BIOMOD: A USER'S VIEW OF AN INTERACTIVE COMPUTER SYSTEM FOR BIOLOGICAL MODELING (A PRELIMINARY REPORT)

G. F. Groner, R. A. Berman, R. L. Clark and E. C. DeLand

Prepared for
NATIONAL INSTITUTES OF HEALTH
BIOMOD: A USER'S VIEW OF AN INTERACTIVE COMPUTER SYSTEM FOR BIOLOGICAL MODELING (A PRELIMINARY REPORT)

G. F. Groner, R. A. Berman, R. L. Clark and E. C. DeLand

This investigation was supported by Public Health Service Research Grant Number GM 15896-03, from the National Heart Institute, National Institutes of Health, and in part by The Rand Corporation.
PREFACE

This Memorandum, part of Rand's continuing effort in biological modeling and facilitating man-computer communication, is a preliminary report about BIOMOD, an interactive computer-graphics system presently under development. We present the current status of BIOMOD, and demonstrate the use of this system for biological modeling. This Memorandum will be useful to those interested in biological modeling or in the application of interactive computer graphics. Later reports will describe and evaluate BIOMOD in greater detail.

This work was conducted under NIH Research Grant GM15896-03.
SUMMARY

This Memorandum presents an example of the use of the current version of BIOMOD, an interactive computer-graphics system for biological modeling. The model in the example is of water and solute distribution between the intravascular and extravascular spaces of the body. BIOMOD is presently under development; the Memorandum distinguishes between implemented and planned features.

The BIOMOD system features interaction, hierarchical model structuring, and user-oriented model-definition languages. The system operates on an interactive graphics console comprising a cathode ray tube screen, a RAND Tablet, and a keyboard. It allows a user to draw block diagrams, handprint or type text, push displayed "buttons," and drag labels. BIOMOD provides immediate feedback about its interpretation of user actions and their validity. A user may represent a model by a block diagram--each component of which may be defined by another block diagram. This facilitates devising complex models a portion at a time. A user ultimately defines model components by analog computer-like elements, or by CSMP/360 statements. When BIOMOD is completed, model components may alternately be defined by chemical equations, differential equations, or data curves, obviating the requirement that the user translate his model description into a conventional simulation language. During model simulation, a user may display curves for different variables, change scales, or alter simulation parameters.
ACKNOWLEDGMENTS

We are indebted to our colleagues J. F. Heafner and W. L. Sibley for their advice in modifying GRAIL for the BIOMOD construction phase, and to Iona Blackwell for her assistance in implementing BIOMOD.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td>iii</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vii</td>
</tr>
<tr>
<td>FIGURES</td>
<td>xi</td>
</tr>
</tbody>
</table>

### Section I. INTRODUCTION
- System Overview .................................. 1
- The Environment .................................. 2
- The RAND Tablet .................................. 3
- The Display Screen ................................ 4
- The Keyboard ...................................... 4

### Section II. CONSTRUCTION PHASE
- Block Diagrams .................................... 5
- Multilingual Design ................................ 7
- CSMP Statements ................................... 8
- User-Defined Functions ............................ 10
- Hierarchy ......................................... 11
- Primitive Functions ............................... 11
- Data Page ......................................... 14
- A Second User-Defined Function .................. 16
- Using User-Defined Functions .................... 18
- Model Data Page ................................... 21

### Section III. SIMULATION PHASE
- Simulation Display ................................ 24
- Plotting Region ................................... 25
- Y-NAME Box ....................................... 25
- X-NAME Box ....................................... 25
- Range Boxes ...................................... 25
- Linear-Log Buttons ............................... 26
- Plotting Interval ................................ 26
- Current X ........................................ 26
- Message Area ..................................... 27
- Integration Parameters ........................... 27
- Specifying Y-NAME Variables .................... 27
- Rescaling ......................................... 29
- Thinning Data ..................................... 29
- Trouble Shooting .................................. 30
- Changing Parameters .............................. 31

