration is sketched on Fig. 27 where model response based on the initial chosen parameters are compared to actual recordings of the performance of the modeled system. The problem is now to fit the model response to the actual data through a changing of the parameterset within its appropriate intervals. The calibration may be carried out manually or automatically.

Manual Calibration

From the sensitivity analysis we have an idea of the effects on solution of changing the parameters. With support in this knowledge we select parameters to be changed and run simulations with different parameter sets seeking to fit the data as closely as possible.

The manual calibration is often a fairly cumbersome task, but it has the advantage that the modeler gets quite an experience of the behavior of the model. This experience often proves

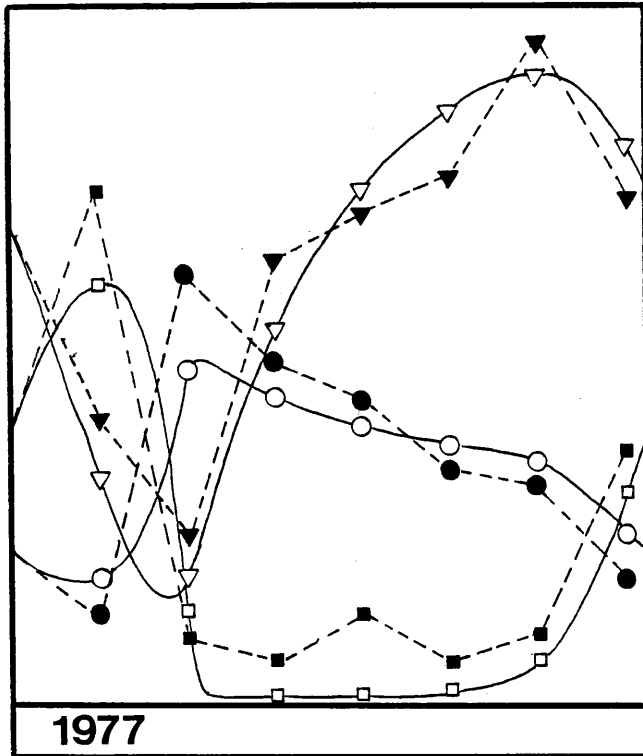


FIGURE 27: Exemplification of the calibration problem.

SIGNATURES:

- □ ▽ simulation results from uncalibrated model
- ■ ▼ actual measurements of the modeled system.

important later in the modeling process.

Automatic calibration*

In the calibration we seek to minimize the difference between observed and modeled behavior. Defining $Y_i^{OBS}(t)$ as the observed value of state variable i at time t we may say that we seek to minimize a function $V = v(\underline{Y}(t), \underline{Y}^{OBS}(t))$. The function v may be chosen freely as long as the following implications are valid

$$(27) \quad \underline{Y}(t_J) \neq \underline{Y}^{OBS}(t_J) \Rightarrow V > 0 \quad \wedge$$
$$\underline{Y}(t_J) = \underline{Y}^{OBS}(t_J), \forall J \Rightarrow V = 0$$

The most used form of v is a quadratic equation

$$(28) \quad V = \frac{1}{m*n} \sum_{i=1}^n w_i \sum_{J=1}^m \left(\frac{Y_i(t_J) - Y_i^{OBS}(t_J)}{Y_i^{OBS}} \right)^2$$

where w_i are weighting factors, e.g. reverse proportional to experimental standard deviations.

The calibration may now be stated as the following non-linear programming problem

$$(29) \quad \text{minimize } V$$

subject to

$$\dot{Y} = f(\underline{Y}, \underline{P}, t)$$

$$P_{i_{LO}} \leq P_i \leq P_{i_{UP}} \quad (Vi)$$

where $P_{i_{LO}}$ and $P_{i_{UP}}$ are lower and upper bounds for the i 'th parameter respectively.

* This section is partly based on [10]

Several iterative methods for solving (29) exist. For an introduction to the methods see [11, pp. 120-139]. A more detailed description of a suitable method is found in [12].

4.2 GOODNESS OF FIT

With the parameterset found under the calibration, a comparison to a dataserie recorded independently of the one used in the calibration now constitutes the actual validation.

Ideally, the comparison should be carried out through a statistical test for identity of two time series. However, it is very seldom to have enough data to analyse the measurements as a meaningful timeserie. Consequently, the "Goodness of fit" is mostly judged using (28) or through a verbalization of a rough estimate.

[13] gives the following method of comparison with the emphasis placed on mean values.

The simulation time is divided into a number of equally long intervals. Let \bar{Y}_{iK}^{OBS} be the observed mean for variable i in the k 'th time interval and \bar{Y}_{iK} the corresponding simulated mean. Then, the difference of means $\bar{d} = \bar{Y} - \bar{Y}^{OBS}$ may be assumed to be distributed as a student "t" probability density function. Assuming the variance of the model equal to the observed variance we have

$$(30) \quad t = \frac{\bar{d} - \delta}{S_{\bar{d}}}$$

where δ is the true difference between the model and the data and $S_{\bar{d}}$ is the standard deviation of the difference

$$(31) \quad S_{\bar{d}} = \frac{2S_x^2}{N}$$

where S_x^2 is the data variance for a specific time interval and variable.

