

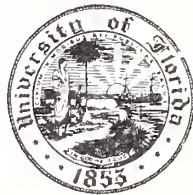
DYNAMO

User's Manual

Second Edition

651.26
P978d2

UNIVERSITY
OF FLORIDA
LIBRARIES





Digitized by the Internet Archive
in 2013

<http://archive.org/details/dynamousersmanua00pugh>



DYNAMO
USER'S MANUAL



DYNAMO USER'S MANUAL

Second Edition

ALEXANDER L. PUGH, III



THE M.I.T. PRESS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts, and London, England

COPYRIGHT © 1961 AND 1963 BY THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ALL RIGHTS RESERVED

Second Edition, Revised, May 1963
Reprinted, March 1966



LIBRARY OF CONGRESS CATALOG CARD NO. 61-18482

PRINTED IN THE UNITED STATES OF AMERICA

Chapter

CONTENTS

	<u>Page</u>
Introduction	1
I.1 Brief History of DYNAMO	2
I.2 Facts and Figures about DYNAMO	3
I.3 DYNAMO Checker for the IBM 1620 Computer	3
1. Primer	4
1.1 Explanation of Time Notation and Variable Types	4
Level	5
Rate	5
Auxiliary	5
Order of Computation	5
1.2 The Situation to Be Modeled and the Model Equations	5
1.3 Translating the Model into DYNAMO Language	7
1.4 Completing the Model	9
Initial Values	9
Input	10
Constants	10
Output	11
Run Details	14
1.5 Card Punching	15
1.6 Assembling a Run	16
Order of Cards	16
Reruns	16
1.7 Error Examples	17
2. Reference Manual	20
2.1 Introduction	20
2.2 Equation Cards	20
Quantity Types	20
Quantity Names	21
Equation Forms	21
Subscripts and Permissible Defining Quantities	23
Initial Values	24
Equation Card Punching	25
Rule of the Final Space	25
Continuation Cards	25
Numerical Values	26
Comments	26
Constant Cards	26
Tables and Initial Values for Boxcar Trains	27

2.3	Direction Cards	27
	Identification Card	27
	RUN Card	28
	SPEC Card	28
	PRINT Cards	29
	Scaling of Quantities for Printing	30
	PLOT Cards	30
	NOTE Cards	31
	NOISE Cards	31
2.4	Order of Cards; Runs and Reruns	32
2.5	Boxcars and Boxcar Trains	32
2.6	Special Functions	34
	Common Functions	34
	Random Numbers	35
	Third-Order Delay	36
	Pulse Source	36
	Step Function	37
	Ramp Function	37
	Sampler	37
	Maximum Function	38
	Minimum Function	38
	Clipping or Limiting Function	38
	Switch Function	38
	Table Functions	38
	Summing Functions	40
	Treatment of Round-off in Comparisons with TIME	41
	Variable Boxcar-Shifting, Pulse, Sample, Step, and Ramp Times	41
2.7	The Choice of DT	42
2.8	Estimating the Computer Time Required	43
3.	How DYNAMO Works	44
	3.1 Input Phase	44
	3.2 Generation Phase	45
	3.3 Running Phase	48
	3.4 Printing Phase	49
	3.5 Plotting Phase	49
	3.6 Rerun Phase	49
	Appendix A Scaling Letters	50
	Appendix B Equation Forms	51
	Index	57

INTRODUCTION

DYNAMO is a computer program for translating mathematical models from an easy-to-understand notation into tabulated and plotted results. The models may be of any dynamic feedback system such as arises in business, economics, or engineering. The principal limitation on the model is that it be a continuous representation of the real world. As DYNAMO does not recognize individual items or events, models of job shops and the like can not be tested.†

(The question occasionally arises, "What does DYNAMO look like?" This question particularly bothers some people because neither the input model nor the results of the model are DYNAMO. Unfortunately, the answer is the same as the answer to "What does a procedure look like?" A program is a procedure for converting some sort of data into some desired results. As such, it can be expressed in any of several languages, and its appearance is that of the language and the medium on which it is recorded: a printed page, a deck of punched cards, or a magnetic tape.)

A model written in DYNAMO consists of a number of algebraic relationships that relate the variables one to another.‡ The notation is very similar to that of a general-purpose scientific compiler (for example, FORTRAN). The advantages of using DYNAMO rather than a general-purpose compiler are

- The DYNAMO language is easily understood. The time notation greatly aids one in comprehending the order of computation.
- DYNAMO language is easy to learn. The lack of unnecessary flexibility means that there is less to study, and what must be learned can be readily mastered in the context of its eventual use.
- Ingenuity, which is such an important part in the use of most computer languages, is not involved in most DYNAMO formulations. A corollary is that models formulated by different users of the same situation will be quite similar and easily compared.
- Unlike most compilers that have both a "source" and "object" language, DYNAMO has but one. DYNAMO's very short compilation time means that there is little value in saving the machine language code from one run to the next.
- DYNAMO's output includes graphic results, saving the user the time and effort that he would otherwise spend plotting his data. Furthermore, all forms of the output can be specified easily, even to the point of not having to supply the scale factors for the tabulated results or the scales for the plotted results.

† Persons familiar with both digital and analogue computers will find that DYNAMO in many ways behaves more like an analogue than a digital computer.

‡ Mathematicians will recognize these relationships as constituting a set of zero- and first-order difference equations.

- DYNAMO does not require that the equations be in the order necessary for computation. If the order is wrong (as is generally the case), DYNAMO will correct it.
- DYNAMO will create some of the required initial conditions. While the user must decide upon the desired initial state for the model, he can leave many of the trivial relationships to DYNAMO.
- Finally, DYNAMO's extensive error checking simplifies the problem of developing a meaningful model. DYNAMO checks for practically every logical inconsistency. Furthermore, the error comments are designed to reduce the problem of understanding and correcting the error.

This manual is divided into three chapters. Chapter 1 is a primer, Chapter 2 is a reference manual, and Chapter 3 is a nontechnical description of the internal structure of DYNAMO.

Chapter 1 begins with a brief explanation† of the time notation and definitions of the three principal variable types: levels, auxiliaries, and rates. Next, it presents a model expressed in this time notation and shows what modifications are necessary to make the model acceptable to DYNAMO. It discusses punching the appropriate cards and preparing them to be run on the computer. Finally, it discusses some of the errors that might have been made in this model and the error comments that DYNAMO would have produced.

Chapter 2 provides the rules of DYNAMO in a reference format. The user should read Chapter 2 once to become familiar with the details of DYNAMO not mentioned in Chapter 1. After that he will call on Chapter 2 to answer quite specific questions. An index to Chapter 2 is provided for this purpose.

For those interested in the internal structure of DYNAMO, Chapter 3 provides a brief, nontechnical description of how a model is translated into machine language and ultimately into tabulated and plotted results.

I.1 Brief History of DYNAMO

The history of DYNAMO begins with a program called SIMPLE (Simulation of Industrial Management Problems with Lots of Equations), which was written by Richard K. Bennett in the spring of 1958 for the IBM 704 Computer. SIMPLE contained most of the basic features of DYNAMO, including the plotting routine (written by Edward B. Roberts), but the model specifications had to be stated in a rather primitive form, and very few checks were performed on these specifications. These shortcomings were corrected by DYNAMO (DYNAMIC Models), which was written by Dr. Phyllis Fox (Mrs. George Sternlieb) and Alexander L. Pugh, III, and was completed in the spring of 1959. Mr. Roberts modified and improved the plotting routine to be a part of DYNAMO, and Mrs. Grace Duren and David J. Howard assisted with other phases of the program.

The program was improved the following summer by the same group of people and called DYNAMO II. Since then there have been a number of improvements made by Mr. Pugh, including the conversion of the program from the IBM 704

† Further information can be found in Industrial Dynamics by Jay W. Forrester, The M.I.T. Press, Cambridge, Mass., 1961.

Computer to the IBM 709 and 7090 Computers. As none of these changes was a major change, the series of names DYNAMO III, DYNAMO IV, etc., was dropped in favor of the older name, DYNAMO.

The financial support for all of these programs was provided by the Ford Foundation as a part of its support of the research in Industrial Dynamics. Computer time was made available by the M.I.T. Computation Center. A part of the latest modification was supported by a direct grant of computer time by the International Business Machines Corporation.

I.2 Facts and Figures about DYNAMO

DYNAMO consists of approximately 10,000 instructions written in machine language. The present version required 6 man-years of effort (including the work that went into SIMPLE).

The largest model that DYNAMO can handle is about 1400 equations (not counting initial-value equations).

It will operate on any IBM 709 or 7090 Computer with at least one data channel and three tape drives. It does not require data-channel traps. The earlier 704 version is still available, although no effort is being made to keep that version current.

The 709-90 version of DYNAMO is available from SHARE Distribution Agency, IBM Corporation, 112 East Post Road, White Plains, New York.

I.3 DYNAMO Checker for the IBM 1620 Computer

A program that checks for most of the common errors in DYNAMO models has been written for the IBM 1620 Computer by two M.I.T. students, Henry V. Lett, Jr., and Douglas W. Gaidry. When a 1620 computer is more accessible than the 7090, the "DYNAMO Checker" speeds the process of eliminating mistakes from a model. The checker was originally written for a machine with 60,000 digits of memory, a card read-punch unit, and the indirect addressing feature, but has since been modified for 1620's with less memory.

Chapter 1

PRIMER

This chapter is divided into seven sections. The first explains the time notation and the variable types in DYNAMO. The second describes the model that will be used as a vehicle to present DYNAMO. The third translates that model to the DYNAMO language. The fourth section adds some necessary details such as initial values and the specification of the desired results. The fifth discusses punching the cards for the model. The sixth explains the order of the cards and the method for rerunning a model. The seventh discusses various types of errors and the error comments DYNAMO makes.

1.1 Explanation of Time Notation and Variable Types

The time notation and the types of variables are discussed in detail in Chapter 7 of Industrial Dynamics[†] but will be described briefly here.

The basis for the time notation is the procedure by which the computer calculates the results, which is to move through TIME in discrete steps and calculate all the variables at each step. Figure 1-1 shows the procedure graphically.

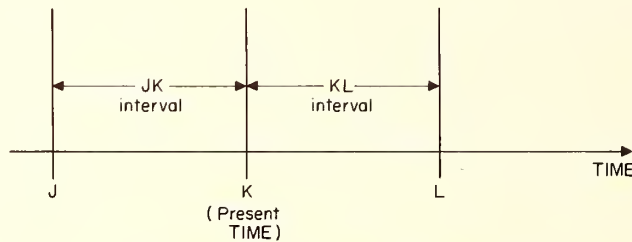


Figure 1-1 Time Notation.

The TIME for which the calculations are currently being made is called TIME K. The previous TIME for which calculations were made is called J, and the next instant for which calculations will be made is L. The intervals between these times are called JK and KL. The length of these intervals is called DT.

The names of instants and the intervals are used to specify when a quantity is calculated and when the quantities used in the calculation were previously calculated. Once all the variables have been calculated for the instant K and the interval KL, the computer moves forward one time step, and the values that were associated with TIME K are now related to TIME J.

[†] Forrester, Jay W., Industrial Dynamics, The M.I.T. Press, Cambridge, Mass., 1961.

This notation system was chosen in preference to one derived from differential equations because many users unfamiliar with higher mathematics will be able to understand and to use this step-by-step notation. (The user is cautioned against thinking of his model as being discrete because of the discrete nature of the notation system or the procedure of the computer.)

There are three principal types of variables in DYNAMO: levels, rates, and auxiliaries.

Level

A level, which is calculated at TIME K, is a quantity that depends upon its value at TIME J and on other quantities at that TIME or in the JK interval. Inventory is an example in that the inventory today is equal to the inventory at an earlier time plus what has been received minus what has been shipped during the interim.

Rate

The decisions in business and economic models are rates. Rates are the flows of tangible things from one level to the next. They are computed at TIME K for the interval KL from levels and auxiliaries at TIME K and occasionally from rates in the JK interval.

There is some confusion about rates in that all quantities with the dimension of something per unit time are not necessarily rates. In particular, averaged or smoothed rates, which are measured in something per unit time, are levels.

Auxiliary

Auxiliaries are variables that are introduced to simplify the algebraic complexity of rate equations. They generally have some physical meaning and consequently simplify the understanding of the model. They are computed at TIME K from levels and other auxiliaries at the same time and occasionally from rates in the JK interval. By their nature they can be eliminated by substitution into the rate equations.

Order of Computation

The order of computation at TIME K is first levels, which are based on quantities from TIME J and the JK interval, next auxiliaries, which are based on levels and auxiliaries computed earlier at TIME K and on rates JK, and finally rates, which are based on levels and auxiliaries from K and other rates from JK.

1.2 The Situation to Be Modeled and the Model Equations

As an example, we shall model a retail store. This model will be similar to the retail sector of the production distribution system described in Chapter 15 of Industrial Dynamics.

The retail store receives orders for goods. These orders reside in a "pool" of unfilled orders until they are filled from inventory. The shipments are proportional to the number of unfilled orders, provided there is sufficient inventory. The store orders to replace the items sold and to correct inventory to the "desired" level. The desired level corresponds to a number of weeks of averaged sales. The goods that the store orders arrive after a delay.