### Section IV. CONCLUSION
- ................................................. 34

### REFERENCES
- ................................................. 35
FIGURES

1. A User at the Console ........................................ 3
2. System of Conceptual Compartments ..................... 6
3. Block Diagram of Compartmental Model ............... 6
4. Language Choice Page ...................................... 7
5. A CSMP Coding Form ....................................... 8
6. Graph of Pressure Pulse in Plasma Compartment .......... 9
7. CSMP Coding Form to Define PRESSURE ..................... 10
8. Block Diagram for CMPT1 .................................. 12
9. Argument List for INTGRL Function ..................... 12
10. Formal and Local Arguments for COMPUTE VOLUME ........ 13
11. CSMP Coding Form to Compute VOLDOT .................... 15
12. Data Page for CMPT1 Function ........................... 16
13. CSMP Coding Form to Define CMPT2 ..................... 19
14. Data Page for CMPT2 ..................................... 19
15. Formal and Local Arguments for RED CELL ........... 20
16. Formal and Local Arguments for INTRAC .............. 20
17. Formal and Local Arguments for PLASMA .............. 21
18. Formal and Local Arguments for INTERS ............. 22
19. Data Page for Overall Model ............................ 22
20. The Simulation Display .................................. 24
21. The Integration Method Display ......................... 28
22. Plot of VP versus TIME .................................. 28
23. Rescaled Plot of VP versus TIME ...................... 29
24. Complete Plot of VP versus TIME ...................... 30
25. Plots of VP and P versus TIME ........................ 31
26. The Parameter Display .................................. 32
27. Plot of VP versus TIME with KS=800 Shown with a Meter ....................................................... 33
28. The Variables Display .................................. 33
I. INTRODUCTION

SYSTEM OVERVIEW

The BIOMOD system is designed to bring the power of a continuous system simulation language to a user at an interactive graphics console. Although primarily intended for biological modeling, BIOMOD can model a broad spectrum of continuous systems. Implementation of BIOMOD is still in process; we will attempt to distinguish between features that have actually been incorporated and those planned for future versions.

BIOMOD operates in two phases: a construction phase, in which the user draws and specifies models, and a simulation phase, which simulates the models and outputs results. Both phases are designed to provide a high degree of naturalness and rapid interactive response.

The construction phase is derived from GRAIL [1], a system that enables users to draw and execute program flow-charts. As in GRAIL, the BIOMOD user can construct a hierarchy of block diagrams to represent his model. The computational rules associated with the diagrams may be expressed in a variety of languages common to biological modeling.

The simulation phase utilizes the IBM System 360 Continuous System Modeling Program (CSMP) [2] to translate models into executable code and to compute the results. Data curves, representing selected output variables, may be displayed and modified as the simulation proceeds.

CSMP is a function-oriented language for simulating continuous systems. Statements in CSMP indicate either arithmetic operations or functional relationships between variables. The CSMP statement

\[ X = A \times \frac{Y}{Z} + (B-Y)^2 \]
is equivalent to the mathematical formula

\[ x = ay/z + (b-y)^2, \]

and the statement

\[ X = \text{INTGRL}(X_0,Y) \]

represents the relationship

\[ x(t) = x_0 + \int_0^t y \, dt. \]

CSMP automatically orders the user's statements so that inputs to functional blocks are computed before their outputs are evaluated. Thus, a user may enter statements in an arbitrary order. CSMP performs simulations by incrementing the independent variable (usually time) and computing the values of all other variables at each step.

THE ENVIRONMENT

The BIOMOD system operates on an IBM System 360 computer that has at least the capabilities of a model 40, using a partition of approximately 170,000 bytes. The operating system may be either the MFT II or MVT version of OS/360, augmented by Rand's Video Message Handler, a link to Rand's Video Graphics System [3].

A user generally communicates with BIOMOD by using a RAND Tablet [4] in conjunction with a television-like video graphics display, shown in Fig. 1; a text keyboard may also be used. The tablet was chosen as the primary input device because, in conjunction with appropriate software, it permits a user to enter information and perform manipulations directly and naturally without looking away from the display screen.
The screen displays information generated by the system in response to the user's pen actions.

Fig. 1--A User at the Console

The RAND Tablet

The RAND Tablet consists of a 10-in. square writing surface and a pen-like stylus. As the user moves the stylus near the tablet surface, a dot on the screen follows its motion; this direct feedback helps him position the stylus for pointing or drawing. When he presses the stylus against the writing surface, a switch in the stylus closes, notifying the computer that an action is beginning.