Under the null hypothesis: $\delta=0$, there is a "critical" \bar{d} which delineates the region of rejection of the hypothesis and is given by

$$(32) \quad \bar{d}_c = \pm t s_{\bar{d}}$$

and for a 95% confidence range (5% chance of making a Type I error),

$$(33) \quad \bar{d}_c = \pm 2 s_{\bar{d}} = \frac{2.83}{\sqrt{N}} s_x$$

The distribution of \bar{d} and the critical regions are shown in Fig. 28. If

$$(-\bar{d}_c < \bar{d} < \bar{d}_c)_{ik}$$

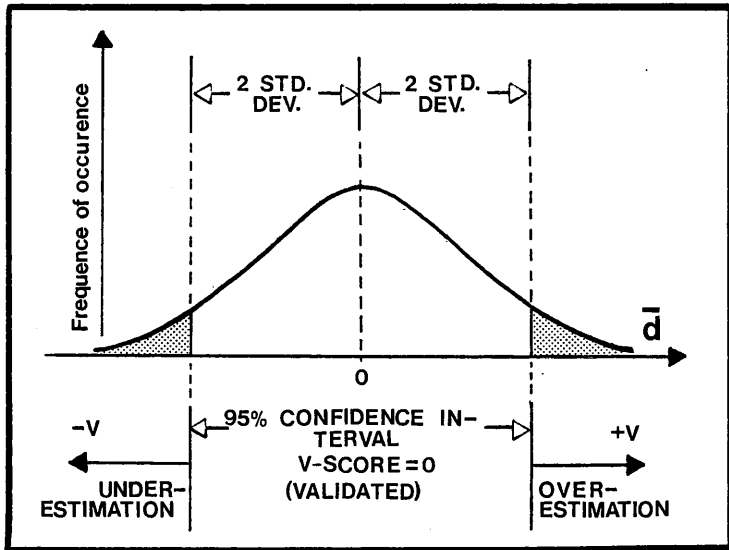


FIGURE 28: Determination of validation score V

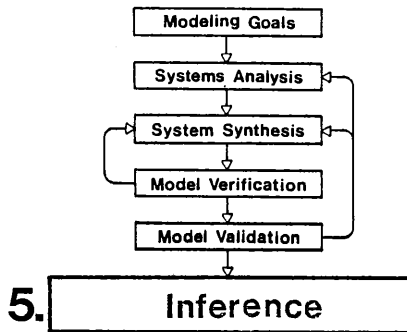
the model is considered validated for the i 'th variable in the k 'th time interval and a score $V=0$ is given.

A positive value of V , therefore, indicates an overestimate of the mean while a negative value of V indicates an underestimate of the mean. The V score is therefore a measure of the degree to which the model deviates from the observed data.

The full V score is given by

$$(34) \quad V_{ik} = 0 \text{ for } |\bar{d}_{ik}| < \bar{d}_c$$
$$V_{ik} = \bar{d}_{ik} - \bar{d}_c \text{ for } -\bar{d} > \bar{d}_c$$
$$= -\bar{d}_{ik} + \bar{d}_c \text{ for } -\bar{d} > -\bar{d}_c$$

[13] also gives an application of the method above. Another example of the use of the method is found in [14].



In the inference stage, experiments are conducted solely with the verified validated model. Mihram [2] suggests, that the aims of this experimentation for most models may be placed in one of the following categories:

- 1) The determination of the dynamic behavior of the model during a stipulated period of time.
- 2) The determination of the relative (or marginal) effects of unit changes in the environmental conditions on the model's expected response at the end of a specified period of time.
- 3) The determination of the particular environmental specifications at which the models expected response is optimized.

5.1 MASTER EXAMPLE

In order to simplify the exemplification of the various stages in the modeling process, we have postponed the formulation of a total model for Dump City lake and only exemplified the process with the biological subsystem model. In this section we build a total model and assume this model to have undergone the modeling process, i.e. to be a fully verified validated model. We then exemplify an aspect of the inference stage with the model.

Our primary concern in formulating a total model of the lake is the transportation subsystem which we have neglected this far.

Asking a hydrologist, we get the following result of an investigation:

The upper and lower lake may be regarded as being completely mixed. The water has been determined to flow as depicted on Fig. 29.

There is no storing of water in the lake so the following continuity expressions are valid

$$Q_{11i} + Q_{12i} = Q_{1u}$$

$$Q_{1u} + Q_{2i} = Q_{2u}$$

Fig. 30 gives a schematic picture of the lake.

The material flows caused by the water are concentration flows, i.e. the material transport equals the concentration where the flow originates times the flow. The sediment does not take part in the transportation process.

With this knowledge we decide to split up the biological subsystem into two, one for each box. The forcing from the transportation subsystem may then be modeled by material flows in and out of the two biological compartments.

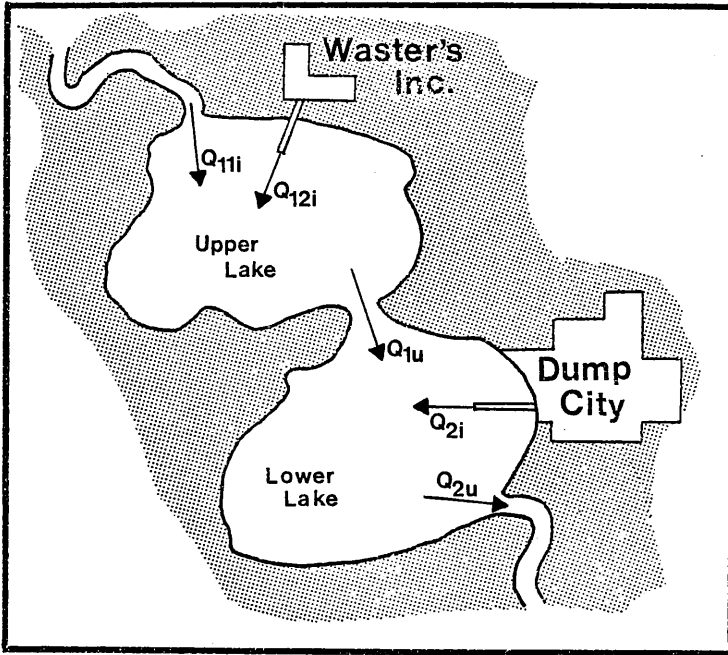


FIGURE 29: Water flows in Dump City lake.

