SUMMARY

BIOMOD is an operational system specifically designed to make it convenient for unsophisticated computer users to study models of dynamic systems. The system features a high degree of interaction, user-oriented model-definition languages, and flexible, in-depth model structuring. It employs a graphics console comprising a television screen, a data tablet, and a keyboard.

A BIOMOD user may represent a model by drawing block diagrams and handprinting or typing text; he receives immediate feedback about the system's interpretation of his actions. Each component of a model block diagram may be defined by another block diagram; this facilitates organizing models into meaningful substructures. A user ultimately defines each component block by analog-computer-like elements, algebraic, differential, or chemical equations, and/or Fortran statements. A modeler may thus define his model in whatever terminology is meaningful to him. Displayed curves are continually and automatically updated during model simulation. A user may stop the simulation and plot different variables, change scales, or alter parameter values, and then either continue simulating or revise the description of his model.

This report demonstrates BIOMOD by presenting a scenario of how a user might describe and simulate a drug-effect model, briefly describes the system implementation, and discusses experience with users.

II. A MODEL FOR EVALUATING DRUG ADMINISTRATION POLICIES

Medications and their prescribed dosages are designed to maintain a critical amount of drug in the blood for a specified period of time. The conventional method for determining optimal dosages involves numerous laboratory experiments. If the drug effects can be modeled, however, a more efficient method is to experiment by running computer simulations.

One technique for maintaining the prescribed amount of drug is to use a capsule containing a large number of differently coated pellets. The pellets dissolve at different times, so that, as drug leaves the blood, it is replaced by drug released by newly dissolved pellets. Garrett and Lambert [8] have proposed the following model to describe this situation. A capsule comprises a number of pellet populations with different mean times of release. The rate of drug release for each population is assumed to be normally distributed (with the same standard deviation for each population) about the mean time of release for that population. The rate of adding drug to the body is therefore specified by a sum of normal distributions. The transfer of drug through the body is described by

$$\text{drug} \longrightarrow \text{GI} \xrightarrow{k_{GI,B}} \text{B} \xrightarrow{k_{B,U}} \text{U}$$

where GI refers to the gastrointestinal tract, B to the blood, and U to the urine and other excretory parts of the body; this means that drug flows from GI to B at rate $k_{GI,B}$ and from B to U at rate $k_{B,U}$. Thus, for example,

$$dD_B/dt = k_{GI,B}\, D_{GI} - k_{B,U}\, D_B$$

where D_B is the amount of drug in B, and D_{GI} the amount of drug in GI.

Given this description together with the requisite parameter values, BIOMOD can be used to simulate the model. Note that the following dialogue describes only one of many possible ways of reaching the same goal, and that BIOMOD does not force the user to take action in

When using BIOMOD, we communicate via a data tablet pen and a keyboard. The pen's location on the tablet is always indicated by a dot displayed in the corresponding location on the television screen. BIOMOD's interpretation of user pen actions depends on where the pen is placed and on what is currently displayed on the screen. We may handprint characters in most areas. As we write, a displayed "ink" track appears to flow from the pen; each time we complete a character, its track is replaced by a stylized character. We can change a character by writing another over it. Some symbols are used for editing; for example, we may use a caret to insert text, or we may scrub with the pen to delete text. Some areas displayed on the screen act as pushbuttons; if we "push" one of these (by touching the pen down), the system performs the indicated action. If we push a displayed arrow, a continuous action takes place, such as the rescaling of a set of curves. Some figures can be "dragged"; if we "touch" one of these and move the pen, the displayed figure follows the pen's motion. We may type (with the keyboard) in any area where writing is possible. The keyboard cursor may be positioned either with the pen or with keyboard control keys.

To create our model, we enter our identification, name our model DRUGS, then begin constructing the model. Because it has two major components, we first draw two rectangles; these are replaced by stylized function boxes. We write CAPSULE in one box and names of parts of the body in the other box, and then draw a flowline to connect them. This diagram (Fig. 2) provides not only a picture of our model, but also a means of defining the two components of the model separately.