The stylus may be used to write, push, or drag. When writing, the user places the stylus on the tablet surface, moves it across the surface, and lifts it off as he would a pen on paper. As the stylus moves on the tablet, its track is displayed on the screen, as if it had ink. When the stylus is lifted, its switch opens and inking ceases.
The user's actions are analyzed by a set of symbol-recognition routines [5]. When a symbol is recognized, the ink is replaced by a hardware-generated symbol. (The recognized symbols are the uppercase English letters, the decimal digits, some punctuation marks, and several geometrical figures.) A symbol is changed by writing over it; it may be deleted with a scrubbing action.

*Pushing* is associated with a virtual button—a displayed area on the screen and a corresponding area on the tablet. To push a button, the user simply presses and releases the stylus in the button area. The program responds by taking the indicated action.

*Dragging* is accomplished by pressing the stylus in the area of a suitable display, then moving the stylus. The display follows the stylus as it moves. There is no ink associated with dragging.

**The Display Screen**

The display screen is divided into two or more distinct areas. The upper area always contains virtual buttons used to invoke the various system functions. The remainder of the screen is used in several ways, depending on the function being performed. In the lower portion of the screen, the user constructs his models and observes the results of his simulations.

**The Keyboard**

The keyboard resembles a conventional typewriter, with letters and numerals in the usual positions. A number of special characters common to computer usage are also provided. Although not required by the BIOMOD system, the keyboard occasionally proves useful for entering large amounts of text.
II. CONSTRUCTION PHASE

The BIOMOD construction phase is used to define the functions (computational procedures) comprising a model. Models may be represented by a hierarchy of block diagrams—that is, blocks within a diagram may represent other block diagrams. The specific functions represented by the lowest-level blocks may be expressed in a variety of languages. The system provides a number of primitive functions that may be used without further definition. The user can augment these by defining new functions of his choice.

We shall examine the capabilities of BIOMOD by using it to construct a model of water and solute distribution between the intravascular and extravascular spaces of the body. This example is considerably simplified in order to demonstrate the capabilities of BIOMOD, rather than to investigate physiological problems. The model is based on the "Starling Hypothesis," in which fluid and solute distribution across the capillary wall is determined by a balance between intravascular pressure and countervailing osmotic gradients. (We omit any consideration of Gibbs-Donnan gradients or varying distribution along the length of the capillary, and assume equa-osmolarity for the permeable species at steady states [6]).

The model (Fig. 2) consists of the conceptual compartments: red cell (R), plasma (P), interstitial (I), and intracellular (C). In each compartment, we wish to compute the volume, \(V_j\), and the solute concentration, \(S_j\), where \(j\) is the appropriate subscript for the compartment. The compartments are separated by membranes, but in this model only the vascular wall will support a pressure gradient.

BLOCK DIAGRAMS

We represent this BIOMOD model by drawing a block diagram, Fig. 3. The rectangles in the diagram are called function boxes, and represent computational procedures. The flow lines
Fig. 2--System of Conceptual Compartments

Fig. 3--Block Diagram of Compartmental Model
have no computational significance, but serve to clarify the diagram by indicating data flow. Ovals may be used to label flow lines.

In the block diagram, each of the four aligned boxes represents one of the conceptual compartments shown in Fig. 2. Each compartment contains water and two solute components, the permeable species and the large molecular impermeable species. The volume of each compartment is assumed equal to its water volume.

MULTILINGUAL DESIGN

The block diagram provides an overall view of our model. To define the PRESSURE function in detail, we push the definition button, (X), on PRESSURE's box. BIOMOD responds by displaying a list of languages that may be used to define functions (Fig. 4).

Fig. 4--Language Choice Page
Not all languages listed have been implemented. With the existing system, we can draw block diagrams, write CSMP statements, or copy previously constructed functions. Future versions of the system, providing chemical equations, differential equations, and data curves as alternatives, will enable the user to express his problem in the languages he finds most convenient. Statements in all languages will be automatically converted to CSMP statements by BIOMOD.