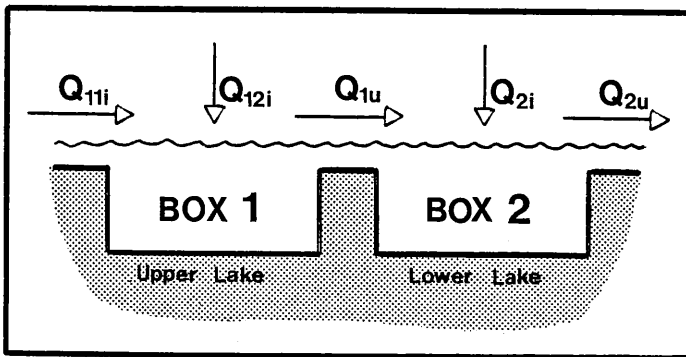


FIGURE 30: Schematic representation of Dump City lake.

Were we to assemble the model now with the proper flows and loads of pollutants, we would get a resulting output from the model in terms of an Algal biomass. The Algal biomass is of little relevans to the citizens of Dump City. They would be far more interested in e.g. the transparency of the water in the lake.

The biologist tells us, that at the present, a much used method of determining the water transparency is with a secchi-disk. A secchi-disk is a white disk with a diameter of 25 cm. For measurements, the disk is lowered in the shadow of a boat. The distance from the surface at which the disk may no longer be seen is the water transparency (the secchi-disk reading). Furthermore, the biologist produces Fig. 31. from Mathiesen [15] showing the effect on water-transparency of changes in primary production from a number of Danish lakes.

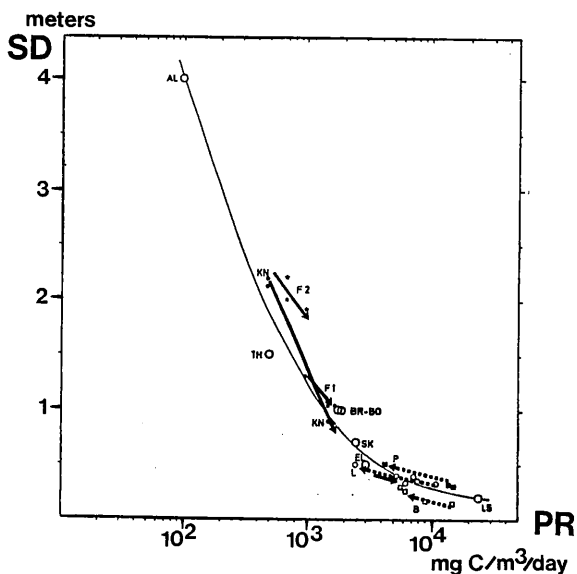


FIGURE 31: Secchi-disk reading (SD) versus primary production for a number of Danish lakes. From [15].

Setting a maximum transparency at 10 meter and using our unit of measurement of primary production, the relation from Fig. 31 between primary production (PR) water transparency (TP) for a lake with volume VOL may be expressed as

$$(35) \quad TP = \frac{.5}{\sqrt{PR/VOL} + .05}$$

Using (35) we may translate the algal biomass into something meaningful for the citizens of Dump City.

We now check on the outlet of pollutant from Wasters Inc. and Dump City. We find that these may be regarded as constant over time. Furthermore, we find, that the river leading to Upper lake is completely unpolluted.

Using the transportation subsystem and the pollution as forcing on the two-compartment biological subsystem and denoting quantities pertaining to Upper lake with a "1" and to Lower lake with a "2" we may draw the System Dynamics flowchart on Fig. 32.

The erection of a waste water treatment plant from the view of the biological system means cutting off a part of the pollution. We model this using the fraction of the pollutants remaining after waste water treatment, R_1 and R_2 for Upper and Lower lake respectively.

After a determination of various constants of the system, the following DYNAMO program for the lake may be written with support in Fig. 32 (See p.107).

The result of running the following program is given in Fig. 33. (See p. 109).

We now assume, that the two box model has undergone the modeling process, i.e. the two box model is fully verified and validated. We may then proceed to exemplify the inference stage with the model.