To define the capsule component, we first push the DEFN button on its box. The system replaces the block diagram with a list of languages that we may choose from to define the component. The languages are block diagrams, mathematical equations, chemical equations, and Fortran statements. We choose block diagrams so that we can define the capsule as a set of boxes, each representing a pellet population. This ability to define a box by another block diagram enables us to organize a model as a hierarchical collection of a number of components at different levels. We draw four boxes and write PILL on the top line

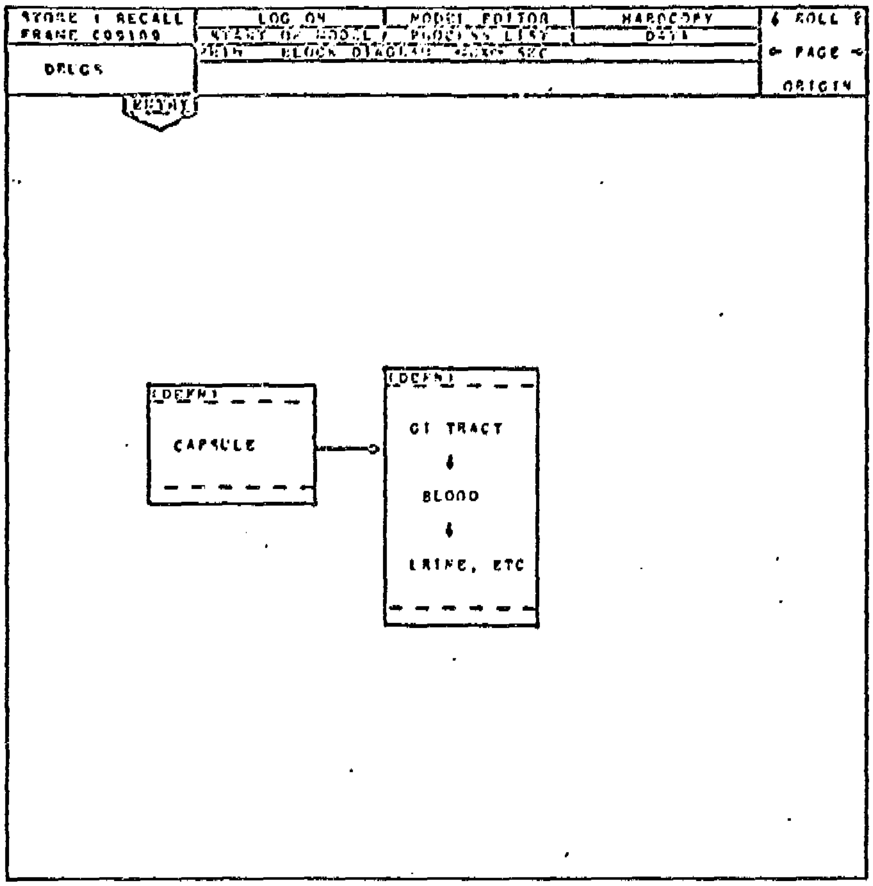


Fig. 2—The DRUGS model block diagram

of each to represent four pellet populations. We name a function box in this way whenever we anticipate using the same function repeatedly.

Because each population is defined by a normal distribution, we indicate that we want to define the PILL function with mathematical equations. BIOMOD responds by displaying a form for writing algebraic and differential equations. We assume that the probability of a pellet dissolving in an interval about time t is given by the probability density function

$$p = \frac{1}{\sigma\sqrt{2\pi}}\ e^{-(t-m)^2/2\sigma^2}$$

Using this function to approximate the drug release rate by a deterministic variable, we write

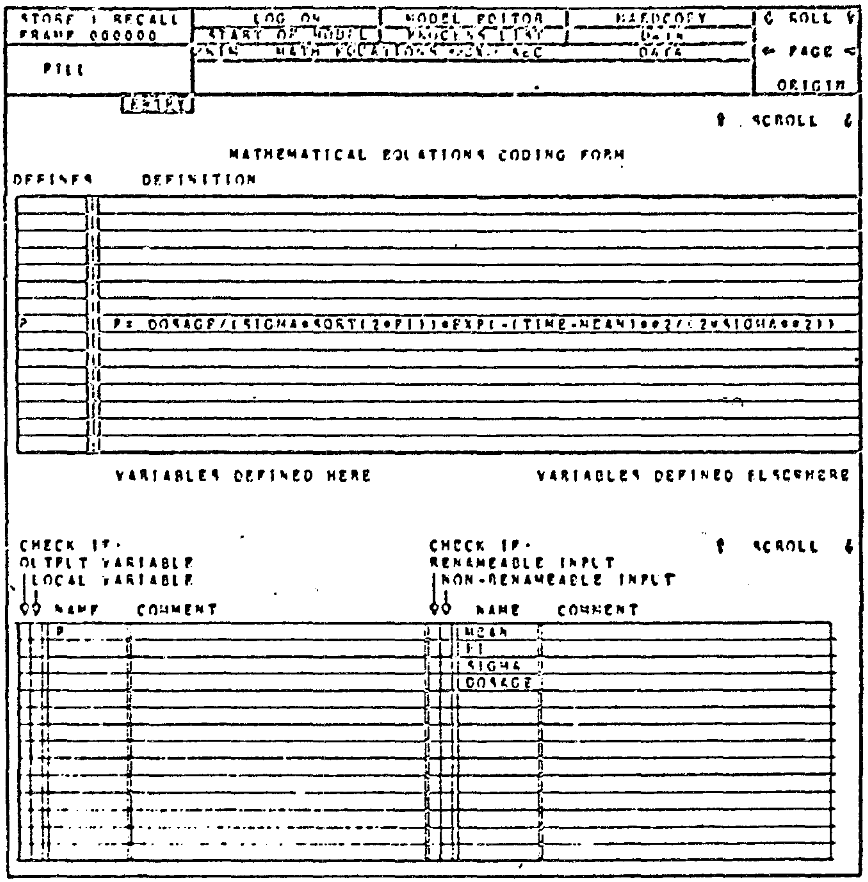
```
P = 1/(SIGMA*SQRT(2*PI))*EXP(-(TIME-MEAN)**2/(2*SIGMA**2)
```