We shall use the same variable names as are used in Chapter 15 of Industrial

Dynamics. The inventory at time K is equal to the inventory at time J plus what was received during the interval JK minus what was shipped during the interval JK. Thus we have the level equation

$$IAR.K = IAR.J + (DT)(SRR.JK - SSR.JK) \quad 1-1, L$$

IAR Inventory Actual at Retail (goods)
 DT Delta Time (weeks)
 SRR Shipments Received at Retail (goods/week)
 SSR Shipments Sent from Retail (goods/week)

Similarly, the unfilled orders at time K are equal to the unfilled orders at time J plus new orders received during the interval JK minus shipments sent during the interval JK. This is a similar level equation:

$$UOR.K = UOR.J + (DT)(RRR.JK - SSR.JK) \quad 1-2, L$$

UOR Unfilled Orders at Retail (goods)
 RRR Requisitions Received at Retail (goods/week)
 SSR Shipments Sent from Retail (goods/week)

The rate equation for shipments from retail is

$$SSR.KL = UOR.K/DFR$$

provided that there is sufficient inventory. The constant of proportionality is $1/DFR$. If $(SSR.KL)(DT)$ exceeds $IAR.K$, we shall ship more from inventory than at present exists in the inventory. Consequently, we want to ship the smaller of $UOR.K/DFR$ and $IAR.K/DT$, or

$$SSR.KL = \text{MIN}(UOR.K/DFR, IAR.K/DT) \quad 1-3, R$$

SSR Shipments Sent from Retail (goods/week)
 UOR Unfilled Orders at Retail (goods)
 DFR Delay in Filling orders at Retail (weeks)
 IAR Inventory Actual at Retail (goods)

where MIN indicates that the quantity on the left of the equality sign will have the value of the minimum of the two terms in the parentheses.

The store places orders to replace the items sold and to correct the inventory to the desired level. Therefore, the rate equation for the orders will contain two terms, the goods that have just been sold and the correction to adjust the actual inventory to the desired level:

$$PSR.KL = RRR.JK + (1/DIR)(IDR.K - IAR.K) \quad 1-4, R$$

PSR Purchase orders Sent from Retail (goods/week)
 RRR Requisitions Received at Retail (goods/week)
 DIR Delay in adjusting Inventory at Retail (weeks)
 IDR Inventory Desired at Retail (goods)
 IAR Inventory Actual at Retail (goods)

The fraction $1/DIR$ is that fraction of the difference between ideal and actual inventory that is ordered each week.

The desired inventory is a number of weeks of averaged or smoothed sales. This implies the auxiliary equation

$$\text{IDR.K} = (\text{AIR})(\text{RSR.K}) \quad 1-5, \text{ A}$$

IDR Inventory Desired at Retail (goods)
 AIR constAnt for Inventory at Retail (weeks)
 RSR Requisitions Smoothed at Retail (goods/week)

If we average orders received by exponential smoothing,[†] we have the level equation

$$\text{RSR.K} = \text{RSR.J} + (\text{DT}/\text{DRR})(\text{RRR.JK} - \text{RSR.J}) \quad 1-6, \text{ L}$$

RSR Requisitions Smoothed at Retail (goods/week)
 RRR Requisitions Received at Retail (goods/week)
 DRR Delay in smoothing Requisitions at Retail (weeks)

The goods that the store orders arrive after a delay. As delays of rates occur so frequently in models of this sort, DYNAMO is equipped to create a delay from a simple specification. All that is necessary is to indicate what variable is to be delayed and how long the delay is:

$$\text{SRR.KL} = \text{DELAY}(\text{PSR.JK}, \text{DTR}) \quad 1-7, \text{ R}$$

SRR Shipments Received at Retail (goods/week)
 PSR Purchases Sent from Retail (goods/week)
 DTR Delay in Transit at Retail (weeks)

1.3 Translating the Model into DYNAMO Language

DYNAMO cannot accept the equations in exactly the form given in Equations 1-1 through 1-7. This section will complete the equations, and the next will add the necessary details so that the model can be run on the computer.

The variable and constant names are already satisfactory for DYNAMO. DYNAMO permits the use of up to and including five upper-case letters, for example, X, INV, ABCDE. After the first letter, a number may be substituted for a letter if desired. For example, A3BC is permitted but not 3ABC.

In addition to the equation for each variable, DYNAMO requires the variable type and the equation form number. The variable type was discussed with each equation. The equation form number is required because DYNAMO is not able to translate any arbitrary order of variables and algebraic symbols into machine language. Rather, it can translate only those types of equations shown in Appendix B. In fact, the equation must have the exact form shown under "Exact Punching Format" in Appendix B. No algebraic symbols may be interchanged. No parentheses may be added or forgotten. On the other hand, there is some flexibility in the sign of a variable. We shall discuss this in the context of Equation Form 1, which incidentally is the proper equation form for IAR and UOR.

The exact punching format for Equation Form 1 as given in Appendix B is

$$1L \quad V.K = V.J + (DT)(\pm P \pm Q)$$

[†] Exponential smoothing is discussed in Appendix E of Industrial Dynamics.

The initial number 1 is the equation form number. The letter L indicates that this equation form may be used to define only levels. The letter V stands for the variable being defined. The letters P and Q stand for the two variables that when summed and multiplied by DT will be added to V.J to form V.K. The quantities P and Q are not restricted to being variables; they might also be constants (either the name of a constant or an actual numerical value). The \pm before P and Q indicates that either sign may be used. Following a left parenthesis, slash, † equality sign, or comma, the plus sign may be dropped. Hence the equation for inventory may be written either

$$1L \quad IAR.K = IAR.J + (DT) (+SRR.JK - SSR.JK)$$

or

$$1L \quad IAR.K = IAR.J + (DT) (SRR.JK - SSR.JK) \quad 1-8$$

The equation for UOR is similar:

$$1L \quad UOR.K = UOR.J + (DT) (RRR.JK - SSR.JK) \quad 1-9$$

Note that most of the equation numbers in Appendix B are not followed by any letter and that V has no time subscript. These equation forms may be used to define auxiliaries, rates, or three variable types that have not been discussed yet: supplementaries, initial values, or computed constants. If these forms are used, the equation type must be followed by the letter corresponding to the variable type of the quantity being defined, and all the variables in the equation must have the proper subscripts.

Searching through Appendix B for the equation type for SSR, we find no such equation type. But there is one that reads

$$54 \quad V = \text{MIN}(\pm P, \pm Q)$$

If we define two new auxiliary variables and set one equal to $IAR.K/DT$ and the other equal to $UOR.K/DFR$, we can substitute them for P and Q. (The computation of complicated terms for other equations is one of the common roles of auxiliaries.) If we designate the largest flow that will just keep the inventory from going negative as the Negative Inventory rate at Retail (NIR) and designate the goods we are trying to ship provided there is sufficient inventory as Shipments Tried at Retail (STR), we have

$$20A \quad NIR.K = IAR.K/DT \quad 1-10$$

$$20A \quad STR.K = UOR.K/DFR \quad 1-11$$

$$54R \quad SSR.KL = \text{MIN}(STR.K, NIR.K) \quad 1-12$$

The equations for PSR and IDR raise no new points:

$$40R \quad PSR.KL = RRR.JK + (1/DIR) (IDR.K - IAR.K) \quad 1-13$$

$$12A \quad IDR.K = (AIR) (RSR.K) \quad 1-14$$

On the other hand, there is no equation form that exactly matches the equation for RSR.

† Or solidus.

$$V.K = V.J + (DT/\pm Y)(\pm P \pm Q)$$

just does not exist in Appendix B, although there is

$$3L \quad V.K = V.J + (DT)(1/\pm Y)(\pm P \pm Q)$$

which is algebraically equivalent. Therefore, the equation for RSR must be written

$$3L \quad RSR.K = RSR.J + (DT)(1/DRR)(RRR.JK - RSR.J) \quad 1-15$$

Again, there is no equation form that exactly matches the equation for SRR, although

$$39R \quad V.KL = DELAY3(\pm P, C)$$

comes very close. (In Equation Form 39, C indicates that a constant must be used for the length of the delay.) Equation Form 39 is a "third-order" delay, which is described in detail in Chapter 9 and Appendix H of Industrial Dynamics. The equations for it are listed on page 36 of this manual. Accepting the third-order delay as a satisfactory delay for SRR, we have

$$39R \quad SRR.KL = DELAY3(PSR.JK, DTR) \quad 1-16$$

1.4 Completing the Model

Though all the model equations are written, the model is not complete. There are five more problems to be taken care of before it can be run:

- We must specify the initial values.
- We must provide an input.
- We must assign values to our constants.
- We must specify what variables are to be printed or plotted.
- We must identify our model, and specify the desired length of run.

Initial Values

There is no place where we can begin the calculations without requiring the values of some variables from an earlier time. Starting with the auxiliaries[†] at TIME = 0 reduces the number of values required to a minimum, but still "initial values" must be provided for certain variable types. These initial values are the values the variables would have had if the model had been running prior to TIME = 0.

DYNAMO is able to determine the initial value of most auxiliaries and rates that require one, but the user must provide values for all levels. (A discussion of the problems that DYNAMO might encounter trying to assign initial values to auxiliaries and rates will be postponed to Chapter 2.)

At this point it is necessary to decide what condition in the model we wish our initial values to represent. One good choice is to start the model in "steady state," that is, to assume that the input has been constant for some time, all transients have died out, and everything is in balance.

If the incoming orders in our example have been constant for some time, then

[†] The sequence of computations is given in Section 3.3 and Chapter 7 of Industrial Dynamics.

we should expect retail shipments, orders that the retail store generates, and shipments to the retail store all to have the same value as the incoming orders. The problem now is to choose our levels so that these rates will be equal. Let us assume that RRR has some initial value. (If the rate RRR requires an initial value, DYNAMO will compute one.) We shall assume that the inventory is sufficient so that $SSR = UOR/DFR$. If SSR is to equal RRR, then UOR must equal $(DFR)(RRR)$, and we have the initial value for one of our levels:

$$12N \quad UOR = (DFR)(RRR) \quad 1-17$$

Again we must explain DYNAMO conventions. The equation type is N for iNitial value. No time subscripts are used on either side of the equation.

The initial value for RSR is quite simple. If RRR has been constant for a long time, then the smoothed value of RRR would be equal to it:

$$6N \quad RSR = RRR \quad 1-18$$

If $PSR = RRR$, then $IAR = IDR$:

$$6N \quad IAR = IDR \quad 1-19$$

Input

As it is not possible to change a model or to add data during a run, the input must be planned at the time the model is being constructed. There is a wide range of functions that might be used for the input. Their significance is discussed in Chapter 12 of Industrial Dynamics. We shall try a step. In order to determine whether or not our initial values actually started the model in steady state, we shall delay the step until the 5th week:

$$7R \quad RRR \cdot KL = RRI + RCR \cdot K \quad 1-20$$

$$45A \quad RCR \cdot K = STEP(STH, 5) \quad 1-21$$

where STH is the height of the step.

Since this is the first time we have used a numerical value in an equation, it deserves comment. Either a numerical value or a literal constant (for example, STH) may be used on the right side of any equation. The use of literal constants has several advantages. If we decide to change a constant that appears in several places (for instance, DFR appears in both the equation for STR and the initial-value equation for UOR), only one change is necessary to change it throughout the model. There is no chance of forgetting to change an initial condition that also depends on this constant. Furthermore, literal constants are the only ones that can be changed in reruns, as will be seen shortly.

When a constant appears only once in a model and there is no desire ever to change it, using a numerical value is slightly simpler.

Constants

The constants are very easy to specify. For example,

$$C \quad AIR = 8$$

The letter C on the left indicates that a constant follows. Thus we have set the retail store's desired inventory at 8 weeks of smoothed sales. We shall not try to

justify the realism of our choice of constants; that is not the purpose of this primer. We shall choose the other constants as follows:

```
C      DFR=1
C      DIR=4
C      DRR=8
C      DTR=2
C      RRI=1000
C      STH=100
```

Output

A sample of the first page of tabular results that DYNAMO can generate is shown in Figure 1-2. Incidentally, these are the tabular results for the model we have been discussing above. On the first two lines are the names of the variables whose values are tabulated below. Immediately beneath the names are the scale factors for the results that follow. The printed results must be multiplied by the appropriate scale factor to get the actual value of the results. The scale factors are expressed in powers of 10. The entry E+03 means the tabulated results should be multiplied by 10^{+3} , or 1000, to get the actual value of the results. In the case in Figure 1-2, the scale factors are very simple because E+00 indicates that the tabulated results should be multiplied by 10^0 or 1. On later pages of the tabulated results the names are repeated, but not the scaling factors.

To obtain such a tabulation, it is necessary to specify only the names of the variable and the column in which it is to be printed. As an example, the card that yielded Figure 1-2 was

```
PRINT 1)IAR, IDR/2)UOR/3)RRR, SSR/4)PSR, SRR
```

Note the following:

- TIME was printed in the leftmost column of the output without being requested.
- In addition to the column with TIME, there are 14 columns across the page, and up to 10 items can be printed in each.
- On the PRINT card, items within a column were separated by commas while columns were separated by slashes.
- More than one PRINT card may be used.

There is one other DYNAMO output that is relatively unique, the plotted output. Figure 1-3 is an example. On the left is the equivalence between the variable name and the character plotted. Under this are the scales, and at the end of the scales are the characters that are plotted on those scales.