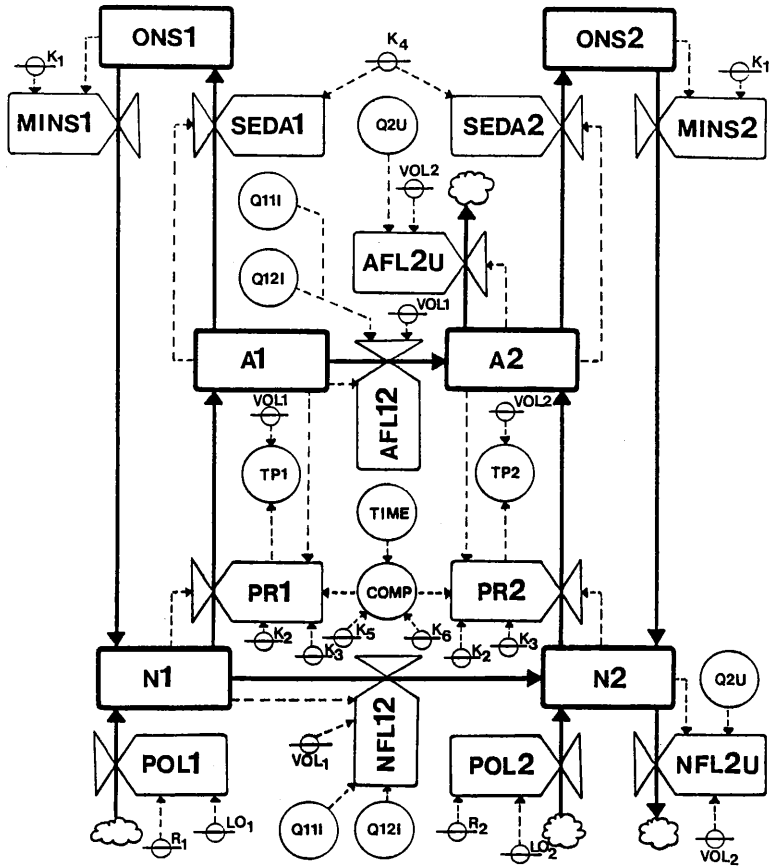


FIGURE 32: System Dynamic flowchart for the total model of Dump City lake.

* MODEL OF BIOLOGICAL SUBSYSTEM.

NOTE

NOTE UPPER LAKE

NOTE

L $N1.K = N1.J + DT * (MINS1.JK - PR1.JK - NFL12.JK + POL1.JK)$

L $A1.K = A1.J + DT * (PR1.JK - SEDA1.JK - AFL12.JK)$

L $ONS1.K = ONS1.J + DT * (SEDA1.JK - MINS1.JK)$

NOTE

R $MINS1.KL = K1 * ONS1.K$

R $PR1.KL = K2 * A1.K * (N1.K / (N1.K + K3)) * COMP.K$

R $SEDA1.KL = K4 * A1.K$

R $NFL12.KL = (N1.K / VOL1) * (Q111.K + Q121.K)$

R $AFL12.KL = (A1.K / VOL1) * (Q111.K + Q121.K)$

R $POL1.KL = LO1 * R1$

NOTE

A $TP1.K = .5 / (SQRT(PR1.JK / VOL1) + .05)$

NOTE

N $N1 = 14.$

N $A1 = 9.$

N $ONS1 = 125.$

NOTE

C $VOL1 = 10.$

C $LO1 = .85$

C $R1 = 1.$

NOTE

NOTE LOWER LAKE

NOTE

L $N2.K = N2.J + DT * (MINS2.JK - PR2.JK + NFL12.JK - NFL2U.JK + POL2.JK)$

L $A2.K = A2.J + DT * (PR2.JK - SEDA2.JK + AFL12.JK - AFL2U.JK)$

L $ONS2.K = ONS2.J + DT * (SEDA2.JK - MINS2.JK)$

NOTE

R $MINS2.KL = K1 * ONS2.K$

R $PR2.KL = K2 * A2.K * (N2.K / (N2.K + K3)) * COMP.K$

R $SEDA2.KL = K4 * A2.K$

R $NFL2U.KL = (N2.K / VOL2) * Q2U.K$

R $AFL2U.KL = (A2.K / VOL2) * Q2U.K$

R $POL2.KL = LO2 * R2$

NOTE

A $TP2.K = .5 / (SQRT(PR2.JK / VOL2) + .05)$

NOTE

N $N2 = 2.8$

N $A2 = 1.8$

N $ONS2 = 25.$

NOTE

C $VOL2 = 3.5$

```
C      L02=.25
C      R2=1.
NOTE
A      COMP.K=.5*(1+K6*SIN((6.283/K5)*(TIME.K-80)))
NOTE
NOTE  CURRENTS
NOTE
A      Q11I.K=.1
A      Q12I.K=.01
A      Q2U.K=.12
NOTE
C      K1=.001
C      K2=.07
C      K3=.12
C      K4=.01
C      K5=365.
C      K6=1.
NOTE
NOTE  RUN ADMINISTRATION
NOTE
A      PLTPER.K=5.
PRINT A1,N1,ONS1,TP1,A2,N2,ONS2,TP2
PLOT  A1=A/A2=E/N1=N/N2=M/ONS1=0/ONS2=0/TP1=1/TP2=2
SPEC  DT=.10/LENGTH=365/PRTPER=10
RUN   TWO BOX MODEL
```