The system analyzes this statement and immediately responds with the message

```
UNBALANCED PARENTHESES
```

We then add a closing parenthesis to correct the statement. We also realize that we should parameterize the amount of drug released by each pellet population, so we insert DOSAGE after the equals sign and scrub the 1. The display now appears as in Fig. 3.

BIOMOD has generated separate lists of the defined and undefined variables; TIME does not appear because it is always the simulation independent variable. These lists enable us to indicate which variables have different meanings or values each time we use the function. We indicate that the names (and therefore the values) of PI, SIGMA, and DOSAGE are the same each time we use the PILL function. This is because PI is a constant, and because we assume that the standard deviation and dosage amount are the same for each population. On the other hand, we indicate that MEAN may have a different value for each pellet population.



STORE / RECALL
FRAME 000000

PILL

LOG ON	MODEL EDITOR	HARDCOPY	← ROLL →
START OF MODEL	PROCESS LIST	DATA	
EXIT MATH EQUATIONS [illegible] SEC		DATA	← PAGE →
			ORIGIN

ENTRY

↑ SCROLL ↓

MATHEMATICAL EQUATIONS CODING FORM

DEFINES	DEFINITION
P	P= DOSAGE/(SIGMA*SQRT(2*PI))*EXP(-(TIME-MEAN)**2/(2*SIGMA**2))

VARIABLES DEFINED HERE

CHECK IF:
OUTPUT VARIABLE
LOCAL VARIABLE

NAME	COMMENT
P	

VARIABLES DEFINED ELSEWHERE

↑ SCROLL ↓

CHECK IF:
RENAMEABLE INPUT
NON-RENAMEABLE INPUT

NAME	COMMENT
MEAN	
PI	
SIGMA	
DOSAGE	

Fig.3—The definition of a pellet population

Now that we have defined the PILL function, we are ready to use it to define the individual populations. We push a button to get back to our diagram of the four PILL boxes, then push the DEFN button on one of these. Because PILL is now defined, BIOMOD displays

```
P      →
MEAN   ←
PI     ← PI
SIGMA  ← SIGMA
DOSAGE ← DOSAGE
```

for us to provide the names of the output and mean of this particular population. We write P1 next to P →, to name this output P1, and write M1 next to MEAN ←, to name this mean time of release M1. We similarly establish the correspondence between the names of variables in the other three pellet populations and the names (P and MEAN) used when defining the function PILL.

We can describe the flow of the drug through the body by chemical equations because these are mathematically equivalent to mass transport equations. When we push the DEFN button on the box that describes the body, and select chemical equations, BIOMOD presents an appropriate form. According to our original model description, we would like to write

$$D_{GI} \xrightarrow{k_{GI,B}} D_B \xrightarrow{k_{B,U}} D_U$$

or

$$D_{GI} \underset{0.}{\overset{k_{GI,B}}{\rightleftarrows}} D_B$$

$$D_B \underset{0.}{\overset{k_{B,U}}{\rightleftarrows}} D_U$$

where 0. indicates that there is no backward flow. BIOMOD requires that we linearize each equation, and write the rate coefficients and equations

```
S      KGIB      0.      DGI = DB

S      KBU       0.      DB  = DU
```