**CSMP STATEMENTS**

Pushing the appropriate button indicates that we wish to write CSMP statements. The screen displays a CSMP coding form on which we may write freely in two dimensions (Fig. 5).

![Fig. 5--A CSMP Coding Form](image)

The coding form represents a window into an unlimited expanse of text. To display hidden text, we can move this window either a line at a time via the +SCROLL+ buttons, or a page at a time via the +PAGE+ buttons. Convenient editing
features allow us to overwrite or erase characters, insert between characters (by using a caret), or close up a line (by erasing blanks). As we write statements on the form, BIOMOD checks them for syntactical errors. Errors are indicated by brightening the incorrect statements.

Let us assume that the pressure in the plasma compartment is incremented by an amount proportional to the single pulse given by the formulas

\[
P = \begin{cases} 
0, & \text{TIME} < T_1, \\
1, & T_1 \leq \text{TIME} \leq T_2, \\
0, & \text{TIME} > T_2.
\end{cases}
\]

A graph of \( P \) is given in Fig. 6. When BIOMOD is complete, we will be able to define \( P \) by entering this graph.

![Graph of Pressure Pulse in Plasma Compartment](image)

Fig. 6--Graph of Pressure Pulse in Plasma Compartment

At present, we can define \( P \) in CSMP notation as

\[
\text{TRIGR} = \text{FCNSW(ABS(TIME-T_1)} - \text{DELT/2.,1.,1.,0.})
\]

\[
P = \text{PULSE(T2-T_1,TRIGR)}
\]
which we write directly on the coding form (Fig. 7). FCNSW, ABS, and PULSE are functions provided by CSMP, and DELT is the numerical integration step size. The above equations completely define PRESSURE. Pushing the button labeled START OF MODEL causes the block diagram to be redisplayed.

Fig. 7 -- CSMP Coding Form to Define PRESSURE

USER-DEFINED FUNCTIONS

We now specify the calculations associated with each compartment. With a little forethought we can construct two general compartment functions, one for the red cell and intracellular compartments, the other for the plasma and interstitial compartments. In CSMP, this facility is provided by MACROS; the BIOMOD equivalent is called a user-defined function.

To construct a user-defined function, we first select a name and write it on the top line of a box. In this case we select CMPT1 for one type of compartment, and enter it
on the RED CELL box. Because this function is not yet defined, pressing the definition button produces the list of definition languages. This time we select block diagrams. Once we indicate our choice, the screen displays a blank page on which to draw.

HIERARCHY

We have just used an important feature of the BIOMOD system—the capacity for hierarchical structuring. A function box, whether named or not, may be further defined by another block diagram, consisting of more boxes, each of which is represented in turn by still more boxes. The nesting of definitions may be continued to arbitrary depth. This feature allows the user to view his model from various levels of detail, without being bothered by unnecessary information.

We now draw the block diagram for CMPTl (Fig. 8). One box is used to compute the change in water volume; the other is used to integrate this result to produce the total water volume.

PRIMITIVE FUNCTIONS

All functions that CSMP provides (e.g., FCNSW, ABS, and PULSE, used above) are considered primitive to BIOMOD; they may be used without definition. We may use the primitive function, INTGRL, to compute the volume from the change in volume.

The first step in using this primitive function is to print the name INTGRL on the top line of the COMPUTE VOLUME box. Since this box has an implicit definition, pressing its definition button results in a list of the arguments of INTGRL, shown in Fig. 9.

In our case, the output is VOL1; we call the initial volume VOL10, and the input is VOLDOT. We print these
Fig. 8--Block Diagram for CMPT1

Fig. 9--Argument List for INTGRL Function
names, as in Fig. 10. We now return to the block diagram for CMPT1 and press the definition button for the COMPUTE VOLDOT box. The list of definition languages appears and we select CSMP. In defining the COMPUTE VOLDOT box, we refer to Fig. 3.