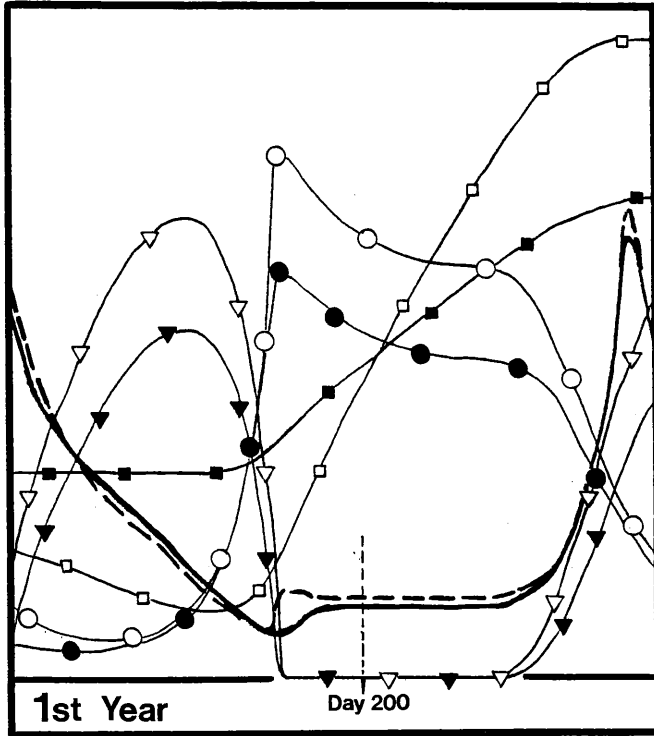
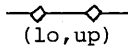




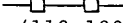
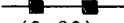

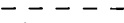



FIGURE 33: First year simulation of the two box model.

SIGNATURES:



lo: Lower bound of figure
 up: Upper bound of figure

	Upper lake	Lower lake
Algal biomass	 (0, 80)	 (0, 40)
Nutrients	 (0, 80)	 (0, 40)
Sediment	 (110, 190)	 (0, 80)
Water transparency	 (0, 12)	 (0, 12)

From Fig. 33 we notice that the water transparency is almost constant during the summer. For the sake of simplicity, we choose the level of summer water transparency to be our water quality parameter and decide to link this to the value at the 200th day of the year.

Simulating different values of R_1 and R_2 and defining E_1 and E_2 as the efficiency of the waste water treatment at Upper and Lower lake respectively ($E_1=1-R_1$) we obtain Fig. 34.

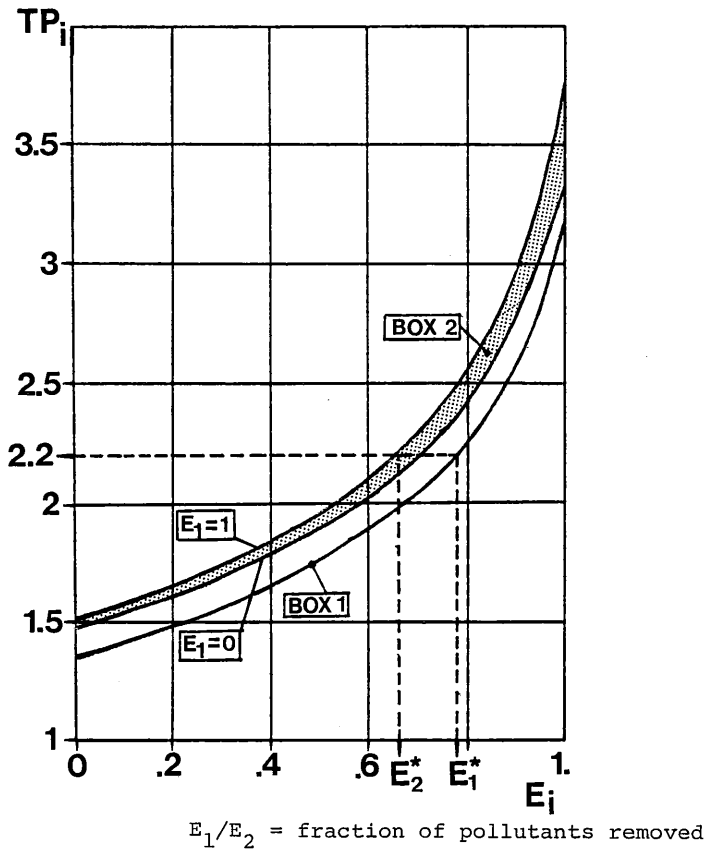


FIGURE 34: Efficiency of waste water treatment versus resulting water transparency. Efficiency, E_1 , is expressed as the fraction of pollutants removed from the waste water. TP_1 is the value on day 200. E_1^* and E_2^* : See text.

With Fig. 34 we may advise the citizens of Dump City. If they want a specific water quality, the figure will show how efficient the waste water treatment must be. On Fig. 34 this is exemplified with a desired water transparency of 2.2 meters in both Upper and Lower lake. To obtain this goal, the efficiency of the waste water treatment in Upper lake must be $E_1^* = 78\%$ and $E_2^* = 67\%$ for Lower lake.

However, the good citizens may not want to set a specific water quality for each of the lakes. More likely, they will be prepared to spend a certain amount of money and will want the best overall improvement of waterquality this amount will buy.

To advice in this matter we check the cost of waste water treatment with firms in the field. The result is the costfunction on Fig. 35. The reason why the cost is not linear in efficiency is

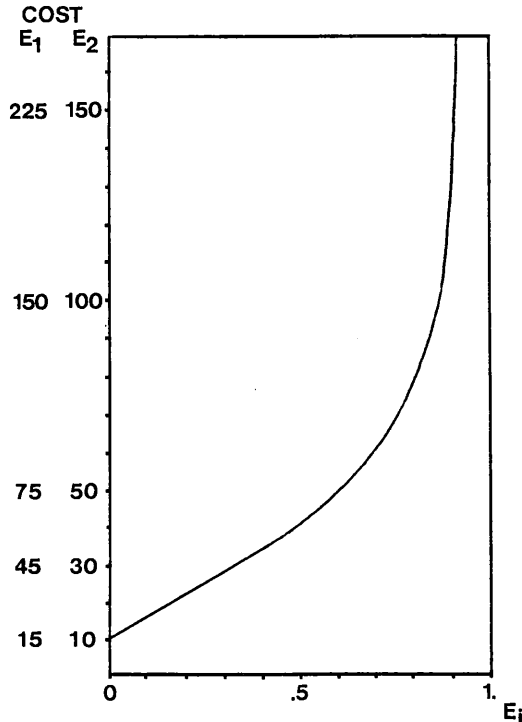


FIGURE 35: Efficiency of wastewater treatment versus costs.

that it becomes increasingly difficult to remove pollutants the more diluted they are. Furthermore, a bigger mass of pollutants have to be removed at Upper lake leading to the higher cost there.