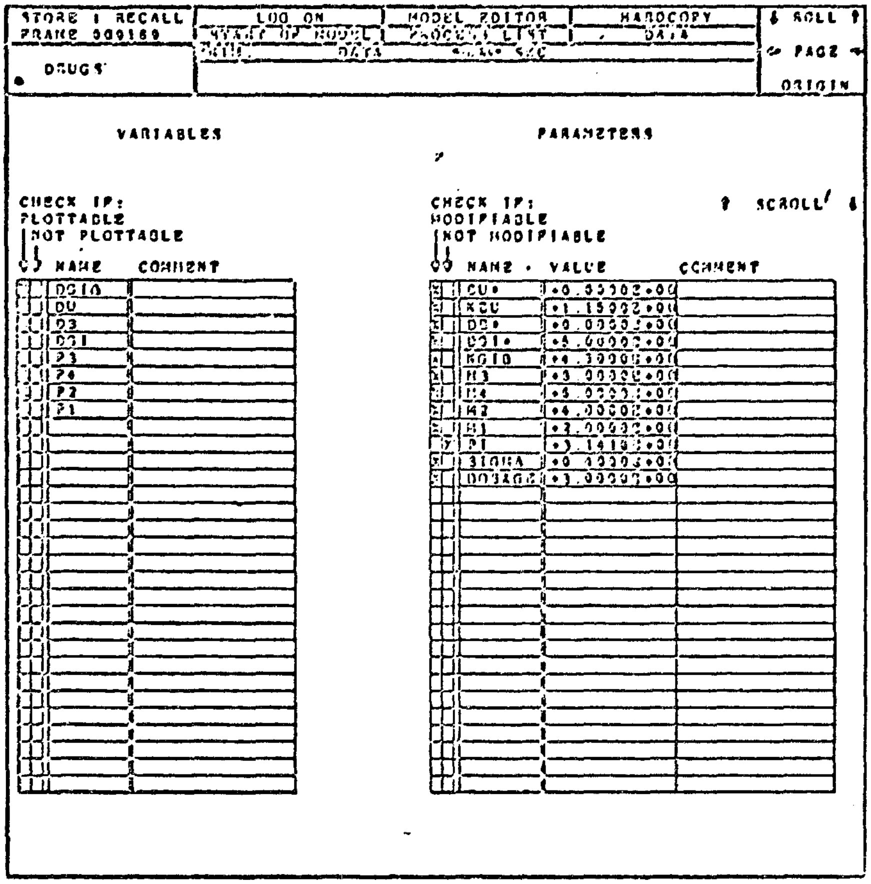
Here S (for slow reaction) means that BIOMOD should derive integral equations from our equation. Since the pellets release drug into the gastrointestinal tract, we also write

```
G      DGI               P1+P2+P3+P4
```

This statement (with G for gain) indicates that the gain of DGI, i.e., the increased rate of change of DGI due to drug entering the body from outside, is equal to the sum of the rates of drug release from the four pellet populations.

The model is now defined except for parameter values. When we indicate that we are ready to provide these values, BIOMOD displays the names of model variables and parameters in two separate lists (Fig. 4). Names such as DGI° indicate initial values; they are derived from the chemical equations by BIOMOD. We enter the values given or implied by Garrett and Lambert. We assume that some drug is immediately released into the gastrointestinal tract and therefore set DGIo to 5; the other drug amounts are initially zero. In order to minimize storage requirements, BIOMOD limits the number of variables whose values are saved during simulation and the number of parameters whose values can be modified during simulation. Since this model is small, we indicate that we want to save values of (and possibly plot) all the variables, and that we might want to modify values of all parameters except PI.

It has taken us less than half an hour to completely describe our model. We now indicate that we would like to simulate it. BIOMOD first produces a CSMP/360 [2] program that describes our model and provides for graphic display of the results. CSMP, in turn, generates a Fortran program, which is compiled and linked with other programs required to run the simulation. If an error is detected at one of these steps, program listings and error messages are displayed on the screen; otherwise, our only awareness of the intermediate steps is via displayed



VARIABLES

CHECK IF:
PLOTTABLE
NOT PLOTTABLE

		NAME	COMMENT
		DG10	
		DU	
		D3	
		DG1	
		P3	
		P4	
		P2	
		P1	

PARAMETERS

CHECK IF:
MODIFIABLE
NOT MODIFIABLE

↑ SCROLL ↓

		NAME ·	VALUE	COMMENT
X		CU*	+0.33302+00	
X		KCU	+1.15002+00	
X		DC*	+0.00003+00	
X		DC1*	+5.00002+04	
A		KG10	+4.30005+00	
X		K3	+3.00005+00	
X		K4	+5.00003+00	
X		K2	+4.00002+00	
X		K1	+2.00002+00	
	Y	PI	+3.14159+00	
X		SIGMA	+0.40003+00	
X		DOSAGE	+1.00003+00	