Fig. 10--Formal and Local Arguments for COMPUTE VOLUME

The rate of change in volume is equal to the water movement, which is proportional to the osmotic gradient given by the solute concentrations in adjacent compartments. Thus, for red cells

\[ \dot{V}_R = k_1 \cdot (S_R - S_p) , \]

where \( k_1 \) is a diffusion constant, a parameter determined by the permeability of the membrane to water. \( S_R \) and \( S_p \) are the concentrations of solute in red cells and plasma, respectively. For red cells
\[ S_R = V_{R0} \cdot S_{R0}/V_R , \]

where \( V_{R0} \) and \( S_{R0} \) are initial values.

For the intracellular compartment, the corresponding equations are

\[ . \quad V_C = k_3 \cdot (S_C - S_I) \]

and

\[ S_C = V_{C0} \cdot S_{C0}/V_C . \]

In continuing to define a general CMPT1 function, we generalize these two equations, and print the CSMP statements

\[ VOLDOT = KK1 \times (S1 - S2) \]

\[ S1 = VOL10 \times S10/VOL1 \]

on the coding form, Fig. 11.

\section*{DATA PAGE}

Because we are creating CMPT1 as a general-purpose compartment function to use for both the red cell and intracellular compartments, we must indicate which variables are inputs or outputs and which only appear locally as part of the definition. We shall later designate correspondences between CMPT1's inputs and outputs (i.e., its arguments) and variables in the two compartments that CMPT1 defines.

We push the DATA button, and BIOMOD displays the names of all variables in CMPT1; these are listed in two columns, one for defined variables, the other for those used but not defined. (A variable is defined if it appears on the left side of a CSMP assignment statement, or if it is an output of a function.)
Fig. 11--CSMP Coding Form to Compute VOLDOT

Each column provides space for marking the attributes of the listed variables. The variables in the defined list may represent formal output arguments, or they may be simply local to the function. (Local variables cannot be referenced outside the function.) In Fig. 12, we marked VOL1 and S1 as formal outputs by pushing the FORMAL OUTPUT column adjacent to these names. We marked VOLDOT local.

Variables used but not defined within a function must be defined on some higher level. Some of these represent different values each time the function is used; these should be identified as formal input arguments. Other variables always represent the same data whenever the function is used; these we call external variables because their names within the function are the same as their names on its exterior. In this case, we mark all the "defined elsewhere" variables as formal input arguments.
A second user-defined function

Similarly, we create a general compartment function, CMPT2, that can be used for both the plasma compartment and the interstitial compartment. Thus, for the plasma compartment

\[
\dot{V}_P = k_1 \cdot (S_P - S_R) + k_2 \cdot (S_P - S_I) + k_H \cdot (P_I - P_P) + \frac{k_X}{V_P},
\]

where \(k_H\) is a diffusion parameter relating \(V_P\) to the pressure gradient, and \(k_X\) is equal to an osmotic coefficient times the amount of the impermeable species. \(\dot{V}_P\) is proportional to the concentration of the fixed species and hence inversely proportional to \(V_P\). The corresponding equation for the interstitial compartment is

\[
\dot{V}_I = k_3 \cdot (S_I - S_C) + k_2 \cdot (S_I - S_P) + k_H \cdot (P_P - P_I) - \frac{k_X}{V_P}.
\]
Generalizing this equation, we write

\[ \text{VOLDOT} = K_1(S_2 - S_1) + K_2(S_2 - S_3) + KH(P_I - P_P) + \underbrace{\text{SIGNKX} \times KX/VP} \]

and

\[ \text{VOL} = \text{INTGRL(VOL0,VOLDOT)} \]

where \text{SIGNKX} will be assigned the value +1 for the plasma compartment but -1 for the interstitial compartment.

We also need to define the pressures. This can be done for the plasma compartment as

\[ P_P = P_{P0} + \alpha \times (V_P - V_{P0}) + \alpha \times P/k_H, \]

for the interstitial compartment as

\[ P_I = P_{I0} + \beta \times (V_I - V_{I0}), \]

and in general as

\[ P_P = P_{P0} + \text{ALPHA} \times (\text{VOL} - \text{VOL0}) + C \times P/kH. \]

\( P \) is the pressure pulse computed in the PRESSURE process box, \( P_{P0} \) is the initial steady-state pressure, \( \text{ALPHA} \) is a proportionality constant, and \( C \) adjusts the height of the pulse.