Defining the overall waterquality as the sum of the two water transparencies $TP1+TP2$ we may draw the isoquality lines on Fig. 36 from simulations with different R_1 and R_2 . With support in Fig. 35 we may draw the isocost lines on the figure. From the figure we may conclude that with small investments,

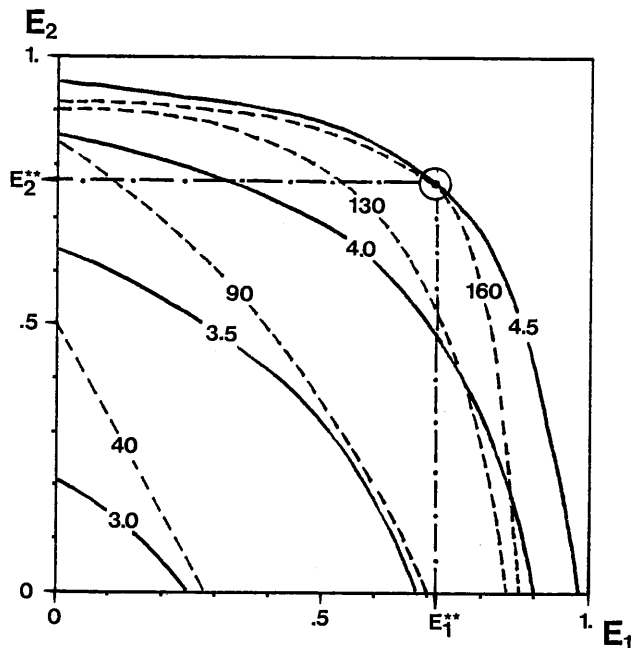


FIGURE 36: Isocost and isoquality lines for Dump City lake.

SIGNATURES: ——— Isoquality ($TP1+TP2$).
 - - - - - Isocost.

E_1^{**} and E_2^{**} : See text.

e.g. 40 monetary units, the best overall waterquality is obtained through solely treating the wastewater in Lower lake. With increasing investments the situation changes. In the case of an investment of 160 monetary units the best overall waterquality is obtained for $E_1^{**} = 72\%$ and $E_2^{**} = 77\%$. Using Fig. 34 we find the resulting water transparencies $TP_1 = 2.08$ meters and $TP_2 = 2.42$ meters. We may compare this investment to the one needed for the implementation of the first suggestion $TP_1 = TP_2 = 2.20$ meters. From Fig. 35 we get $E_1 = 78\% \Rightarrow \text{cost} = 115$ and $E_2 = 67\% \Rightarrow \text{cost} = 58$ or a resulting cost of 173 monetary units for an overall waterquality of 4.40 meters.

And the good citizens of Dump City, now having an idea of where to clean and how much, at once started an even fiercer argument about who should pay

REFERENCES

- [1] Clausen, S., Kybernetik - Systemer og Modeller, IMSOR, 1969, in Danish.
- [2] Mihram, G. A., The Modeling Process, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-2, No. 5, 1972.
- [3] Buckminster Fuller, R., Operating Manual for Spaceship Earth, Southern Illinois University Press, 1969.
- [4] Jackson, A. S., Analog Computation, McGraw-Hill Book Company, 1960.
- [5] Pugh III, A. L., DYNAMO II User's Manual, The MIT Press, Cambridge, Mass., 1973.
- [6] Behrens, J. C., Sørensen, P. E., COLTS (Continuous Long-Term Simulation - a System Dynamics Computer Simulation Tool, to appear in Management Science special issue on System Dynamics, fall 1979.
- [7] Gear, C. W., The Numerical Integration of Ordinary differential Equations, Math. Comp., 21, 1967.
- [8] Gear, C. W., Algorithm 407, DIFSUB for Solution of Ordinary Differential Equations, Comm. ACM, 14, 1971.
- [9] Behrens, J. C., A Semi-Analytical Approach to the Transient Sensitivity of Non-Linear Systems, Applied Mathematical Modelling, vol. 3, no. 2, 1979, p. 105-115.