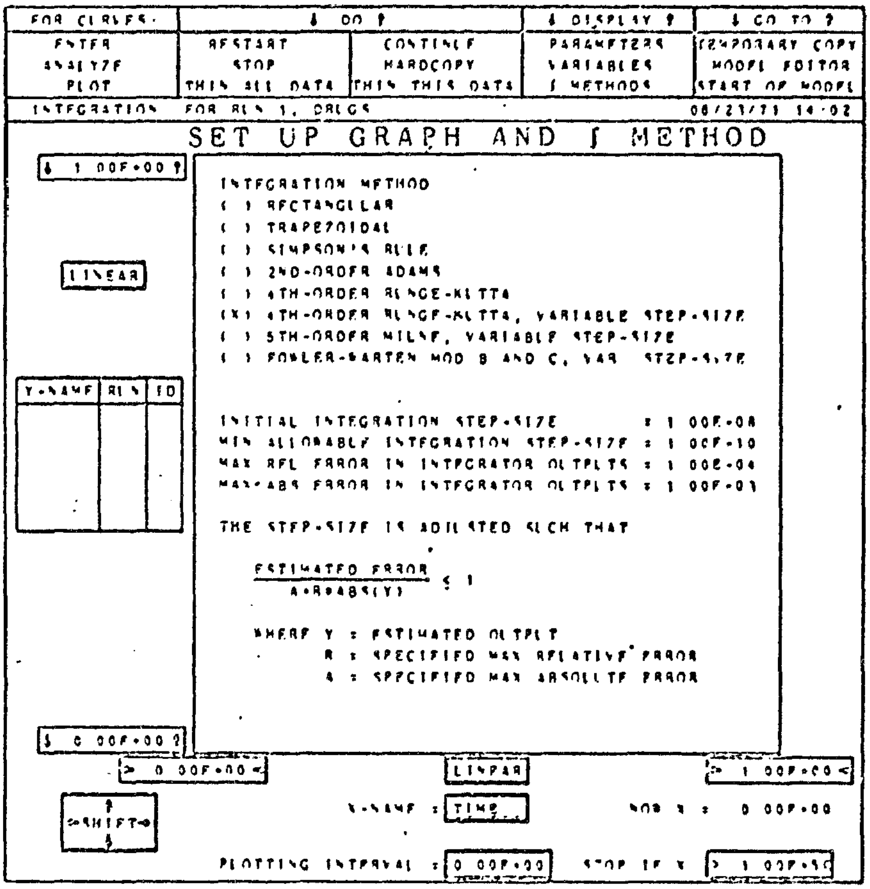
Fig. 4—The model variables and parameters

messages. The time required for the translation depends on the load on the (multiprogrammed) computer; generally it is about three minutes.

Our DRUGS model translates successfully, so the form shown in Fig. 5 is displayed on the screen. As in the other forms, software pushbuttons appear across the top. The central area is for selecting numerical integration methods, modifying parameter values, examining variable names and values, or plotting graphs. The areas to the left and below the central area are for specifying the y and x axes of the graphs.

We expect the values of our model variables to change smoothly and over several units to TIME, so we choose a simple integration method--Simpson's method with step-size = 0.1. This is a fixed step-size method, so the information regarding variable step-sizes disappears. Before studying how to use a multi-pellet capsule, we want to ensure proper model behavior when there is initially some drug in the gastrointestinal tract, but no capsule. To eliminate the capsule drug we push the PARAMETERS button to display the list of modifiable parameters in the central area, then overwrite the value of DOSAGE, changing it to 0. Next we display the list of plottable variables. Because we are most interested in the amount of drug in the gastrointestinal tract, blood, and urine, we drag the names DGI, DB, and DU to the y axis. We want to watch the model for several simulated hours, so we change the upper range of TIME (in the small box at the lower right of the central area) from 1. to 7. We push PLOT; now we are ready to plot DGI, DB, and DU from 0. to 1. against TIME from 0. to 7. hours.