When two or more characters fall into the same position, the superposition is indicated at the top of the graph. For example, at TIME = 0, the letters R, S, P, and Q all fell on the same spot; R was printed on the graph, and RSPQ was printed at the end of the line. The first letter, R, is always the one appearing on the graph. If superpositions occurred at more than one position on a line, the superimposed sets are separated by commas, as can be seen at TIME = 15. The card that instructed DYNAMO to generate Figure 1-3 is

PAGE 2	2698JP																			
TIME	IAR	UOR	RRR	PSR	IAR	UOR	RRR	PSR	IAR	UOR	RRR	PSR	IAR	UOR	RRR	PSR	IAR	UOR	RRR	PSR
E+00	E+00	E+00	E+00	E+00	DR		SSR	SRR	DR		SSR	SRR	DR		SSR	SRR	DR		SSR	SRR
	E+00	E+00	E+00	E+00			E+00	E+00			E+00	E+00			E+00	E+00				
	8000.0	1000.0	1000.0	1000.0			1000.0	1000.0			1000.0	1000.0			1000.0	1000.0				
	8000.0	1000.0	1000.0	1000.0			1000.0	1000.0			1000.0	1000.0			1000.0	1000.0				
	2.000	8000.0	1000.0	1000.0			1000.0	1000.0			1000.0	1000.0			1000.0	1000.0				
	8000.0						1000.0	1000.0			1000.0	1000.0			1000.0	1000.0				
	4.000	8000.0	1000.0	1000.0			1000.0	1000.0			1000.0	1000.0			1000.0	1000.0				
	8000.0						1000.0	1000.0			1000.0	1000.0			1000.0	1000.0				
	6.000	7968.0	1065.1	1100.0			1100.0	1131.6			1065.1	1015.2			1065.1	1015.2				
	8094.6						1065.1	1015.2			1065.1	1015.2			1065.1	1015.2				
	8.000	7929.0	1095.8	1100.0			1100.0	1180.6			1100.0	1180.6			1100.0	1180.6				
	8251.5						1095.8	1118.5			1095.8	1118.5			1095.8	1118.5				
	10.000	8031.4	1099.5	1100.0			1100.0	1185.5			1100.0	1185.5			1100.0	1185.5				
	8373.5						1099.5	1173.2			1099.5	1173.2			1099.5	1173.2				
	12.000	8192.1	1099.9	1100.0			1100.0	1169.1			1100.0	1169.1			1100.0	1169.1				
	8468.3						1099.9	1181.8			1099.9	1181.8			1099.9	1181.8				
	14.000	8344.8	1100.0	1100.0			1100.0	1149.3			1100.0	1149.3			1100.0	1149.3				
	8542.1						1100.0	1168.2			1100.0	1168.2			1100.0	1168.2				
	16.000	8463.5	1100.0	1100.0			1100.0	1134.0			1100.0	1134.0			1100.0	1134.0				
	8599.5						1100.0	1149.8			1100.0	1149.8			1100.0	1149.8				
	18.000	8548.0	1100.0	1100.0			1100.0	1124.0			1100.0	1124.0			1100.0	1124.0				
	8644.1						1100.0	1134.7			1100.0	1134.7			1100.0	1134.7				
	20.000	8607.1	1100.0	1100.0			1100.0	1117.9			1100.0	1117.9			1100.0	1117.9				
	8678.8						1100.0	1124.6			1100.0	1124.6			1100.0	1124.6				
	22.000	8649.8	1100.0	1100.0			1100.0	1114.0			1100.0	1114.0			1100.0	1114.0				
	8705.7						1100.0	1118.3			1100.0	1118.3			1100.0	1118.3				
	24.000	8682.2	1100.0	1100.0			1100.0	1111.1			1100.0	1111.1			1100.0	1111.1				
	8726.7						1100.0	1114.2			1100.0	1114.2			1100.0	1114.2				
	26.000	8707.6	1100.0	1100.0			1100.0	1108.9			1100.0	1108.9			1100.0	1108.9				
	8743.0						1100.0	1111.2			1100.0	1111.2			1100.0	1111.2				
	28.000	8727.7	1100.0	1100.0			1100.0	1107.0			1100.0	1107.0			1100.0	1107.0				
	8755.7						1100.0	1108.9			1100.0	1108.9			1100.0	1108.9				
	30.000	8743.7	1100.0	1100.0			1100.0	1105.5			1100.0	1105.5			1100.0	1105.5				
	8765.5						1100.0	1107.0			1100.0	1107.0			1100.0	1107.0				
	32.000	8756.3	1100.0	1100.0			1100.0	1104.2			1100.0	1104.2			1100.0	1104.2				
	8773.2						1100.0	1105.5			1100.0	1105.5			1100.0	1105.5				

Figure 1-2 First page of tabulated results of model.

PLOT IAR=I,UOR=U/RRR=R,SSR=S,PSR=P,SRR=Q

Note:

- All the variables were plotted against TIME, without mentioning TIME on the PLOT card.
- DYNAMO chose the scales so that none of the variables went off scale. Page 31 explains how the user can set his own scales.
- Up to 10 variables may be plotted on a graph.
- Variables that are to be plotted to the same scale are separated by commas on the PLOT card, while variables or groups of variables that are to have their own scales are separated by slashes.
- Only one PLOT card (plus as many continuation cards as necessary) is used to specify the variables that are to be plotted on a graph, but more graphs may be obtained by using additional PLOT cards.
- Variables may be either printed or plotted, or both.

Run Details

Three additional cards are necessary to complete the model. They are the run SPECifications card, the RUN number card, and the identification card.

The run SPECifications card tells DYNAMO the interval of time between computations, DT, how long to run the model, how frequently to tabulate the results, and how frequently to plot the results. For example,

SPEC DT=0.1/LENGTH=50/PRTPER=2/PLTPER=0.5

This card sets DT equal to 1/10 week[†] and the length of the run equal to 50 weeks; it instructs DYNAMO to tabulate (print) the results every 2 weeks and plot them every half week. Note that only one unit of measure of time may be used in a model. Once we have started measuring time (including delays) in weeks, we must continue to do so throughout the model.

The RUN-number card provides DYNAMO with the number of this particular run. DYNAMO prints this number at the top of each page along with the page number. Any characters up to a total of six may be used. Each group of DYNAMO users will devise its own method of generating these numbers. For example,

RUN 2698JP

The identification card provides the Computation Center with the information necessary to identify and run the deck:[‡]

* M478-248,DYN,TEST,1,1,0,0

The asterisk denotes an identification card. M478-248 is the problem number and programmer's number (both are assigned by the Computation Center when the user requests computer time). DYN identifies the run as a DYNAMO run. The word TEST (or the word RESULTS) establishes whether the model is

[†] The choice of the size of DT is discussed here in Section 2.7 and in Industrial Dynamics in Appendix D.

[‡] For the procedure at installations other than M.I.T., see page 27.

just being tested or whether results are expected which may be used in a report, thesis, or publication. The first figure following TEST is the number of minutes of computer time that the model is expected to take. Section 2.8 provides a formula to estimate this time. The second figure is the maximum number of minutes it should take, including all errors in estimating the time. This is the maximum time the Computation Center will allow the model to run before they will stop it. The final two zeros pertain to other programs but are required on all DYNAMO runs.

1.5 Card Punching

Up to this point we have not mentioned the details of card punching, that is, where the information goes on the card. The word PRINT, PLOT, RUN, or SPEC starts in column 1. Similarly, the equation number starts in column 1 and is immediately followed by the letter corresponding to the quantity type. The C for a constant and the asterisk indicating the identification card also are punched in column 1. The information starts in column 7 and is punched without any intervening spaces.

The information may not extend beyond column 72. If more space is required, continuation cards must be used. These cards will not be discussed further here but are described on pages 25 and 26.

The space following the information indicates the end of the information, and thus is actually part of it. If there is no room on the card for this space, a continuation card must be provided to supply it.

Following the "final space," any sort of comment may be added. Such comments are helpful in remembering what a variable stands for and, in the case of constants, the units of the numerical value. It is easier to read these comments if they all start in the same column.

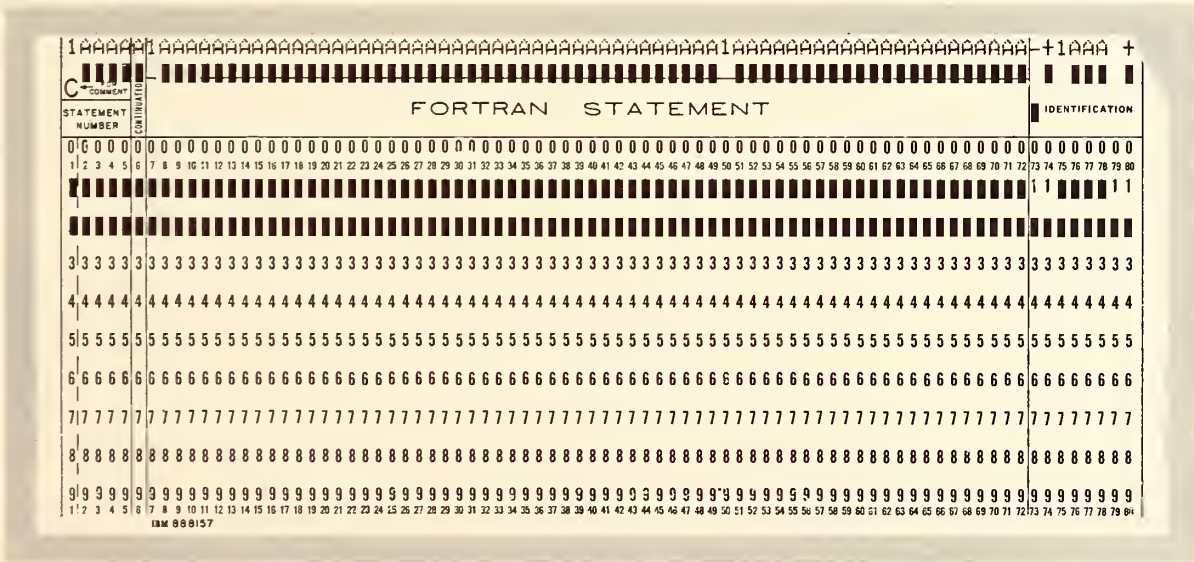


Figure 1-4 Program card for DYNAMO.

For convenience the cards may be numbered in columns 75 through 80.

The author will not try to explain how to operate a card punch. The fastest way to learn is to find someone to demonstrate how to use one. M.I.T. students and staff can seek help in either the M.I.T. Computation Center or the School of Industrial Management Computer Facility. The IBM manual for the card punch, number A24-0520, also is a source of information, but does not differentiate between necessary information and unimportant details for the novice.

A program card (to simplify punching the information into the proper columns) is shown in Figure 1-4. It places the machine in alphabetic shift through column 78. It provides skip stops in columns 7 and 50 (for comments). When the card being punched reaches column 73, it will automatically skip to column 75. Columns 75 through 78 may be either punched (using the NUM button for any numerals) or copied from the previous card by pressing the DUP button. Columns 79 and 80 may be punched with numerical information without using the NUM button.

The program card may be punched by passing it through the card punch twice. The first time the letters and numbers shown across the top of the card in Figure 1-4 should be punched. The second time a 2 should be punched in every column. To have the card look exactly like that in Figure 1-4, the print switch must be turned off while punching the 2's.

1.6 Assembling a Run

This section explains the few requirements on the order of the cards, and the method for obtaining several runs of a model with different constants.

Order of Cards

DYNAMO is quite liberal about the order of the cards. The only requirement is that the identification card must be first and the RUN card second. (Also, continuation cards must be in order and must follow the card they continue.)

This is an appropriate time to introduce one last card, the NOTE card. The NOTE card permits the user to identify portions of his deck or to add any comments that help clarify the model. NOTE is punched in columns 1 through 4, and the comment may start in column 7 or later. A NOTE card with nothing but NOTE punched in it will result in a blank line in the listing DYNAMO makes of the model.

Figure 1-5 is a listing of the model that we have been discussing, with the cards in an order suitable for DYNAMO.

Reruns

Frequently a DYNAMO user will want to run a model several times with different values for one or more constants. He can do this by following the basic run with another RUN card and one or more constant cards with the new values of the constants. The RUN card must be first, and only those constants that were originally specified by C cards may be altered. For example, in order to rerun the retail store model with a larger step input and longer delay in delivery of goods to the retail store, the cards would be

```

RUN      2699JP
NOTE    LARGER STEP INPUT AND LONGER DELAY IN TRANSIT
C       STH=200 ITEMS/WK
C       DTR=4 WKS

```

As can be seen in this example, NOTE cards may also be included.

If several reruns are made, all constants that were modified in the previous rerun are first restored to the values they had in the original run. Thus, if the user wished to make a second rerun with the same value of STH as in the rerun shown, he would have to include another C card for it in the second rerun.

```
*      M478-248,DYN,TEST,1,1,0,0
RUN    2698JP
NOTE   MODEL OF RETAIL STORE
NOTE
1L     IAR.K=IAR.J+(DT)(SRR.JK-SSR.JK)      INVENTORY ACTUAL
1L     UOR.K=UOR.J+(DT)(RRR.JK-SSR.JK)      UNFILLED ORDERS
20A    NIR.K=IAR.K/DT                       NEGATIVE INVENTORY
20A    STR.K=UOR.K/DFR                     SHIPMENTS TRIED
54R    SSR.KL=MIN(STR.K,NIR.K)             SHIPMENTS SENT
40R    PSR.KL=RRR.JK+(1/DIR)(IDR.K-IAR.K)   PURCHASE ORDERS SENT
12A    IDR.K=(AIR)(RSR.K)                 INVENTORY DESIRED
3L     RSR.K=RSR.J+(DT)(1/DRR)(RRR.JK-RSR.J) REQUISITIONS SMOOTHED
39R    SRR.KL=DELAY3(PSR.JK,DTR)          SHIPMENTS RECEIVED
NOTE
NOTE   INITIAL CONDITIONS
NOTE
12N    UOR=(DFR)(RRR)
6N     RSR=RRR
6N     IAR=IDR
NOTE
NOTE   INPUT
NOTE
7R     RRR.KL=RRI+RCR.K                   REQUISITIONS RECEIVED
45A    RCR.K=STEP(STH,5)                 REQUISITION CHANGE
NOTE
NOTE   CONSTANTS
NOTE
C      AIR=8 WKS                          CONSTANT FOR INVENTORY
C      DFR=1 WK                            DELAY IN FILLING ORDERS
C      DIR=4 WKS                           DLY REFILLING INVENTORY
C      DRR=8 WKS                           REQUISITION SMTHNG T C
C      DTR=2 WKS                           DELAY IN TRANSIT
C      RRI=1000 ITEMS/WK                   REQ. RECEIVED INITIALLY
C      STH=100 ITEMS/WK                   STEP HEIGHT
NOTE
PRINT  1)IAR,IDR/2)UOR/3)RRR,SSR/4)PSR,SRR
PLOT   IAR=I,UOR=U/RRR=R,SSR=S,PSR=P,SRR=Q
SPEC   DT=0.1/LENGTH=10/PRTPER=5/PLTPER=0
```

Figure 1-5 Listing of model.