- [10] Mejer, H., Systemanalyse for Miljøkontrollinien, KU, 1977, in Danish.
- [11] Vidal, R. V. V., Notes on Static and Dynamic Optimization, IMSOR, 1976.
- [12] Goldfarb, D., Extension of Davion's Variable Metric Method to Maximization under Linear Inequality and Equality Constraints, SIAM Journal on Applied Mathematics, Vol. 17, 1969, pp. 739-764.
- [13] Thomann, R. V., Winfield, R. P., On the Verification of a Three-Dimensional Phytoplankton Model of Lake Ontario, Proceedings of the Conference Environmental Modeling and Simulation, 1976, Cinn., Ohio, EPA 600/9-76-016, (July, 1976), ORD and OPM, EPA, Wash. DC.
- [14] Dahl-Madsen, K. I., Mathematical Modelling of Eutrophied Coastal Areas, the Water Quality Institute, DK-2970 Hørsholm, 1976.
- [15] Mathiesen, H., Miljøændringer og biologisk effekt i Søer, Sårtryck ur Vatten, 2/70, 1970, in Danish.
- [16] Naill, R. F. Numerical Solution to System Dynamics Models, System Dynamics Group, Thayer School of Engineering, Hanover, N.H., 1975.
- [17] Jacobsen, S. K., Elementær Operationsanalyse, IMSOR, 1975, in Danish.
- [18] Sharp, J. A., Sensitivity Analysis Methods for System Dynamics Models, Proceedings of the 1976 International Conference on System Dynamics, Geilo, Norway, 1976.

- [19] Forrester, J. W., Industrial Dynamics, the MIT Press, 1961.
- [20] Forrester, J. W., Principles of Systems, Wright-Allen Press, 1968.
- [21] Behrens, J.C., Sørensen, P.E., Bjørn-Hansen, S., and Mungaard, D., COLTS version 1.0 - Syntax and Semantics, IMSOR, Research report no. 6/1979, pp. 60.
- [22] Nørsett, S.P., Semi Explicit Runge-Kutta Methods, Math. and Comp. 6/74, dept. of Math, NTH, ISBN-82-7151-009-6.
- [23] Lambert, J.D., Computational Methods in Ordinary Differential Equations, J. Wiley & Sons, 1973, pp. 278.
- [24] Thomsen, P.G., Numerisk løsning af sædvanlige differentiaalligninger, Numerisk Institut, DTH, Hæfte 27, 1974, pp. 70 (in danish).
- [25] Munter, P., Numerical Methods for ordinary Differential Equations containing Discontinuities, (in danish), The Refrigeration Laboratory, The Technical university of Denmark, 1978.
- [26] Fehlberg, E., Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten - Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme, Computing 6, 61-71, Springer verlag, 1970.
- [27] Bulirsch, R. and Stoer, J., Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods, Numerische Mathematik 8, 1-13, 1966.
- [28] Lotka, A.J., Elements of Mathematical Biology, 1956, Dover.

APPENDIX A: JCL for DYNAMO at NEUCC

Small programs may run under the express monitor with the following setup:

```
//A123456 (***,NEU),'USERNAME',CLASS=W
/*ROUTE PRINT LOCAL
AJOB DYNAMO A123456

*****
*                               *
* DYNAMO PROGRAM                *
*                               *
*****

/*
```

If a larger model is to be run, the following setup will execute under OS:

```
//A123456 (***,NEU,T,L),'USERNAME',REGION=156K
/*ROUTE PRINT LOCAL
//~ EXEC DYNAMO
//SYSIN DD *

*****
*                               *
* DYNAMO PROGRAM                *
*                               *
*****

/*
```

In the first line, T is the allocated execution time in tenths of a minut, L is the number of output lines in thousands.

APPENDIX B: JCL for COLTS
- syntax and examples

Notice: version 1.0 requires a minimum of 260K storage.

SYNTAX (default values are underscored)

// EXEC COLTS

{,CPARM='<List of COLTS parms>'}⁰₁

<List of COLTS parms>::=

<COLTS parm>{,<COLTS parm>}⁰_n

<COLTS parm>::= {

{ SINGLE }	0	}	1
{ DOUBLE }	1		
{ TRY }	0	}	1
{ NOTRY }	1		
{ PRINT }	0	}	1
{ NOPRINT }	1		

 }

{,FPARM='<List of valid FORTG parms>'}⁰₁

{,FPRINT=' { UNIT=AFF=SYSPRINT }¹, }⁰₁
DUMMY

{,LPARM='<List of valid LOADER parms>'}⁰₁

{,LPRINT=' { UNIT=AFF=SYSPRINT }¹, }⁰₁
DUMMY

{,LLIB='<dataset name>'}⁰₁

{,COLT4='<dataset name>',DISP='<disp.List>'}⁰₁

$$\left\{ \begin{array}{l} , \text{CMAX} \\ , \text{FMAC} \\ , \text{LMAX} \end{array} \right\}^1 = \left\{ \begin{array}{l} 0 \\ \underline{4} \\ 8 \end{array} \right\}^1 \left. \vphantom{\begin{array}{l} , \text{CMAX} \\ , \text{FMAC} \\ , \text{LMAX} \end{array}} \right\}^3_0$$

//CSYSIN DD <COLTS input definition>

//FSYSIN DD <FORTRAN input definition>

SEMANTICS

CPARM

$\left\{ \begin{array}{l} \underline{\text{SINGLE}} \\ \underline{\text{DOUBLE}} \end{array} \right\}$

with SINGLE, COLTS will generate a SINGLE precision FORTRAN program (REAL*4), DOUBLE will give a DOUBLE precision program (REAL*8). It should be noted that user defined functions and integration routines must be given in a precision fitting for the chosen CPARM.

$\left\{ \begin{array}{l} \underline{\text{TRY}} \\ \underline{\text{NOTRY}} \end{array} \right\}$

In case of error while executing a generated FORTRAN program, CPARM='TRY' will cause the output routine PRINT to output the results that have been stored. It should be noted that the last result from PRINT may not coincide with the error point due to untransferred buffers.