We push RESTART and the simulation begins running. We see (from the curves) that DB, and later DU, are being generated; the "NOW X = " number changes continuously to indicate the current value of simulated TIME. Because DGI is plotted off scale along the upper boundary, we touch the pen down to stop the simulation. In order to determine the range of DGI, we return to the display that lists the variables along with their current, minimum, and maximum values. The maximum value of DGI is 5. (its initial value), so we write 5 over the 1 that specifies the upper y-axis value, then redisplay the curves. The curves are now nicely scaled. We continue the simulation, then stop it when we see that nearly all the drug has entered the urine, and values are changing



```
FOR CURVES:  |          ↓ DO ↑                  | ↓ DISPLAY ↑  |   ↓ GO TO ↑
   ENTER     |    RESTART    |    CONTINUE      |  PARAMETERS  | TEMPORARY COPY
  ANALYZE    |     STOP      |    HARDCOPY      |  VARIABLES   |  MODEL EDITOR
   PLOT      | THIS ALL DATA | THIS THIS DATA   |  ∫ METHODS   | START OF MODEL
 INTEGRATION  FOR RUN 1, DRUGS                                   08/23/73 14:02
                  SET UP GRAPH AND ∫ METHOD
 ↓ 1.00E+00 ↑
                 INTEGRATION METHOD
                 ( ) RECTANGULAR
                 ( ) TRAPEZOIDAL
                 ( ) SIMPSON'S RULE
   LINEAR        ( ) 2ND-ORDER ADAMS
                 ( ) 4TH-ORDER RUNGE-KUTTA
                 (X) 4TH-ORDER RUNGE-KUTTA, VARIABLE STEP-SIZE
                 ( ) 5TH-ORDER MILNE, VARIABLE STEP-SIZE
                 ( ) FOWLER-WARTEN MOD B AND C, VAR  STEP-SIZE

 Y-NAME|RUN|ID
       |   |         INITIAL INTEGRATION STEP-SIZE        = 1.00E-08
       |   |         MIN ALLOWABLE INTEGRATION STEP-SIZE  = 1.00E-10
       |   |         MAX REL ERROR IN INTEGRATOR OUTPUTS  = 1.00E-04
       |   |         MAX ABS ERROR IN INTEGRATOR OUTPUTS  = 1.00E-03

                 THE STEP-SIZE IS ADJUSTED SUCH THAT

                     ESTIMATED ERROR
                     ---------------  ≤ 1
                       A+R*ABS(Y)

                     WHERE Y = ESTIMATED OUTPUT
                           R = SPECIFIED MAX RELATIVE ERROR
                           A = SPECIFIED MAX ABSOLUTE ERROR

 ↓ 0.00E+00 ↑
      > 0.00E+00 <              LINEAR                      > 1.00E+00 <
      ↑
   ←SHIFT→               X-NAME = TIME              NOW X =   0.00E+00
      ↓
             PLOTTING INTERVAL = 0.00E+00      STOP IF X > 1.00E+50
```

Fig.5—The simulation control form with Integration Methods

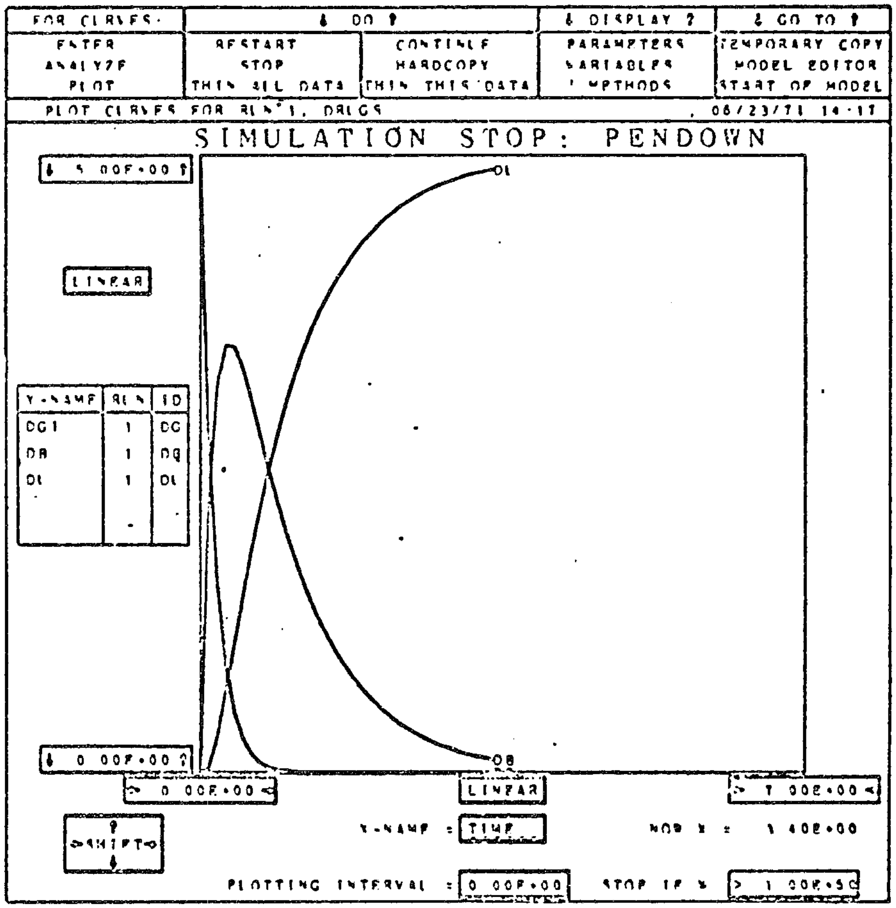
slowly. We assume, from the curve's reasonable appearance (Fig. 6), that we described at least the body component of the model correctly, and our choice of integration method is adequate. We see from the curve labeled DB that, as expected, the drug remains in the blood for only a short time.