1.7 Error Examples

DYNAMO checks every model extensively. It will not run a model unless the model is consistent. The checks do not guarantee that the model is what the user intended, but most of the errors will be caught.

The following are a few examples of the types of errors that are frequently made, followed by brief explanations.

- DYN EXPECTED) WHERE FOUND CHAR. INDICATED

$$12A \quad IDR.K = (AIR) (RSR.K)^V$$

should have been

$$12A \quad IDR.K = (AIR) (RSR.K)$$

- QUANTITY TYPE AND TIME-SUBSCRIPT ON QUANTITY DEFINED DO NOT AGREE

$$20A \quad NIR.KL = IAR.K / DT$$

should have been

$$20A \quad NIR.K = IAR.K / DT$$

- DYN EXPECTED (WHERE FOUND CHAR. INDICATED

$$12A \quad STR.K = UOR.K / DFR^V$$

This comment is the consequence of DYNAMO's assumption that the equation form number was right. The correct equation form number is 20, and the card should read

$$20A \quad STR.K = UOR.K / DFR$$

- DYN EXPECTED + WHERE FOUND CHAR. INDICATED

$$40R \quad PSR.KL = RRR.JK - (1/DIR) (IAR.K - IDR.K)^V$$

This sign may not be changed to -, this equation must be written

$$40R \quad PSR.KL = RRR.JK + (1/DIR) (-IAR.K + IDR.K)$$

or

$$40R \quad PSR.KL = RRR.JK + (1/DIR) (IDR.K - IAR.K)$$

- TOO MANY CHARACTERS IN (DFR)(

$$20N \quad UOR = (DFR) (RRR)$$

Unfortunately, this is a rather obscure explanation of a simple error. The error is that the equation form number should be 12 rather than 20. As above, DYNAMO assumed that the equation form number was right and tried to find a quantity name between the = and a /. As it could not find a / before it found six other characters, it printed out that there are too many characters in what it thought was a quantity name. The corrected card is

$$12N \quad UOR = (DFR) (RRR)$$

- PSR HAS UNALLOWED SUBSCRIPT

$$39R \quad SRR.KL = DELAY3(PSR.KL, DTR)$$

should have been

```
39R   SRR.KL=DELAY3(PSR.JK.DTR)
```

- STR DEFINED TWICE

```
20A   STR.K=UOR.K/DFR
```

There are two cards that define STR. DYNAMO has printed the second. One must be removed. (An initial-condition equation for STR is not considered a double definition, although two N equations for STR would be.)

- IAR REQUIRES INITIAL VALUE

The N equation for IAR was left out.

- NER IS UNDEFINED IN EQN FOR SSR

Equation for SSR was mispunched with NER rather than NIR, and consequently NER appeared to be an undefined variable.

- RCR IS UNDEFINED IN EQN FOR RRR

Card defining RCR was forgotten.

- SIMULTANEOUS EQNS AMONG N EQN

```
PSR
SRR
RRR
RSR
IDR
IAR
```

This error was contrived by setting RRR equal to SRR initially. When the user encounters such an error comment, he must examine the "ring" of simultaneous equations to see where it can be broken. If the simultaneity is among N equations, he will probably discover that some of them were created by DYNAMO out of A or R equations. Generally, he will find a variable in the ring to which he can easily assign an initial value. In the example, eliminating the equation for RRR will force DYNAMO to create one that will be satisfactory.

Chapter 2

REFERENCE MANUAL

2.1 Introduction

Although DYNAMO is simple to use, its rules must be followed carefully. This chapter presents the rules in a complete and precise manner.

It is recommended that anyone reading this chapter for the first time ignore all references to boxcars and boxcar trains until Section 2.5, which describes them in detail. Their inclusion in the earlier sections is for completeness.

Two types of punched cards are used by DYNAMO in generating a running code: equation cards and direction cards. The equation cards describe the model and the numerical values; the direction cards specify the length of run, the variables to be printed and plotted, and so forth.

2.2 Equation Cards

In brief, equations for a model are written by

- Deciding upon the quantity type[†] of the quantity being computed
- Deciding which quantities (variables and constants) determine this quantity
- Deciding upon the algebraic form of this relationship
- Looking up this algebraic form in the appendix for the equation form number and the exact punching format of the equation

Quantity Types

The following seven quantity types are recognized by DYNAMO:

L	Level
A	Auxiliary
R	Rate
C	given Constant
N	iNitial value or iNitially computed constant
B	Boxcar train
S	Supplementary

A Level is a quantity whose value at a given time is computed from its value at the previous time that calculations were made and from its rate of change during the time interval between calculations.

An Auxiliary is a quantity introduced to simplify the set of equations or

[†] Quantity type is perhaps not as suggestive a term as desirable. Earlier the name "variable type" was used, but this term was poor because constants are included in quantity type.

because a particular equation format is not available; it is auxiliary in the sense that it could be eliminated by substitution into the rate equations.

A Rate is a quantity whose value indicates a rate of flow during a time-step interval.

A given Constant is a quantity whose numerical value is given explicitly and is held constant for a particular run. (Its value may be changed in succeeding runs.)

Initial values are specified for various quantities at the start of a run. (Section "Initial Values," pages 24 and 25, describes in detail those circumstances where initial values are required.)

Initially computed constants are constants that have a direct relationship to other (given or computed) constants and are computed from them at the beginning of the run.

A Boxcar train is a set of quantities that are all shifted down one boxcar position at specified shift times. (Section 2.5 discusses boxcar trains in detail.)

A Supplementary quantity is one which is extraneous to the set of equations proper and which is introduced only as a quantity to be printed or plotted. It is computed only at the output times.

Quantity Names

Each quantity must be given a name of five or fewer alphabetic or numerical characters (no hyphens, slashes, etc.), the first character of which must be alphabetic, for example, X, INV, or PROD1.

Equation Forms

After the algebraic relationship has been decided, Appendix B is checked for the equation form number and the exact punching format of the equation.

This appendix will be described by discussing its first entry:

General Form		Exact Punching Format
$V = V + (DT)(P + Q)$	1L	$V \bullet K = V \bullet J + (DT) (\pm P \pm Q)$

The number in the middle is the equation form number. The letter following it indicates certain restrictions on the use of that equation form. The "General Form" gives the equation form in its familiar algebraic format. As most of the signs are flexible (as will be seen shortly), they are all given as plus. The "Exact Punching Format" gives the precise order of quantities and algebraic symbols that must be used for DYNAMO.

The letters following the equation form number in Appendix B place the following limitations on the equation form usage:

A or R	may be used to define only auxiliaries or rates.
B	may be used to define only boxcar trains. Also, no other equation types may be used to define boxcar trains.
L	may be used to define only levels. Furthermore, levels may be defined by only so-marked equation forms.
N	may be used to define only initial values.

R	may be used to define only rates.
no letter	may be used to define auxiliaries, rates, computed constants, initial values, or supplementaries, that is, any quantity type except level or boxcar train.

The letters used in the "General Form" and the "Exact Punching Format" indicate the kinds of quantities which may be used in that particular equation form:

V	the quantity being defined, appearing on the left of the equality sign. It of course must be a quantity name.
P through U } W through Z }	may be either a quantity name [†] or a numerical value. [‡]
N, N1, N2	must be a Numerical value.
C, C1, C2	must be either a Constant (given or computed) or a numerical value.

The equation forms must be used exactly as shown in the "Exact Punching Format." For example, the sum of two quantities divided by a third must be punched

$$X = (1/Y) (P+Q)$$

as shown in Equation Form 18 and not

$$X = (P+Q) / Y$$

or

$$X = (P+Q) / (Y)$$

or

$$X = (P+Q) (1/Y)$$

although they are algebraically equivalent. However, there is wide latitude with respect to the algebraic sign of the quantities used in the equation forms. Any place that a "+" appears in the equation form, either sign may be used. Furthermore, as in high-school algebra, the plus sign may be omitted after an equality sign, a left parenthesis, a comma, or a slash. For example,

$$V = +P + (+Q) (+R)$$

and

$$V = P + (Q) (R)$$

are both equally acceptable to DYNAMO.

[†] Further restrictions will be discussed in the section, "Subscripts and Permissible Defining Quantities," page 23.

[‡] The permissible formats of numerical values are described on page 26.

Occasionally a desired equation form cannot be found in Appendix B. This may occur because the desired equation form is more complicated than any of those provided or because an almost satisfactory form contains too many factors or terms. The latter situation can generally be taken care of by setting the extra factors equal to 1 or the extra terms equal to 0. For example, the sum of five terms can be constructed from Equation Form 10 as follows:

$$10 \quad V=P+Q+R+S+T+0.0$$

and the ratio of two factors to two factors can be constructed from Equation Form 46 as follows:

$$46 \quad V=(P)(Q)(1)/((R)(S)(1))$$

If the equation form desired is not available because the equation is too complicated, the equation can generally be constructed from two or more existing equation forms by defining extra quantities. For example, the equation form

$$V=(P)(Q)/(R+S+T)$$

is not available but can be constructed from Equation Forms 44 and 8:

$$\begin{array}{ll} 44 & V=(P)(Q)/U \\ 8 & U=R+S+T \end{array}$$

Subscripts and Permissible Defining Quantities

Quantity names are given a time subscript to remind the user of the type of quantity and the order of computation. (As standard card-punching equipment does not permit the usual subscript notation, the time subscript becomes a postscript set off from the quantity name by a decimal point; for example, X.K denotes the quantity X at time K.) Levels, auxiliaries, and supplementaries have a single-letter subscript, for instance, LEV.J, AUX.K. Rates always have a double-letter subscript, for example, RATE1.JK, RATE2.KL. Both given and computed constants have no subscript. Boxcars have a single-letter subscript. Boxcar trains have no subscript when being defined (by a B equation) but are given the subscript .K when used on the right side of Equation Forms 53, 55, or 57. Initial values have no subscripts.

Since the present time at which a level is calculated is defined as the time instant K, and the forthcoming time interval for which a rate is calculated is defined as the time K to L, the variable quantity being defined will always be given the subscript K or KL depending on the quantity type, for example, LEV.K, AUX.K, RATE.KL, SUPL.K.

Levels may be based on constants, levels or auxiliaries from the previous time instant, or rates from the previous time interval. Therefore, the quantities on the right of the equality sign in level equations must have either no subscript (indicating a constant) or the subscripts .J or .JK (indicating a variable). Auxiliaries may be based on constants (no subscripts), levels or other auxiliaries at the same time instant .K, or rates from the previous time interval .JK. Rates may be based on constants (no subscripts), levels or auxiliaries at the present instant .K, or rates from the previous time interval .JK. Supplementaries may

be based on constants (no subscripts), levels, auxiliaries, or supplementaries at the present time instant K, or rates from the previous interval JK. Note that a supplementary may appear only on the right side of another supplementary equation.

Computed constants may be based on given constants and other computed constants.† Initial values may be based on given or computed constants or other initial values, except the initial value of boxcars. Thus neither side of an N equation has any quantities with subscripts.

Table 2.1 summarizes these subscript conventions.

Table 2-1 Subscript Table

Quantity Type on Left of Equation		Subscript on Left	Subscripts on Quantities on Right if Quantity is					
			L	A	R	S	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
S	Supplementary	K	K	K	JK	K	none	none
C	Constant	none	—	—	—	—	—	—
N	{Initial value or computed constant	none	—	—	—	—	none	none

Note that every quantity that appears on the right side of any equation must be defined, that is, appear on the left of that or some other equation. (The easiest way to check this is to let DYNAMO check it.) There are two special cases of this rule: DT and TIME. DT is specified on the SPEC card (see page 28) and may be used as a constant on the right of any equation. TIME is a true exception. It is not defined anywhere but rather is built into DYNAMO. It may be used as a level on the right of any equation where levels are permitted.

Initial Values

In order to get a model started, some variables will require initial values. Some of these initial values can be computed by DYNAMO; others will require explicit values provided by the user.

All Levels require initial values that the user must provide. If the level is not a boxcar, the initial value will be specified by an N-type equation. If the level is also a boxcar, the initial value will be specified with the initial values for the boxcar train.

Every Boxcar Train requires initial values specified by a C card (see page 27) or by the BOXLOAD equation (see pages 33-34).

No Supplementary ever requires an initial value, and the inclusion of one in a model is an error recognized by DYNAMO.

† DYNAMO also permits computed constants to be based on initial values (except the value of boxcars), but this flexibility will be of little use to most users.

DYNAMO will provide initial values for those auxiliaries and rates that require them by simply repeating the auxiliary or rate equation as an N-equation. This frequently leads to additional auxiliaries and rates requiring initial values as now they appear on the right side of an initial-value equation.† DYNAMO continues the process of generating N equations from A and R type equations until all the required initial values are specified or until DYNAMO finds a set of simultaneous equations. If this occurs, DYNAMO prints out that fact along with a list of the variables involved in the simultaneous solution.