$\left\{ \begin{array}{l} \underline{\text{PRINT}} \\ \underline{\text{NOPRINT}} \end{array} \right\}$

tells whether the COLTS input deck should be listed or not.

FPRINT

UNIT=AFF=SYSPRINT

will give a listing of the generated FORTRAN program

LPRINT

UNIT=AFF=SYSPRINT will give output from the LOADER

LLIB

specifies a user load module library

COLT4

If COLT4 is defined as a user dataset with DISP='KEEP' (or equivalent disposition) the generated FORTRAN program will be retained after COLTS has terminated.

{
 CMAX
 FMAX
 LMAX
}

allows for a redefinition of the condition codes at which the various remaining phases will be skipped by the COLTS monitor, e.g.

FMAX=8 allows for LOADER, GO and OUTPUT phases even if a returncode = 8 was received from the FORTRAN compiler.

EXAMPLES

1) Running a COLTS program

- a) in SINGLE PRECISION with COLTS input on punch cards

```
// EXEC COLTS  
//CSYSIN DD *
```

COLTS PROGRAM

- b) in DOUBLE PRECISION with COLTS input on dataset NEU.A123456.USER under member PROG. No Listing of COLTS input but listing of generated FORTRAN program

```
// EXEC COLTS,  
// CPARM='DOUBLE,NOPRINT',  
// FPRINT='UNIT=AFF=SYSPRINT'  
//CSYSIN DD DSN=NEU.A123456.USER(PROG),  
// DISP=SHR
```

- 2) Supplying FORTRAN functions and/or user defined integration routine with diagnostics routine as source on punch-cards.

```
// EXEC COLTS  
//CSYSIN DD.*
```

COLTS PROGRAM

```
//FSYSIN DD *
```

FORTRAN FUNCTIONS

USER INTEGRATION

USER DIAGNOSTICS

- 3) Supplying functions and/or user defined integration as load modules on the partitioned cataloged dataset
NEU.A123456.USERLIB

```
// EXEC COLTS,  
// LLIB='NEU.A123456.USERLIB'  
//CSYSIN DD *
```

It should be noted that the functions and routines must reside under membername=function or routine name.

- 4) Defining FORTRAN channel 13 to contain datacards which are used by the FORTRAN function supplied as source on NEU.A123456.USER(FUNC)

```
// EXEC COLTS
//FT13F001 DD *
```

DATA

```
//CSYSIN DD *
```

COLTS PROGRAM

```
//FSYSIN DD DSN=NEU.A123456.USER(FUNC),
// DISP=SHR
```

- 5) The use of a variable starting with ZZ will give an E-type error (condition code=8). Being a daring user you wish to run your program in spite of this. (It should be noted that the results could be rather strange)

```
// EXEC COLTS,CMAX=8
//CSYSIN DD *
```

COLTS PROGRAM

- 6) Retaining the generated FORTRAN program on source dataset NEU.A123456.USER(COLPRG)

```
// EXEC COLTS,
// COLT4='NEU.A123456.USER(COLPRG)',DISP='OLD,KEEP'
//CSYSIN DD *
```

COLTS PROGRAM

It should be noted that version 1.0 is incapable of further treatment of retained FORTRAN programs. Version 2.0 is scheduled to have this ability.

<u>INDEX</u>	Page
A	22
ABSERR	70
analogue	12, 25
arithmetic	35
* (=asterix)	33, 41
auxiliary	27, 33, 35
behavior	77
BUST	70
C	33
calibration	93, 94
card format	32
card types	33
causal diagram	21
character set	31
CLIP	36
COLTS	68
communication	9
constant	27, 33
descriptive	12
deterministic	11
drain	27
DT	32, 35, 42
dynamic	11
environment	9
error accumulation	54
EULR	70
feedback	19, 21
FLAG	70
forcing function	19
formal	11, 12

GEAR	70
goodness of fit	97
inference	15, 101, 102
information flow	27
INITDT	70
integration	50
JCL DYNAMO	108
COLTS	119
L	33
LENGTH	32, 42
level	27, 33, 35
master example	17, 7, 10
- - behavior	78
- - complete program	43
- - equation writing	38
- - flowchart	30
- - inference	102
- - introduction	7
- - modeling goals	15, 17
- - rounding errors	60
- - sensitivity	84
- - system	10
- - systems analysis	19
material	11
MAX	36
MAXDT	70
MIN	36
MINDT	70
model	11
modeling goals	15, 17

N	33
NAME	70
NOISE	37
NOTE	33
numerical values	31
phase error	51
PHSPER	75
physical flow	27
II	36
PLOT	33, 41
PLTPER	32, 41, 42
PRINT	33, 41
process	27
PRTPER	32, 42
quantity names	32
R	33
rate	27, 33, 35
recursiveness	10
RELERR	70
replica	12
RKF4	70
rounding errors	60
RUN	34, 43
scaling	46
sensitivity	71
- chart	87, 92
- table	90, 92
SIM	70
simular	12
SIRK	70
source	27
SPEC	33
stability	63

state	19
state variable	19
static	11
STEP	37
stochastic	11
subsystem	10
SWITCH	37
symbolic	12
system	9, 10
systems analysis	15, 19, 20, 95
system synthesis	15, 25, 50, 95
T	33
table	33, 37, 38
TIME	32, 37
time sequencing	34, 35
TIMOUT	70
trigonometric	36
universe	9
validation	15, 93
verification	15, 49
X	34

April 1978 JCB/bp

May 1979 JCB/krn