We reintroduce the capsule drug by changing DOSAGE back to 3., then restart the simulation and watch the curves being continuously updated as it runs. Once it becomes apparent that the capsule is not effective, i.e., that the value of DB drops too low, we stop the simulation. Apparently (Fig. 7), the drug is not released from the pellets in time to replace the drug that leaves the blood. To correct this, we change the mean times of release from 2., 4., 6., and 8. to 1., 2., 3., and 4. We then rerun the simulation and get much better results. The amount of drug in the blood should be greater than 2.5. In order to determine if this is achieved, we place the pen down in the central area to establish an x-y meter, then drag this meter to a place where Y (corresponding to DB as well as DGI and DU) is equal to 2.5 (Fig. 8). Our choice of means was good, but they need to be adjusted to maximize the total duration of capsule effectiveness.

While changing the means for further trials we realize that rather than controlling four means, we would prefer to deal only with the first mean and the interval between mean times of release. To reformulate the model in this way, we return to its description and add another box to the definition of the capsule component. In this box we write

$$M2 = M1 + INTVAL$$

and similar equations for M3 and M4. This replaces the parameters M2, M3, and M4 with the single parameter INTVAL, which we set to 1. and mark modifiable. Once this is accomplished, we retranslate the model, then continue to resimulate it and change parameters until we have established a satisfactory drug formulation and administration policy.