In such cases the user should select a variable from the list and write an N-equation for it that does not involve any of the other quantities in the list.

An example of the process of creating initial-value equations is given in Chapter 3.

Equation Card Punching

Equation cards are created by punching the equation form number starting in column 1, followed without any intervening spaces by the letter corresponding to the quantity type. The actual equation starts in column 7 and may extend no farther than column 72. If more space is required, continuation card(s) must be used. As pointed out earlier, the exact format indicated in Appendix B must be followed, and the time subscripts must be correct. An identifying serial number for the equation may be punched in columns 75 through 80. For example:

COLUMN	123456789.....567898 ⁰
1L	UO.K=UO.J+(DT)(OR.JK-DL.JK)	JP0027
12N	UO=(DEL)(OR)	JP0028
20R	DL.KL=UO.K/DEL	JP0029

Rule of the Final Space. No spaces may be inserted in the equation, but the equation must be followed by at least one space to indicate the end of the equation. If the last character of an equation should fall in column 72, a continuation card (see below) must be used to provide the final space. The same rule applies to the information provided on direction cards (with the one exception of the NOTE card). No spaces may be inserted within the information, but a "final" space must be provided to indicate the end of the information.

Continuation Cards. No equation or direction card information may extend beyond column 72. If the material will not fit, continuation card(s) must be used. Continuation cards are identified by an X in column 1 followed by the number of the continuation card, for example, X1, X2, and so on. The continuation cards must be inserted in ascending order immediately following the original

† Auxiliaries that appear in some initial-value equation and rates that are used on the right side of auxiliary, rate, supplementary, or initial-value equations must have an initial-value equation provided either by the user or DYNAMO.

Third-Order Delays are a special case. The set of equations that make up a third-order delay (see footnote, page 36) contains three initial-value equations that have the input on the right. Thus DYNAMO or the user must provide an initial value for the input. But note that one of those equations is the initial value for the output. Consequently, no initial value is ever required for the output of a third-order delay, and the inclusion of one is an error.

card and must contain the same identifying information in columns 75 through 80 as the original card. This is one of the few places where card order is important in DYNAMO. (Card order is discussed further on page 32.) For example, if the user is employing Equation Form 17,

$$V=(P)(Q)(R)+(S)(T)(U)+(W)(X)(Y)$$

and if the last term is (ABC)(XYZ.K)(TUV.K) with Z falling in column 72, the continuation card is

```
COLUMN 123456789.....
        X1      .K)(TUV.K)
```

Numerical Values. Numerical values on equation or direction cards can be expressed by up to 8 digits, in addition to sign and decimal point. A decimal point on the right of a number may be dropped; and a plus sign following an equality sign, a left parenthesis, a comma, or a slash can be dropped. Examples of several different numerical values are

```
37
+42600.
-0.075
```

Eight digits may be unsatisfactory in expressing very large or very small numbers. For this reason the number expressed may be multiplied by some power of 10 by immediately following the number with the letter E and the power of 10. When the sign is plus, it may be dropped. For example, 82 billion (82×10^9) can be written

82E+9 OR 82E9

and 1 ten-thousandth (1×10^{-4}) can be written

1E-4

The ultimate limit on the size of number that may be used is that the magnitude of the difference between the power of 10 and the number of digits to the right of the decimal point must not exceed 29.

Comments. To save searching through one's notes, it is often convenient to have a description of a quantity on the card that defines that quantity. Such a comment may be added to an equation card (or any card for that matter) following the "final space." These comments are listed along with the equations at the beginning of every run. Standardizing the location where the comment begins (if the length of the equation permits) helps the readability of such comments. The comments may not extend beyond column 72 and may not be continued on a continuation card. If it is desirable to continue them, they may be continued on a NOTE card. See page 31 for NOTE cards. For example,

```
COLUMN 123456789.....
        1L      UO.K=UO.J+(DT)(OR.JK-DL.JK)      UNFILLED ORDERS
```

Constant Cards

Given constants are defined on special equation cards that have no equation form number. Only the letter "C" appears in column 1. The quantity name

of the constant starts in column 7, followed by an equality sign and the desired numerical value. For example.

```
COLUMN 123456789.....
C      ABC=3.0
```

More than one constant may be defined on a single card by separating them with slashes. For example,

```
COLUMN 123456789.....
C      ABC=3.0/BCD=32/CCC=12E6
```

No spaces may be included before the "final space," and a slash after the last constant (before the final space) is an error. The practice of including several constants on a single card has the disadvantage of requiring the repunching of a long, complicated card to change one constant.

Tables and Initial Values for Boxcar Trains.[†] The contents of tables are specified by C cards, as may be the initial values of boxcar trains. In this case the quantity name is followed by an asterisk and then the equality sign. The numerical values follow the equality sign in order and are separated by slashes. The final numerical value is followed by the final space (no slash). For example,

```
COLUMN 123456789.....
C      TRN*=0/1/2/3
```

Only one table may be defined on a C card. Any number of continuation cards may be used (see page 25). The user is cautioned to include an asterisk after the name of the table or boxcar train and to separate the numbers with slashes rather than commas.

2.3 Direction Cards

Direction cards specify such values as the length of run, the printing and plotting intervals, and the quantities to be printed and plotted. They include Identification, RUN, SPEC, PRINT, PLOT, NOTE, and NOISE cards.

Identification Card

The identification card,[‡] which is the first card in a deck,^{††} provides the Computation Center with such basic information as to whom the deck belongs, what kind of a deck it is (DYNAMO), and how long it will run. It has been fashioned after the standard identification card used by the M.I.T. Computation Center and contains the following items:

[†] It is recommended that this section be omitted by anyone unfamiliar with boxcar trains (page 32) and the table function (page 38).

[‡] At installations other than M.I.T., the user must make an identification card that is similar to the one to be described. The problem number and the programmer's number may be replaced with any alpha-numeric information, but the minus sign and comma preceding DYN must be retained.

^{††} The order of the cards is discussed in detail on page 32.

1. The problem number (assigned by the Computation Center)
2. The programmer's number (also assigned by the Computation Center)
3. DYN to indicate that this is a DYNAMO run
4. Either the word TEST, to indicate that the model is still being developed and checked out, or the word RESULT, to indicate that the model is checked out and this run will probably be a part of the final results of the research
5. The number of minutes that the run (and reruns if any) is expected to take
6. The maximum number of minutes that the computer operator should allow the run (and reruns) to take. (This time limit protects the user from using an excessive amount of time should he make an error in the time required for this run. See page 43 for estimating the time.)
7. 0,0 (required to conform to M.I.T. Computation Center procedures)

The permissible format for this information is quite flexible, but the following specifications are consistent with the rest of DYNAMO and therefore recommended.

An asterisk must be punched in column 1. The above information is supplied in the order given, starting in column 7. The problem number is separated from the programmer's number by a minus sign, and other pieces of information are separated by commas. The two zeros are followed by the "final space," but no space may be included within the information. Following the final space, any comment may be added. For example:

```
COLUMN 123456789.....
*      M478-248,DYN,TEST,2,3,0,0      GROWTH MODEL II
```

RUN Card

Each run (or rerun) is assigned a "run number" by which it can be identified and filed. This number appears at the top of each page along with the page number. The number is provided by a card with RUN in columns 1, 2, and 3 and the run number in columns 7 through 12. Any characters may be used. Each group of DYNAMO users will prescribe how these numbers will be generated. Example:

```
COLUMN 123456789.....
RUN    1267JP
```

SPEC Card

The SPEC card specifies the four basic data necessary for any run:

1. DT the interval of time between J and K (see Section 2.7)
2. LENGTH the length of time for which the model should be run
3. PRTPER the interval of time between tabulating the results of the model
4. PLTPER the interval of time between plotting the results of the model

The details of the punching are most easily understood by reference to the following example:

```
COLUMN 123456789.....
SPEC DT=0.1/LENGTH=100/PRTPER=3/PLTPER=1
```

The order of the four items must be as shown, and all must be included. The rules for the numerical values are on page 26. No space may be included in the

information, and the rule of the final space applies. Note that there is no slash immediately preceding the final space.

For internal reasons in DYNAMO, the larger of PRTPER and PLTPER should be an integral multiple of the smaller. For example, PRTPER might be chosen two or three times PLTPER, but should not be chosen 2.5 times PLTPER. In addition, both PRTPER and PLTPER should be an integral multiple of DT.

Two special features in DYNAMO are controlled by the SPEC card. Either printing or plotting may be suppressed, even though the model includes both PRINT card(s) and PLOT card(s), by setting the appropriate one of PRTPER or PLTPER equal to 0.

The other feature is that the original run may be suppressed by omitting the SPEC card until the first rerun. This is useful if the model builder wishes to keep a basic model and make all runs (actually reruns) as modifications of it. Of course only constants may be modified in these reruns.

PRINT Cards

DYNAMO can print any quantity desired in the tabular form shown in Figure 2-1. If the quantity is a level, auxiliary, or supplementary, the value corresponds to the instant .K; if the quantity is a rate, to the interval .KL.

TIME	LEV AUX	INV	BXCR*1
E+00	E+03 E-06	E+06	E+03
.00	2000.0 -.0127	25.000	87.
1.00	1973.8 -.0138	25.724	88.

Figure 2-1 Sample of DYNAMO tabular output.

TIME is automatically printed by itself in the leftmost column. The user may specify up to 10 quantities in each of the 14 remaining columns. The name of the quantity being tabulated is automatically printed at the top of each page. The scale factors (see next page) are printed on the first page under the names. The data follow with dashes separating the data corresponding to different times.

Quantities that are to be printed are specified according to the column in which they are to appear. The column number (1 through 14) is given followed by a right parenthesis. The quantities to appear in that column follow in order, separated by commas. (If a blank is desired in some line, an asterisk is used in place of a quantity name. Every asterisk must be counted as one of the ten entries permitted in a column.) Information pertaining to one column is separated from information for another by a slash, but no slash is used following the word PRINT or preceding the final space. For example, the tabular form shown above might have been the result of the following PRINT card:

```
COLUMN 123456789.....
PRINT 1)LEV,AUX/2)INV/3)*,BXCR*1
```

No space may be included within the information, and the rule of the final space applies. If all the information will not fit on one PRINT card, continuation cards (see page 25) may be used or the information may be subdivided to fit on several PRINT cards (with a final space on each). If there are several references to a particular column, the quantities listed first are printed above those listed later. The order in which the columns are specified is arbitrary.

Scaling of Quantities for Printing. As DYNAMO prints but 5 significant figures, very large or very small numbers must be divided by some power of 10 before printing so that the 5 significant figures will suffice. Such division before printing is called scaling.

These scale factors are printed on the first page following the quantity names. They appear as the letter E followed by the power of 10. As the results were divided by this power of 10 before printing, they must be multiplied by this power of 10 in order to reconstruct the original number. For example, LEV in Figure 2-1 has a scale factor of E+03, which means that the value of 2000.0 indicated for time 0 should be multiplied by 10^{+3} or 1000 to reconstruct the initial value of LEV. Thus the value of LEV was 2 million at TIME equal to 0.

If desired, DYNAMO will calculate these scale factors and the position of the decimal point. However, some users prefer to ensure that the scale factors and positions of the decimal point do not vary between runs. This can be achieved by following the quantity name with the power of 10 by which the quantity should be divided before printing and with the number of decimal places that are desired to the right of the decimal point. These numbers should be enclosed in parentheses and separated by a decimal point. If the power of 10 is positive, the sign may be omitted. As an example, if every quantity in Figure 2-1 had its scale factor specified, the PRINT card (with the columns deliberately taken out of order) might have been

```
COLUMN 123456789.....
PRINT 3)*,BXCR*1(3.0)/2)INV(+6.3)/1)LEV(3.1),AUX(-6.4)
```

If a quantity for which the DYNAMO user has specified the scaling is too large for that scale, DYNAMO will rescale the number and replace the fifth digit with a letter corresponding to the new scaling. Appendix A lists the scaling corresponding to these letters.

PLOT Cards

In addition to tabulating the results, DYNAMO will plot the results (versus TIME) using the same printer as for the tabulation. (Figure 1-3, page 13 is an example.) Much flexibility is available in these plots. The user specifies which quantities shall have the same scales and has the option of either specifying the scales or letting DYNAMO choose the scales so that all the data are on scale.

Each quantity name is followed by an equality sign and the character that is to identify it on the graph. Quantities that are to be plotted to the same scale are separated by commas. Quantities or groups of quantities that are to be plotted to different scales are separated by slashes. For example,

COLUMN 123456789.....
PLOT RPQ=R,TUV=T/XYZ=X

In the preceding example, RPQ and TUV are plotted on the same scale, while XYZ is plotted on its own scale. If the DYNAMO user wishes to specify the scale, he follows the quantity name and plotting character (of the last quantity in the group with the same scale) by a left parenthesis, the lower limit, comma, upper limit, and a right parenthesis. If he would like DYNAMO to set one of the two limits, he puts an asterisk in place of that limit value. For example,

COLUMN 123456789.....
PLOT AUX=A,ABC=C/XYZ=X(0,*)/TUV=T,BXCR*3=B(0,1000)

In this example, AUX and ABC are plotted with the same scale limits, and these limits are set by DYNAMO. XYZ is plotted on its own scale, and while the lower limit is 0, the upper limit is chosen by DYNAMO. TUV and BXCR*3 are plotted on a scale running from 0 to 1000.

More than one plot may be specified by using additional PLOT cards. No more than 10 quantities may be specified on any one plot,[†] but a particular quantity may be specified on any number of plots. No spaces may be included within the information, and the rule of the final space applies (see page 25). Continuation cards may be used.