Finally, we need to define the rate of change of permeable solute concentration in this generalized compartment. If the rate of solute movement is proportional to the solute gradient,

\[ \frac{d(S \cdot V)}{dt} = k_S \cdot \Delta S, \]
we may expand and rearrange to get

\[
\dot{S}_P = \left(k_S \cdot (S_I - S_P) - S_P \cdot \dot{V}_P \right) / V_P,
\]

\[
\dot{S}_I = \left(k_S \cdot (S_P - S_I) - S_I \cdot \dot{V}_I \right) / V_I.
\]

Generalizing we have

\[
S2\text{DOT} = (K_S \cdot (S3 - S2) - S2 \cdot \text{VOLDOT}) / \text{VOL}
\]

and integrating

\[
S2 = \text{INTGR}(S20, S2\text{DOT}).
\]

We print the CSMP statements on the coding form for CMPT2, shown in Fig. 13. When we defined CMPT1, we chose to functionally separate the computation of the volume and its derivative. However, here we wrote all the equations on a single form. This is purely a matter of convenience.

Figure 14 shows the data page for CMPT2. Here we marked $K_S$, $P$, $V_P$, $K_X$, $K_H$, and $K_2$ as external names because these variables have the same meanings in both compartments.

**USING USER-DEFINED FUNCTIONS**

The definition of compartments being completed, we return to the overall block diagram by pushing the START OF MODEL button. The RED CELL function box now bears the name CMPT1. When we press its definition button this time, its argument list appears. Note that this is the same response that we get with primitive functions. The names that appear on the argument list are those we marked as formal when CMPT1 was defined. We specify all of the arguments (Fig. 15) and then return to the block diagram.

Next we print the name CMPT1 on the INTRAC function box and push its definition button, labeling its arguments in the same way (Fig. 16).
Fig. 13--CSMP Coding Form to Define CMPT2

Fig. 14--Data Page for CMPT2
Fig. 15--Formal and Local Arguments for RED CELL

<table>
<thead>
<tr>
<th>FORMAL</th>
<th>LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL1</td>
<td>VL</td>
</tr>
<tr>
<td>SL</td>
<td>SH</td>
</tr>
<tr>
<td>VOL10</td>
<td>VRO</td>
</tr>
<tr>
<td>SK1</td>
<td>K1</td>
</tr>
<tr>
<td>SP</td>
<td>SI</td>
</tr>
<tr>
<td>ST0</td>
<td>SRO</td>
</tr>
</tbody>
</table>

Fig. 16--Formal and Local Arguments for INTRAC

<table>
<thead>
<tr>
<th>FORMAL</th>
<th>LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL1</td>
<td>VL</td>
</tr>
<tr>
<td>SL</td>
<td>SH</td>
</tr>
<tr>
<td>VOL10</td>
<td>VLO</td>
</tr>
<tr>
<td>SK1</td>
<td>K1</td>
</tr>
<tr>
<td>SP</td>
<td>SI</td>
</tr>
<tr>
<td>ST0</td>
<td>SLO</td>
</tr>
</tbody>
</table>
We similarly specify the arguments for PLASMA (already labeled CMPT2) (Fig. 17). We print the name CMPT2 on the INTERS box and label its arguments (Fig. 18). In the lists for CMPT2, BIOMOD automatically fills in KS, K2, KH, KX, P, and VP in the right-hand column because these were marked EXTERNAL NAME on the data page.

![Table of Formal and Local Arguments](image)

**Fig. 17--Formal and Local Arguments for PLASMA**

**MODEL DATA PAGE**

The construction of our model is nearly complete. We need only specify the values to be associated with the yet undefined variables. For this purpose, we return to the block diagram and press the DATA button. This generates a list of all variables used in the model as a whole, except those that are internal to user-defined functions. As with user-defined functions, this list is divided into two columns labeled VARIABLES (defined) and PARAMETERS (undefined) (Fig. 19).
Fig. 18--Formal and Local Arguments for INTERS

Fig. 19--Data Page for Overall Model
We complete the definition by specifying the values of
the parameters, paying strict attention to their units. By
checking columns, we also indicate whether these quantities
may be modified during simulation. We chose K1, K2, K3, KS,
KH, KX, and the initial volumes and solute concentrations as
the modifiable parameters of the simulation.