PLOT CURVES FOR RUN 1, DRUGS , 06/23/71 14:17

SIMULATION STOP: PENDOWN

Y-NAME	RUN	ID
DG1	1	DG
DB	1	DB
DL	1	DL

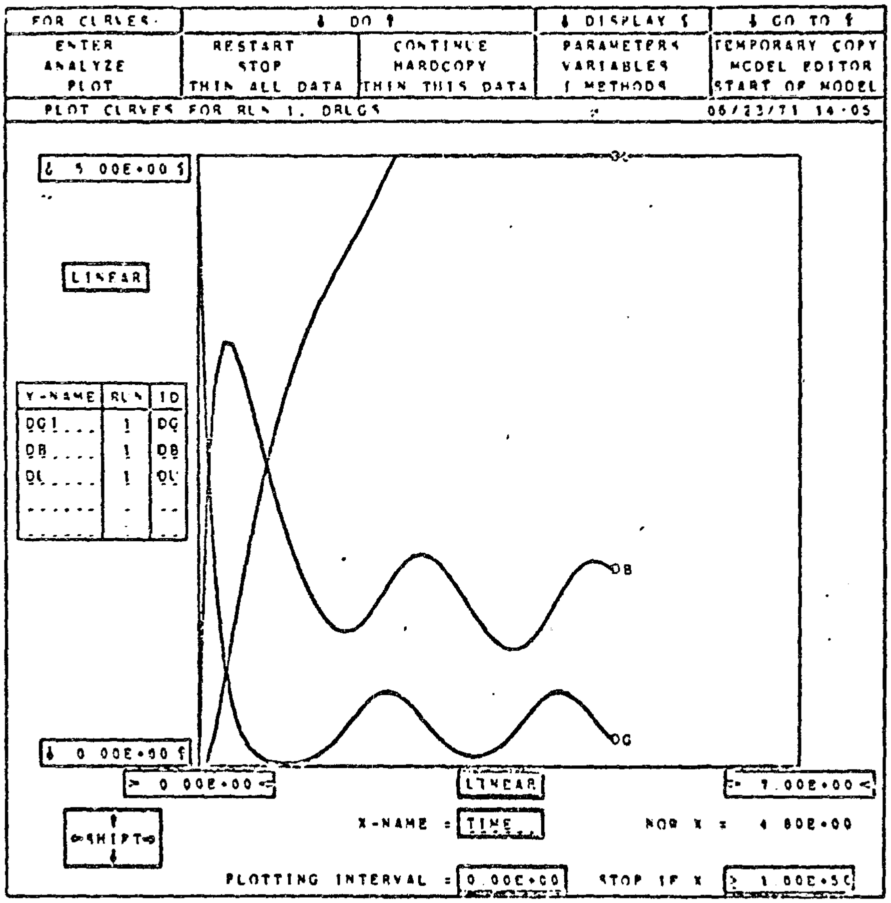
X-NAME = TIME

NOW X = 1.40E+00

PLOTTING INTERVAL = 0.00E+00

STOP IF X > 1.00E+50

Fig. 6—The simulation run with no capsule drug



PLOT CURVES FOR RUN 1. DRUGS 06/23/71 14·05

Y-NAME	RUN	ID
DG1...	1	DG
DB....	1	DB
DL....	1	DL
......	.	..
......	.	..

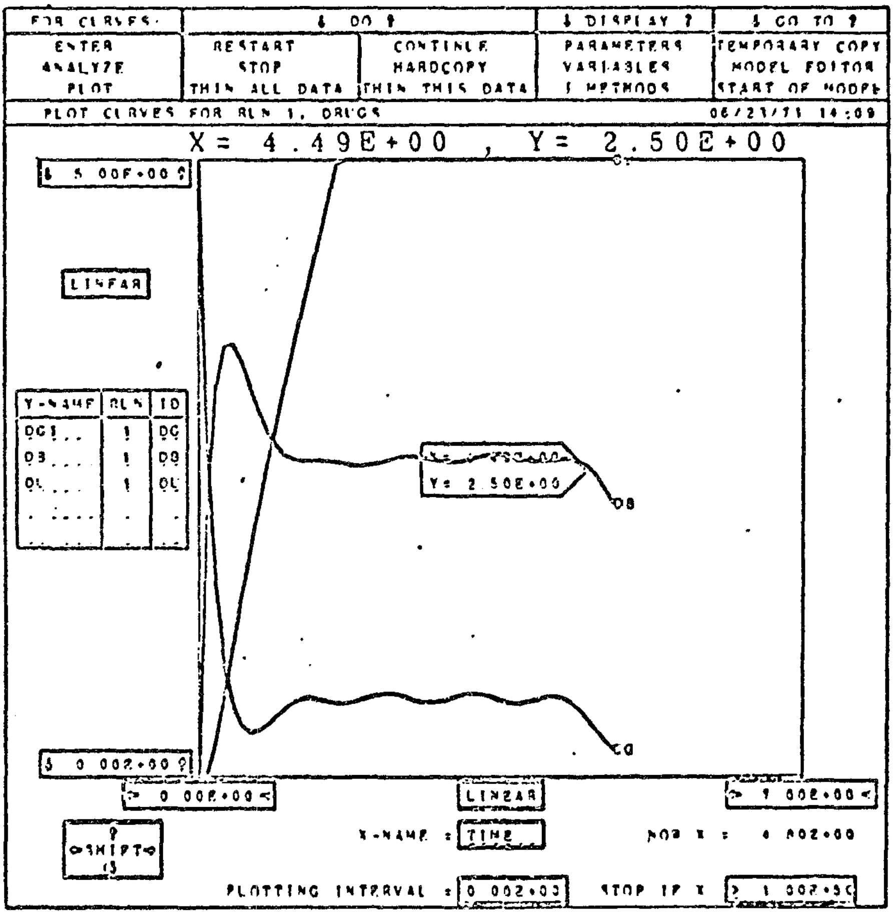
X-NAME = TIME

NOW X = 4.80E+00

PLOTTING INTERVAL = 0.00E+00

STOP IF X > 1.00E+50

Fig.7—The simulation run with the parameter values shown in Fig.4



PLOT CURVES FOR RUN 1, DRUGS 06/23/71 14:09

X= 4.49E+00 , Y= 2.50E+00

Y-NAME	RUN	ID
DC1 . .	1	DC
DB . . .	1	DB
DL . . .	1	DL
. . . . .	.	.
. . . . .	.	.

X= 4.49E+00
Y= 2.50E+00

X-NAME : TIME NOW X = 4.80E+00

PLOTTING INTERVAL : 0.00E+00 STOP IF X > 1.00E+30

Fig. 8—The simulation run with means = 1., 2., 3., and 4.