NOTE Cards

DYNAMO prints all the equation cards and direction cards preceding the printed and plotted results so that a DYNAMO user can be certain which set of equations produced the results. It is sometimes convenient to insert a line of remarks or comments along with the equation cards (in addition to the comments on the cards proper mentioned on page 26). Such remarks may be punched on a NOTE card, starting in column 7 or later and ending before column 73. Any number of spaces may be used. No continuation cards are permitted, but a remark may be continued on additional NOTE cards. A NOTE card is identified by "NOTE" punched in columns 1 through 4. The word NOTE will be omitted when DYNAMO prints the NOTE card.

COLUMN 123456789.....
NOTE MANUFACTURING SECTOR

NOISE Cards

As described on page 35, the random numbers in DYNAMO are generated from a sequence of "pseudorandom numbers." This sequence will always start at the same point unless a NOISE card is included in the run or rerun. The

[†] There is also a limit to the space available to store plotting specifications:

$$2x + 3m + n \leq 200$$

where

- x = total number of variables to be plotted
- m = total number of scales
- n = number of PLOT cards (not counting continuation cards)

NOISE card is punched by punching "NOISE=" in columns 1 through 6 followed by 12 octal digits in columns 7 through 18. The first digit must be less than 4 and the last must be odd.† For example,

```
COLUMN 123456789.....
NOISE=300336163251
```

2.4 Order of Cards; Runs and Reruns

A basic run consists of an Identification card followed immediately by a RUN card and the model (Equation cards, PRINT cards, PLOT cards, and so on). Within the model the order of the cards is arbitrary except for continuation cards (see page 25). DYNAMO automatically sets up the order of computation so that any quantity which depends on any other quantity at the same time step is computed after the other quantity.

Quantities that depend on other quantities of the same type at the same time step can occur only among the auxiliaries and supplementaries and among the N-type equations (initial values and computed constants).‡ If DYNAMO cannot reorder the computation to avoid a simultaneous solution, DYNAMO prints out SIMULTANEOUS EQNS AMONG and lists the quantities it cannot satisfactorily reorder.

If it is desired, the same model may be rerun one or more times with different values of any of the given constants (or with a different initial value of NOISE) without repeating the basic model specification or making the computer regenerate the code for the model. Only C cards and direction cards (except the identification card) may be used. If a PRINT card is included, the whole previous PRINT request is erased. The same is true of a PLOT card and the entire previous PLOT request.

The order of the cards in any rerun is arbitrary except that the RUN card must be first and any continuation cards must be in their usual order.

If more than one rerun is included, all given constants are changed back to the values they had in the first run before any new changes are recorded. Thus if ABC, a given constant, is 10 in the basic run and is to be 20 in two reruns, both reruns must contain C cards that set it equal to 20.

On the other hand, the PRINT and PLOT requests and the constants on a SPEC card are based on the most recent alteration; that is, they are not restored to their initial condition in the first run.

2.5 Boxcars and Boxcar Trains

To permit the storage and later use of historical data, two "boxcar train" functions have been included in DYNAMO. (Boxcar is not a very descriptive term, but usage forces its continuance.) A boxcar train consists of a series of boxcars, or cells, where information can be stored chronologically. Periodically, the contents of the boxcars are shifted,†† the contents of the first becoming the

† A number of values are recommended in the Industrial Dynamics Memo D-36, "Random Number Generators Used in DYNAMO," Phyllis Fox, M.I.T., April 22, 1959.

‡ Two special problems with reordering are discussed on page 36 and page 40.

†† The first shift occurs at TIME = 0.

contents of the second, the contents of the second becoming the contents of the third, and so on. There are two choices for contents of the first boxcar following a shift: the former contents of the last boxcar (cyclic shift) or zero (linear shift). Figure 2-2 shows the two shifts diagrammatically. The cyclic shift is useful for keeping (and updating) seasonal data, while the linear shift provides a convenient way to construct pipeline delays.

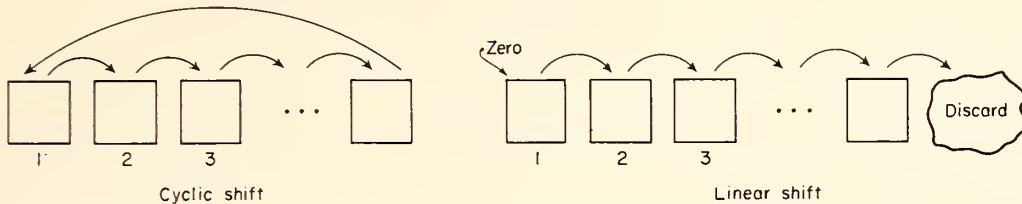


Figure 2-2 Types of shifts.

The name of a train is chosen according to the same rule as any other quantity name, that is, 5 or fewer alphabetic or numerical characters, the first of which is alphabetic. A particular boxcar in a train is specified by following the name of the train with an asterisk and the number of the boxcar desired. For example, the third boxcar in the train called ABCD is ABCD*3. If the boxcar has a time subscript (as it generally does), the subscript will follow the number, for example, ABCD*3.K.

The method for transferring data from a train to another part of a model is to use the name of the appropriate boxcar on the right of any equation except an N-equation. For example,

```
COLUMN 123456789.....
12A    AUX.K=(C)(ABCD*7.K)
```

Data are entered into the train by writing a level (or occasionally an auxiliary) equation for the first (or any) boxcar. For example,

```
COLUMN 123456789.....
1L     ABCD*1.K=ABCD*1.J+(DT)(RATE.JK+0)
```

The two boxcar functions that create the train and specify the number of cars and the frequency of shifting are:

```
35B    V=BOXCYC(N,P)           (CYCLIC SHIFT)
37B    V=BOXLIN(N,P)          (LINEAR SHIFT)
```

where V is the name of the boxcar train and N is the number of boxcars. N must be a numerical value and may not exceed 31. The shifting interval P may be either a numerical value or a quantity name.

Every boxcar train must have initial values specified for each boxcar. These values may be specified by either a C card or the BOXLOAD equation type. In the case of the C card (see page 26), the number of initial values specified must agree with the number of boxcars in the train.

If every car in the train is to be given the same initial value, the BOXLOAD may be used:

36N V=BOXLOAD($\pm P, \pm Q$)

where V is the name of the train to be loaded. Each car in V will be given the initial value of the product of P and Q. Note that the BOXLOAD equation type does not require the inclusion of the number of boxcars in the train.

(Caution: If the initial value of the train is a C card, an asterisk must follow the name of the train on the C card, but if the BOXLOAD function is used, no asterisk is used following the name on the N-equation card.)

As an example, a 20-week pipeline delay[†] will be constructed. In order to have a dynamic input whose characteristics require no explanation, TIME will be used for this example. It will be delayed in a train with 11 cars that are shifted linearly every 2 weeks. (Eleven cars are required because the first car will have roughly the current value of TIME, the second, TIME 2 weeks ago, and so on, until the eleventh car, which will contain TIME 20 weeks earlier.)

37B TRN=BOXLIN(11,2)

The average value of the input over the 2-week interval between shifts is to be entered into the first car. This can be calculated by a level equation that permits a 2-week divisor:

52L TRN*1.K=TRN*1.J+(DT)(1/2)(TIME.J+0)

The initial values of the train will be chosen to behave as though the train had been operating prior to TIME = 0. As the first shift will occur at TIME = 0, the first car must contain initially the average value of TIME for the interval -2 to 0 or -1. The second car will contain the average TIME for the previous interval (-4 to -2 or -3). Initial values will be similarly chosen for the remaining cars.

C TRN*=-1/-3/-5/-7/-9/-11/-13/-15/-17/-19/-21

TRN*11.K or TRN*11.J can be used wherever the delayed value of TIME is desired.

2.6 Special Functions

Common Functions

There are five common functions available in DYNAMO: exponential, natural logarithm, square root, sine, and cosine. All these functions behave exactly as described in elementary courses in calculus. The exact punching format of each function includes a multiplier P, which may be a variable or a constant. If this multiplier is not needed, the number 1 should be punched in place of the letter P.

28 V=($\pm P$)EXP($\pm Q$)

$$V = Pe^Q$$

29 V=($\pm P$)LOGN($\pm Q$)

$$V = P \ln Q$$

[†] A pipeline delay might also be called a perfect delay because a step input results in a step in the output rather than a gradual rise to the new level, as occurs in the first or third order delays. See Chapter 9 of Industrial Dynamics.

$$\begin{aligned}
 30 \quad V &= (\pm P) \text{SQRT}(\pm Q) \\
 &V = P\sqrt{Q} \\
 31 \quad V &= (\pm P) \text{SIN}((2\text{PI}) (\pm Q) / \pm R) \\
 &V = P \sin\left(\frac{2\pi Q}{R}\right) \\
 32 \quad V &= (\pm P) \text{COS}((2\text{PI}) (\pm Q) / \pm R) \\
 &V = P \cos\left(\frac{2\pi Q}{R}\right)
 \end{aligned}$$

The sine and cosine functions will frequently be used to generate functions of TIME. In such cases P will be the amplitude, TIME will replace Q, and R will be the period of the sine function. For example,

$$31A \quad \text{INPUT.K} = (\text{AMPL}) \text{SIN}((2\text{PI}) (\text{TIME.K}) / \text{PERD})$$

Random Numbers

DYNAMO generates random numbers by computing a sequence of "pseudo-random numbers." A sequence of pseudorandom numbers is a sequence of numbers which satisfies all statistical tests of randomness, but in which each number is calculated from the previous one. That is, if we know the generation procedure and one number in the sequence, we can calculate the next number in the sequence and thus all the numbers in the sequence, in complete violation of the principle of randomness. On the other hand, anyone unfamiliar with the generation procedure cannot detect that these numbers are not completely random.

There are two random-number functions in DYNAMO.

$$33 \quad V = (P) \text{NOISE}$$

gives V as random numbers uniformly distributed between $-P/2$ and $+P/2$. The numbers are not restricted to being integers and, in fact, will very seldom be integers.

$$34 \quad V = (\pm P) \text{NORMRN}(\pm Q, R)$$

gives V as the product of $\pm P$ times random numbers normally distributed (normal deviates) with mean $\pm Q$ and standard deviation R. NORMRN starts with a number from the same generator that Equation Form 33 above uses and then alters it to conform to a normal distribution by the method described in Memo D-36.[†] The alteration scheme does not produce a perfectly normal distribution. In particular, no numbers will exceed 2.4 standard deviations.

There are occasions when the user wishes the same sequence of random numbers and other occasions when he wishes a different sequence. To achieve either case, the following must be considered. In every DT, each equation using Equation Form 33 or 34 obtains the next random number in the sequence from the random-number generator (a different number for each equation). As DYNAMO restarts

[†] Op. cit., page 3 .

the sequence for each run or rerun (if there is no NOISE card), the sequence of numbers used by these equations will be the same. If DT is changed, the sequence will correspond to different times and will be essentially different. If the order or number of the equations utilizing Equation Form 33 or 34 in the model is changed between runs, the sequence used by a particular equation will be different. (If these equations are used to define auxiliaries, the DYNAMO user must be careful to make no changes in the model that cause DYNAMO, in the process of reordering auxiliaries, to order these equations differently from the earlier model.)

Thus, if the user wishes to run a model with the same sequence of random numbers but with different constants or equations, he must keep DT and the number and order of Equation Forms 33 and 34 the same. If, on the other hand, he wishes a different sequence of random numbers, he should use a NOISE card (see page 31) to alter the initial position in the sequence of random numbers.

Third-Order Delay

```
39R V=DELAY3(±P,C)
```

where

V = output rate from the delay
 P = input to the delay
 C = delay constant

The third-order delay is a shorthand notation for seven DYNAMO equations.[†] These seven equations are three cascaded first-order delays and their initial conditions.

The constant C must be either a given or computed constant. The input P must have an initial value. An N equation for V is included in the seven equations provided by DYNAMO. Thus if another N equation is included in the model for V, there will be an error recognized by DYNAMO. For example, if the output of one delay flows into another delay, an initial-value equation is required for the first delay, but the first delay (that is, DYNAMO) will provide the necessary initial-value equation for the input to the second.

Pulse Source

```
41(A OR R) V=PULSE(±P,Q,R)
```

[†] The seven equations are:

```
40R V.KL=V.JK+(1/*003)(*001.JK-V.JK)
40R *001.KL=*001.JK+(1/*003)(*002.JK-*001.JK)
40R *002.KL=*002.JK+(1/*003)(P.JK-*002.JK)
42N *003=C/((3)(DT))
6N V=P
6N *001=P
6N *002=P
```

where *001, *002, and *003 are names of quantities created by DYNAMO. (The input P was assumed to be a rate.)

where V is a pulse train in which the pulses are of width DT and height P .[†] The first pulse appears at time Q , and subsequent pulses appear at $\text{TIME} = Q + R$, $Q + 2R$, $Q + 3R$, and so on.

Note that P can be a variable and thus give a variable height to the pulses, and that Q and R , which determine the pulse spacing, may also be variables. If Q and R are variables, the pulse times are determined by the method described on page 41.

Step Function

$$45(A \text{ OR } R) \quad V = \text{STEP}(\pm P, Q)$$

$$\begin{aligned} V &= \text{initial value} && \text{if } \text{TIME} < Q \\ V &= P && \text{if } \text{TIME} \geq Q \end{aligned}$$

Both P and Q may be variables. If Q is a variable, the step time is determined by the method described on page 41. If P is a variable, the step function has the effect of a gate function that "opens" the gate allowing V to equal P after time Q . The value V will equal the initial value (or 0 if no initial value is provided) until the step time.