From the list of VARIABLES, we select those whose values
we may want to plot during simulation. (Storage constraints
prohibit the plotting of all the variables of a large model;
the user selects the variables he deems most important.) In
our model, we choose P, VP, VI, SP, and SI as candidates for
plotting.
III. SIMULATION PHASE

We simulate our completed model by pushing the SIMULATE button. At this time, BIOMOD translates its internal data into CSMP statements and passes control to CSMP. Included in the CSMP statements are special subroutine calls that enable us to exercise interactive control over the simulation.

SIMULATION DISPLAY

A standard display format (Fig. 20) is used throughout the simulation phase. This display is divided into several distinct areas. As usual, the upper region contains virtual buttons for invoking system functions. The remainder of the screen is normally used to display and control the simulation output.

Fig. 20--The Simulation Display
Plotting Region

The large, square region near the center of the screen is used to plot curves representing selected variables. Curves are plotted as connected line segments with a two-character identifier near the right-most plotted point. In certain cases described below, this screen area may also serve different purposes.

Y-NAME Box

The next largest region on the screen appears as a box, to the left of the plotting region, that contains three columns, Y-NAME, RUN, and ID. In the Y-NAME column, the user prints the names of variables to be displayed on the Y (vertical) axis. (Each name must be one of the variables marked PLOTTABLE on the model data page.) The system selects the first two characters of the name as the plotting identifier and enters them in the ID column. The user may overrule this choice by printing a different ID. The RUN column is used to distinguish among several runs of the same model; this feature is not supported in the current system.

X-NAME Box

The box preceded by "X-NAME =" contains the name of the variable to be displayed on the X (horizontal) axis. The system initially selects TIME for this purpose, but the user may substitute any other plottable variable.

Range Boxes

Four range boxes appear in the display, positioned at each end of the X and Y axes. These boxes indicate the current ranges of the axes and allow the user to modify those ranges. Each box contains a number, in scientific notation (or "E-Format"), with an upward and a downward
arrow. When the user prints over the existing number, the curves are immediately rescaled according to his specifications. He may also push the upward (or downward) arrow to increase (decrease) the range value. When he does, the number in the box is continually incremented (decremented) and the curves are continually rescaled until he lifts the stylus.

Linear-Log Buttons

One box on each axis contains either the word LINEAR or the word LOG, which indicates whether the scales for the associated axes are linear or logarithmic. The boxes also serve as buttons for changing to the other mode. When one of the buttons is pushed, the word is changed and the curves are redisplayed in the other mode, the scale boxes being used to determine the range.

Plotting Interval

The box preceded by "MIN PLOTTED ΔX =" indicates the plotting interval. Storage requirements limit the number of points that can be displayed at one time. By increasing the plotting interval, the user may view a larger portion of an output curve (but with reduced accuracy). Increasing the plotting interval does not destroy the undisplayed data; the interval may be subsequently reduced.

Current X

The value displayed after "NOW X =" indicates the current value of the X-axis variable (usually TIME). This portion of the display is updated once each computation iteration, even if curves are not redisplayed because of a large plotting interval. This serves as a positive indication that the system is running during very complex, slow simulations.
Message Area

Above the plotting region is a space reserved for messages to the user to inform him of system status ("SIMULATION RUNNING," "SIMULATION STOPPED"), to report errors ("X VALUES MUST BE POSITIVE"), and to provide other information that may be helpful.

INTEGRATION PARAMETERS

Before beginning the simulation, the system pauses and displays, in the plotting region of the screen, information regarding the integration technique it has initially selected (Fig. 21). This information includes the integration method, the integration interval, and other data, such as the relative and absolute error tolerances, depending on which integration method is chosen. The user accepts or alters this information as he sees fit. For our compartmental model, we select the Runge-Kutta variable step-size method, with an initial interval of 0.005 sec, and a minimum interval of 0.0001 sec. We leave the other values as chosen by BIOMOD. After we have indicated these changes, we press the PLOT and RESTART buttons.