Ramp Function

$$47(A \text{ OR } R) \quad V = \text{RAMP}(\pm P, Q)$$

$$V = \text{initial value} \quad \text{if } \text{TIME} \leq Q$$

$$V = \text{initial value} + \sum_{Q+DT}^{\text{TIME}} (P)(DT) \quad \text{if } \text{TIME} > Q$$

If P and Q are both constants, V is merely a ramp with slope P that starts at time Q . If Q is a variable, the ramp time is determined by the method discussed on page 41. If P is a variable, V is DT times the sum of all P since the time Q (plus the initial value). The initial value will be assumed 0 unless given.

Sampler

$$43(A \text{ OR } R) \quad V = \text{SAMPLE}(\pm P, Q)$$

where V is set equal to P at sample times separated by intervals of length Q . If Q is a constant, these times are of course Q , $2Q$, $3Q$, and so on. Variable sample times are discussed on page 41. If an initial-value equation is provided for V , V will have that value until the first sample time. Otherwise, it will be 0 until the first sample time.

[†] A reader familiar with the term "impulse" should be careful to note that P is the height of the pulse and not the area under the pulse. If he wants a train of impulses of area P , he should use the combination

$$\begin{aligned} 20 \quad S &= P/DT \\ 41 \quad V &= \text{PULSE}(S, Q, R) \end{aligned}$$

with the appropriate variable types and subscripts.

Maximum Function

$$56 \quad V = \text{MAX}(\pm P, \pm Q)$$

$$V = P \quad \text{if } P \geq Q$$

$$V = Q \quad \text{if } P < Q$$

where P and Q need not be distinct. In particular, setting Q equal to -P sets V equal to the absolute value of P.

Minimum Function

$$54 \quad V = \text{MIN}(\pm P, \pm Q)$$

$$V = P \quad \text{if } P < Q$$

$$V = Q \quad \text{if } P \geq Q$$

Clipping or Limiting Function

$$51 \quad V = \text{CLIP}(\pm P, \pm Q, \pm R, \pm S)$$

$$V = P \quad \text{if } R \geq S$$

$$V = Q \quad \text{if } R < S$$

where P, Q, R, S need not be distinct.

Switch Function

$$49 \quad V = \text{SWITCH}(\pm P, \pm Q, R)$$

$$V = P \quad \text{if } R = 0$$

$$V = Q \quad \text{if } R \neq 0$$

This function is especially useful for testing several decision rules by means of reruns. Both decision rules are included in the model, and the appropriate one is chosen by the SWITCH function in each run or rerun. In this instance R would be a given constant and would be 0 in one run and different from 0 in the other, for example, 1.

Table Functions

The DYNAMO user will occasionally desire to express a variable having a rather arbitrary relationship to another variable. Frequently this relationship can be expressed most easily by a graph or a table corresponding to the graph. Figure 2-3 shows such a graph. A table corresponding to the graph can be made

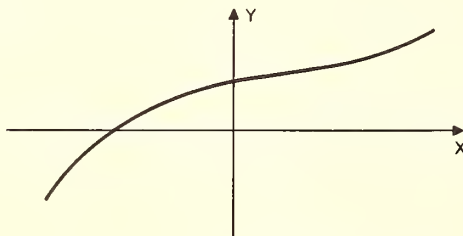


Figure 2-3 Relationship between X and Y.

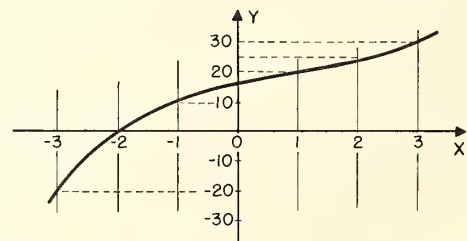


Figure 2-4 Graph segmented to make table.

by dividing the range of the independent variable (X) into equal segments and noting the corresponding values of the dependent variable (Y) as shown in Figure 2-4. The table corresponding to the graph can now be easily prepared by listing side by side the values of X and Y, as in Table 2-2.

Table 2-2 Corresponding Values of X and Y

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

Unfortunately, DYNAMO cannot accept graphic input, but the table of numbers provides no problem. In fact, DYNAMO does not require both columns of numbers. As the left column has equal increments from one value to the next, DYNAMO requires only the first and last values and the size of the increment. The right column is arbitrary and must be provided in complete detail.

The exact punching format of the table look-up is

```
59      V=TABLE(NAME,P,±N1,±N2,N3)
```

V = dependent variable

NAME = name of table

P = independent variable

N1 = lowest value of range of independent variable

N2 = highest value of range of independent variable

N3 = increment in independent variable

If Y in the previous example is an auxiliary, then the equation for Y would be

```
59A     Y.K=TABLE(YTAB,X.K,-3,3,1)
```

where YTAB is the name of the table for Y; YTAB would be given by a constant card (see page 26) as follows:

```
C       YTAB*=-20/0/10/16/20/24/30
```

Since DYNAMO has been given only a series of points along the curve shown in Figure 2-3 rather than the complete curve, it must have some procedure for generating values that it cannot find in the table. The procedure DYNAMO uses is to interpolate linearly between values (draw a straight line between the two closest values and select the answer from the straight line.) This means that the curve DYNAMO is using is the broken line shown in Figure 2-5.

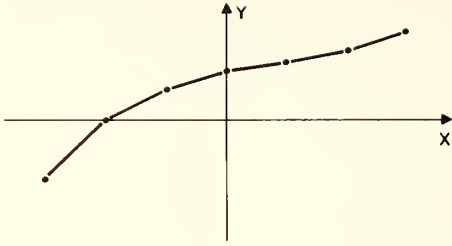


Figure 2-5 Broken curve used by DYNAMO

The range of values of the independent variable during the computer run may not exceed the range provided by the table. Otherwise, DYNAMO will stop the computer run when the range is exceeded and will print the latest value of TIME, the value of the independent variable, and the name of the dependent variable.

Some curves have horizontal asymptotes, and should the range of the independent variable extend quite far along the asymptote, it becomes a chore to punch a table with one value repeated many times. To eliminate

this task, DYNAMO has a second table look-up function that uses the extreme value in the table when the range is exceeded. The exact punching format is

```
58      V=TABHL(NAME,P,+N1,+N2,N3)
```

The explanation of the terms is the same as before. In TABHL, HL signifies that the independent variable may exceed the high and low limits of the range provided for it.

Summing Functions

```
53      V=SUM1(N,P)      V = \sum_{i=1}^N P_i
```

```
55      V=SUM2(N,P,Q)   V = \sum_{i=1}^N P_i Q_i
```

```
57      V=SUM3(N,P,Q,R) V = \sum_{i=1}^N P_i Q_i R_i
```

In the three summation functions, P, Q, R may be either tables or boxcar trains. The names of the tables or boxcars are punched without an asterisk as the arguments of the function. Of course, N must be an integer.†

† An oversight in DYNAMO is the checking of the summing functions used in auxiliary equations for order of computation. The exact condition that is not checked is the summing by an auxiliary equation of a train, one or more of whose boxcars are defined by auxiliary equations. For example, in the following set of equations

```
37B    TRAIN=BOXLIN(5,1)
C      TRAIN*=0/1/1/1/1
53A    SUM.K=SUM1(5,TRAIN.K)
20A    TRAIN*1.K=X.K/Y.K
```

SUM will contain the .J value of TRAIN*1 rather than the .K value.

For example, the entries of the table T1A are to be multiplied by the corresponding entries of the train of boxcars named BBB, and the results are to be summed over the first five entries. The result is to be a rate BBT. The card would have the form

```
55R   BBT.KL=SUM2(5,T1A,BBB.K)
```

Treatment of Round-off in Comparisons with TIME

In the equations whose action is indicated every time that TIME is a certain value (such as boxcar-shifting times, pulse times, sample times, and the like), unless DYNAMO takes certain precautions, round-off errors can cause variations in the time at which action is initiated. To avoid such variations, the time to initiate action is compared with the value of TIME at each computation step, and either if TIME is greater than the action time or if the two agree to within $DT/2$, action is initiated. (The same treatment of TIME governs the output times for printing and plotting.)

Variable Boxcar-Shifting, Pulse, Sample, Step, and Ramp Times

In the cases where the time to initiate an action, say a pulse time, is given as a variable, the following method is used. A record is kept of the last action time for each equation involving some sort of action. As an example, consider a sample equation in which the last sample time will be called Q . Initially the value of Q is set to 0. Then at each computation step the current value of Q is compared with TIME to see if the first sample time has occurred. When this first sample time has been reached, sampling is made, and the current value of Q , say Q_1 , is recorded as the last time a sample was taken. Then at each succeeding computation step, the sum of Q_1 and the current value of Q are compared with TIME to see if the next sample time has been reached. When it has, the sampling is done, the current value of Q , say Q_2 , is added to Q_1 , and the result recorded as the last sample time. The process continues in this fashion so that sampling is done at time Q_1 , Q_1+Q_2 , $Q_1+Q_2+Q_3$,

This method for determining action times may seem somewhat contrived. To explain the motivation, let us consider for a moment not variable action times but action times that are nonintegral multiples of DT . These will illustrate exactly the same problems encountered in the use of variable action times. We shall show why the following method for determining action times (which may seem more natural) is not satisfactory. In this method, the first action is initiated as soon as TIME is equal to or greater than the action time Q , but the current value of TIME (not Q) is stored as the last action time. The next action is initiated as soon as TIME exceeds the sum of the previous action time and Q , and the new value of TIME is stored as the last action time, and so on.

Consider now the case where $DT = 1$ and $Q = 1.5$. Use of the method described directly above would give the action times shown in Table 2-3. In other words, after TIME has reached the value of 6, when we should have expected $6/1.5 = 4$ action times, only 3 have occurred.

If, on the other hand, we consider the method of determining the action times actually used by DYNAMO, we find 4 action times, as given in Table 2-4.

The fact that the correct number of action times occurs is particularly important in functions such as boxcar shifting, where the total number of shifts is especially critical.

Table 2-3 Action Times Using Alternate Method

TIME	ACTION TIMES
0	No
1	No
2	Yes (TIME $\geq 0 + Q$)
3	No
4	Yes (TIME $\geq 2 + Q$)
5	No
6	Yes (TIME $\geq 4 + Q$)

Table 2-4 Action Times Using DYNAMO Method

TIME	ACTION TIMES
0	No
1	No
2	Yes (TIME $\geq Q = 1.5$)
3	Yes (TIME $\geq Q + Q = 3.0$)
4	No
5	Yes (TIME $\geq Q + Q + Q = 4.5$)
6	Yes (TIME $\geq Q + Q + Q + Q = 6.0$)

In practice, an action time is best chosen as an integral multiple of DT and, preferably, considerably greater than DT.

In the above demonstration the action times were selected by comparing the next action time directly with TIME, without consideration of whether the next action time fell within DT/2 of the present time. In practice, the next action time is always compared with TIME + DT/2.

2.7 The Choice of DT

The choice of the size of DT[†] requires a compromise between a large DT that demands less computer time and a small DT that assures numerical accuracy. The accuracy of the model results associated with any particular value of DT is almost impossible to determine except by running the model and comparing these results with those obtained with a much smaller DT. Minor variations in the model should not significantly affect this accuracy.

There is an upper limit on the size of DT that is a consequence of the first-order delays included in the model. DT must be smaller than twice the smallest first-order delay, and experience indicates that half the smallest first-order delay is a good first choice for DT. Since a third-order delay is three cascaded first-order delays with delays one-third of the over-all delay, DT should be chosen

[†] See also Appendix D of Industrial Dynamics.

$$DT \leq \left(\frac{1}{2}\right)\left(\frac{C}{3}\right) \quad \text{or} \quad \leq \frac{C}{6}$$

where C is the magnitude of the shortest third-order delay. It is important to repeat, however, that under certain circumstances acceptable numerical accuracy cannot be attained without resorting to smaller values of DT.

2.8 Estimating the Computer Time Required

The following formula gives a rough estimate of the computer time required to make the basic run of a model on an IBM 709. If no SPEC card is included, consider the LENGTH to be 0; that is, use only the first term. If either PRTPER or PLTPER is 0, ignore the term involving it.

$$\begin{aligned} \text{Time for basic run} = & \frac{\text{number of cards}}{600} + \frac{(\text{number of cards})(\text{LENGTH})}{(200,000)(DT)} \\ & + \frac{(\text{number of quantities being printed})(\text{LENGTH})}{(6000)(\text{PRTPER})} \\ & + \frac{(\text{number of graphs})(\text{LENGTH})}{(300)(\text{PLTPER})} \text{ minutes} \end{aligned}$$

The time required for each rerun is given by the following formula:

$$\begin{aligned} \text{Time for a rerun} = & \frac{(\text{number of cards})(\text{LENGTH})}{(200,000)(DT)} \\ & + \frac{(\text{number of quantities being printed})(\text{LENGTH})}{(6000)(\text{PRTPER})} \\ & + \frac{(\text{number of graphs})(\text{LENGTH})}{(300)(\text{PLTPER})} \text{ minutes} \end{aligned}$$

As in the basic run, if either PRTPER or PLTPER is 0, ignore the term involving it. Of course, the total time required is the sum of the time required for the first run plus the times required for each of the reruns.

The time required on an IBM 7090 will be about 1/5 of the value computed here.

Chapter 3

HOW DYNAMO WORKS

This chapter gives a very brief description of the internal structure of DYNAMO. It has been written in nontechnical language for the person unfamiliar with computer programming.