SPECIFYING Y-NAME VARIABLES

The message area now displays the message "SIMULATION RUNNING," but no curves appear because we have not printed anything in the Y-NAME box. To correct this oversight, we press the STOP button, and the "SIMULATION STOPPED" message appears. We now print VP in the Y-NAME box, and the two characters VP appear in the ID column. Simultaneously, the plotting region displays the portion of the VP curve that has been computed thus far (Fig. 22).
Fig. 21--The Integration Method Display

Fig. 22--Plot of VP versus TIME
RESCALING

Because some points of the VP curve seem to be coincident with the upper border of the plotting region, we change scales. We change the number in the upper range box for the Y-axis from 1 to 1.2 by overwriting, which moves the curve down into the plotting region. We change the number in the lower range box for Y by pushing the upward arrow until the curve is approximately centered. The rescaled curve is shown in Fig. 23.

We want to view VP over a period of about three simulated seconds. We therefore change the number in the right-most range box for the X-axis to 3 (3.00E+00) and push the CONTINUE button in order to resume the simulation.

![Rescaled Plot of VP versus TIME](image)

**Fig. 23**--Rescaled Plot of VP versus TIME

THINNING DATA

The VP curve is continually updated as new values are computed. After a short while, the simulation pauses,
displaying the message "THIN THE DATA," indicating that we have generated more data points than the system can plot. We therefore change the MIN PLOTTED AX value to separate plotted points by at least 0.06 sec of simulated time.

TROUBLE SHOOTING

When the curve reaches the right-most border of the plotting region, TIME has exceeded three seconds so we push the STOP button. We notice (Fig. 24) that the plasma volume has dropped less than expected, possibly because the plasma pressure pulse was too small. To verify this, we print P in the Y-NAME box and examine its curve (Fig. 25). We observe that the pressure has indeed pulsed appropriately; we must therefore look elsewhere. We delete the curve for P by scrubbing its name from the Y-NAME box.

![Simulation stopped diagram](image)

Fig. 24--Complete Plot of VP versus TIME
Fig. 25--Plots of VP and P versus TIME

CHANGING PARAMETERS

When we push the PARAMETERS button, the curves disappear and the plotting region displays a list of the simulation parameters we identified on the model data page. We see that KS, the diffusion constant for solutes, is set to 80, equal to one-tenth that for water. We decide to increase this value to 800 (Fig. 26), then push the PLOT and RESTART buttons. This time the curve for VP more nearly matches our expectations.

We can determine the values of VP and TIME at the minimum, or anywhere else, simply by pushing the curve. The system then establishes an X-Y meter (Fig. 27), which may be dragged anywhere in the plotting region. In addition to examining variables graphically, we can push the VARIABLES button to obtain a table of the current, minimum, and maximum values of the plottable variables (Fig. 28). In this manner
we continue to change parameters and study their effects until satisfied. If necessary, we can return to the construction phase to alter the model structure by pushing the MODEL button.

Fig. 26--The Parameter Display
Fig. 27--Plot of VP versus TIME with KS=800 Shown with a Meter

Fig. 28--The Variables Display
IV. CONCLUSION

Other modeling systems [7,8,9] provide convenient graphical interfaces to simulation languages. Some of these additionally include important analytical facilities. BIOMOD is distinct, however, in that it provides direct and natural interaction, unlimited hierarchy, and several descriptive languages. The interaction minimizes operational mechanics, the hierarchy facilitates devising complex models a portion at a time, and the multilingual approach obviates the requirement that the user translate his model into a conventional simulation language.

The present version of BIOMOD and the other interactive modeling systems have several advantages over batch programs. In the construction phase, they check the modeler's work as he proceeds, and they automate the tedious, error-prone program setup procedure. However, the model description is often very similar to that required by batch programs. The major advantage of interactive systems presently is in the simulation phase. Here the modeler may halt uninteresting simulation runs, investigate his model by changing parameters and immediately rerunning, and conveniently observe and compare simulation results.

The addition of other descriptive languages to BIOMOD will significantly improve the construction phase. A modeler who can write chemical and differential equations and draw graphical functions will feel much closer to his problem, and therefore will work more effectively than at present.
REFERENCES