DYNAMO is divided into six parts or phases:

- The Input Phase reads the cards describing the model, performs most but not all the checks on the model specifications, and generates tables in preparation for later phases.
- The Generation Phase transforms the model specifications (now in tables) into computer instructions.
- The Running Phase computes the values of the model variables, thereby generating data for printing and plotting.
- The Printing Phase tabulates the data in the form requested on PRINT cards.
- The Plotting Phase plots the data in accordance with the PLOT card requests.
- The Rerun Phase modifies constants in preparation for rerunning a model by the running phase.

3.1 Input Phase

As we shall see later, the other phases of DYNAMO depend on tables for their detailed instructions. It is the job of the input phase to read the cards and to create these tables.

The information on the equation cards is stored in three tables. The quantity name on the left side of the equality sign along with the quantity type is stored in the Quantity Name Table. Each quantity name is stored once and only once in this table. If a name appears on the left of the equality sign of two cards, DYNAMO checks to be sure that one equation is the initial condition for the other. If not, DYNAMO prints out that the quantity has been defined twice.

The equation form number tells DYNAMO how many quantity names or numerical values will be on the right of the equality sign. Further, it tells DYNAMO which algebraic symbols, such as +, -,), and (, will separate them. Thus, when one uses Equation Form 7, $V=P+Q$, DYNAMO knows that two quantity names or numerical values will appear on the right of the equality sign, and that the first will lie between the equality sign and the plus sign, and the second will lie between the plus sign and the blank. It is for this reason that the "Exact Punching Format" listed in Appendix B must be followed exactly. It also is the reason why the final space must be provided.

The quantity names on the right of the equality sign, along with the equation form number are stored in Short Tables, one table for each equation. If the DYNAMO user employed a numerical value in place of a quantity name, the

numerical value would be stored in a Numerical Value Table, and its location in this table would be stored in the Short Table.

In the case of constant cards, no Short Table is necessary. The numerical value is stored in the Numerical Value Table. Its location in the Numerical Value Table along with the name is stored in the Quantity Name Table.

The quantity names on PRINT cards are stored in a "PRINT Table" according to the column and row where they are to appear. (The first name to appear in any column is assigned the first row, the second name, the second row, and so forth.)

The quantity names on PLOT cards are stored in a PLOT Table, grouped according to which are to have the same scales. These groups are followed by the upper and lower scale, if the user has provided them, or by "flags" that indicate that DYNAMO is to choose the scales.

Each quantity name that appears on a PRINT or PLOT card is also stored once and only once in a STORE Table. This forms the list of quantities whose values must be stored for later printing and/or plotting.

After DYNAMO has read in all the cards (except the cards for any reruns), DYNAMO assigns to each quantity name in the Quantity Name Table a location where that quantity's current value will be stored. Levels and rates will actually be assigned two locations. Rates will have one location for the new value .KL and another for the previous value .JK. Levels will have one location for the current value .K and another for the difference between the .J and .K values.

The last task of the input phase is going through the Short Tables, which now contain quantity names and locations of numerical values, and replacing the quantity names with their location in storage. Any quantity name that cannot be found in the Quantity Name Table is an undefined quantity; that is, it never appeared on the left of an equation, and consequently, no value is available for it.

3.2 Generation Phase

At this point it is necessary to digress a moment and to discuss briefly how a computer works. Perhaps the two most important parts of a computer are the arithmetic unit and the memory. In accordance with a sequence of instructions, the arithmetic unit takes numbers from the memory, performs arithmetic operations with them, and returns the results to the memory. A key portion of the arithmetic unit is the accumulator. The accumulator is the device in which the additions, subtractions, multiplications, and divisions take place. Before any of these operations, the addend, subtrahend, multiplicand, or dividend must be "loaded" into the accumulator from the memory unless it has been left there from a previous operation. Afterward, the result of each of these operations is left in the accumulator. If for any reason the result is to be saved, contents of the accumulator must be "stored" in some location in memory. Thus, the instructions take the form of loading the accumulator with the contents of some location in memory, adding to, subtracting from, multiplying or dividing the accumulator by the contents of some location in memory, or storing the contents of the accumulator in some location in memory. There are also other types of instructions, but they will not be necessary for this discussion.

Returning to the internal workings of DYNAMO, we find that the generation

phase includes "skeleton instructions" for each equation form. These instructions are skeleton in the sense that the part of the instruction that normally specifies the location of a number in memory specifies only the position in the Short Table, where the location of the quantity is stored.

DYNAMO selects the skeleton instructions according to the equation form number, replaces the position indicator with the proper location for that quantity, and adds these instructions to the sequence that will form the model. As example, let us refer again to Equation Form 7, $V=P+Q$. The skeleton instructions for this equation form are

- Load the value of the first entry in the Short Table into the accumulator.
- Add the value of the second entry in the Short Table to the number in accumulator.
- Store the results in the location for the quantity being defined.

When the generating phase encounters an Equation Form 7, it adds these three instructions to the sequence of instructions that will constitute the model. As it adds each instruction, it replaces "the value of the first entry in the Short Table" with the location it finds as the first entry in the Short Table, and so on.

Before DYNAMO inserts the location for a quantity in an instruction, it checks to determine whether that quantity has been previously calculated. If it is a rate and the code being generated is for an auxiliary (or rate or supplementary) equation, then DYNAMO checks whether the model included an N-equation for the rate. If not, DYNAMO tags the equation for the rate to be repeated among the N equations. An example of DYNAMO-created initial-value equations will be described later.

Another type of quantity that may not have been previously calculated is an auxiliary being used in an auxiliary equation. When DYNAMO finds such a case, it reorders the equations so that no auxiliary is used in an auxiliary equation before it has been calculated. For example, suppose that DYNAMO wishes to generate the sequence of instructions for the auxiliary A.K but finds that A.K depends on another auxiliary, B.K. Furthermore, suppose that DYNAMO checks and finds that B.K has not had its sequence of instructions generated yet. DYNAMO puts A.K in a special list and then attempts to generate the sequence of instructions for B.K. Now let us imagine that B.K depends on the auxiliary C.K, whose sequence of instructions also has not been generated. If this is the case, DYNAMO adds B.K to the list after A.K and tries to generate the code for C.K. Let us consider two cases. In the first case, either C.K does not depend on any auxiliaries, or the auxiliaries on which it depends have already been calculated. Consequently, the sequence of instruction for C.K can be generated. DYNAMO proceeds to do just that and, furthermore, starts up the list and generates the sequence of instructions for B.K and then for A.K.

In the second case, C.K depends on A.K. Since A.K is calculated from B.K and B.K from C.K, DYNAMO is unable to generate the instruction for any of these quantities. It prints "SIMULTANEOUS EQNS AMONG AUX" and lists the quantity names A, B, and C that are involved in the simultaneity.

In this same way DYNAMO also reorders initial conditions and supplementaries.

DYNAMO generates the code for the N equations last, in order to tag the variables that require initial values before any of the N equations have been generated. But during the process of generating them, more variables may be found to require initial values.

The simple model developed in the primer will serve as an example of this whole process. DYNAMO, as it is creating the rate equation for PSR, tags RRR as being a rate that requires an initial value but for which none was provided. Similarly, as it is creating the delay equation for SRR, it tags PSR as a rate requiring an initial value but for which none was provided.

Then, during the generation of the initial-value equations, DYNAMO tries to use the rate equation for RRR:

$$7R \quad RRR.KL = RRI + RCR.K$$

as the initial-value equation for RRR. But since RCR does not have an initial-value, DYNAMO proceeds to use the auxiliary equation

$$45A \quad RCR.K = STEP(STH, 5)$$

as the initial-value equation for RCR. Now DYNAMO can use the rate equation above as an N equation for RRR.

The procedure for PSR is quite similar. Before DYNAMO can use the rate equation for PSR,

$$40R \quad PSR.KL = RRR.JK + (1/DIR)(IDR.K - IAR.K)$$

as an initial-value equation, it must create an initial-value equation for IDR from the auxiliary equation

$$12A \quad IDR.K = (AIR)(RSR.K)$$

Note that RSR is a level with an initial value, and consequently, DYNAMO can generate the equation for IDR immediately. The N equation for IAR also requires that IDR have an initial value. Whichever of PSR or IDR DYNAMO first encounters causes DYNAMO to create the N equation for IRD. When the other is encountered, DYNAMO finds that IDR already has an N equation.

In summary, DYNAMO has created the following four equations:

$$\begin{aligned} 45N \quad RCR &= STEP(STH, 5) \\ 7N \quad RRR &= RRI + RCR \\ 12N \quad IDR &= (AIR)(RSR) \\ 40R \quad PSR &= RRR + (1/DIR)(IDR - IAR) \end{aligned}$$

If the user had written his own initial-value equations, he would probably have written

$$\begin{aligned} 6N \quad RRR &= RRI \\ 6N \quad PSR &= RRR \\ 12N \quad IDR &= (AIR)(RRR) \end{aligned}$$

His equations would have been simpler, but both sets give the same value.

3.3 Running Phase

After all the instructions for a model have been generated, it is ready to be run. The first step is to initialize those quantities that require initial values. Thus, DYNAMO

1. Sets TIME equal to 0
2. Temporarily sets all computed constants, levels, auxiliaries, rates, and supplementaries equal to 0
3. Loads all the boxcar trains with the initial values provided by the special C cards
4. Computes the computed constants, and the initial values of levels and of those auxiliaries and rates for which initial values were provided (that is, computes all N-type equations)

Now the model is initialized and ready to generate data. As the initial values of the levels have just been computed, DYNAMO starts with the auxiliaries or step 4 below. A complete time step consists of the following operations:

1. The differences between levels at time .J and time .K are computed.
2. The differences are added to the levels at time .J to form the levels at time .K.
3. DT is added to TIME.J to form TIME.K.
4. Auxiliaries are computed for time .K.
5. Rates for the interval .KL are computed and stored in the extra location for rates. Thus the rates for the interval .JK are undisturbed and available for the calculation of other rates.
6. TIME is tested to determine whether this is an output time. (TIME=0 is, of course, an output time.) If so, the following three substeps are executed. If not, only substep b is executed, and DYNAMO proceeds on to step 7.
 - a. The supplementaries are computed using the .JK values of the rates.
 - b. The rates just computed for the .KL interval are shifted from their extra location to the normal location for rates.
 - c. The present values of those quantities that are to be printed or plotted are saved on magnetic tape for later processing.

In order that DYNAMO can choose a scaling factor for printing and the upper and lower scale for plotting, DYNAMO must have the maximum and minimum values of all the quantities being saved on magnetic tape. Therefore, as each quantity is located and saved, it is compared with the past maximum and minimum for that quantity. If either the past maximum or minimum is exceeded, that value is replaced with the current value of the quantity.

7. TIME is tested to determine whether the run is completed. If so, DYNAMO proceeds to the printing phase. If not, DYNAMO goes on to the next step.
8. Each boxcar train is tested and shifted if this TIME is appropriate. (All trains are shifted at TIME=0.)

At this point, time .L becomes time .K, time .K becomes time .J, and DYNAMO starts the time-step procedure again with step 1 just mentioned.

3.4 Printing Phase

After DYNAMO has generated all the data for the model and computed the maximum and minimum for each quantity, it is ready to print and plot.

First, DYNAMO goes through the PRINT Table and finds those quantities that DYNAMO is to scale. Then from the maximum absolute value for the quantity, DYNAMO computes a scaling factor and a position for the decimal point so that the maximum number of significant digits will be 5.

Then DYNAMO reads the data that were stored on magnetic tape, locates the decimal point, shifts the data into the proper position for the printed page, and writes the relocated data out on magnetic tape for later printing.

3.5 Plotting Phase

After printing the results, DYNAMO goes on to the plotting phase. If the user has not provided all the scales, DYNAMO first computes the largest of the maxima of the quantities that are to be plotted on the same scale and the smallest of the minima. Then it chooses appropriate scales based on these numbers.

Once all the scales are known, DYNAMO proceeds to plot the results. It locates a character across the plot width by computing the percentage of the distance from the lower scale to the upper scale at which the point should be located. It then rounds to the nearest per cent and locates the character in the corresponding position; that is, if a character should be located 42.6% of the way up a scale, DYNAMO would place it in the 43rd position across the plot width. After all the characters are located, the line of characters is written out on magnetic tape for later printing.

If more than one plot has been requested, DYNAMO simply repeats the above procedure until all the plots are completed.

After the plotting is done, DYNAMO goes on to the rerun phase if there are reruns or proceeds to the next model.

3.6 Rerun Phase

The rerun phase is similar to the input phase in that it is concerned with reading cards. Most of the cards will be constant cards giving new values for the constants. DYNAMO searches the table of quantity names to find where the numerical value of the constant is stored. Then it replaces the old value with the new and stores the old value in the Original Value Table. (If the model is rerun again, the first thing the rerun phase does is restore all the constants with their values from the Original Value Table.) If either a PRINT or PLOT card is included in the rerun, the complete previous print or plot request is wiped out and replaced by the new request.

After DYNAMO has read the last card of this rerun, it returns to the running phase.

This chapter has presented a brief description of how DYNAMO generates the computer instructions for a model, runs the model, and tabulates and plots the results. The necessary steps were: (1) the formation of tables from the model equation and direction cards, (2) the generation of computer instruction from the tables, (3) the running of the computer instructions to generate the data, and finally (4) the conversion of the data to tabulated and plotted results.

Appendix A

SCALING LETTERS

There are several instances when DYNAMO uses a single letter following a number to indicate the scaling DYNAMO has applied to that number. One such instance occurs in the tabulated results when the user has specified the scaling, and the number with that scaling exceeds 5 significant figures. Under these circumstances DYNAMO replaces the fifth digit with such a scaling letter. These letters have the following significance:

<u>Letter</u>	<u>Multiply Printed</u> <u>Results by</u>	or	<u>Consider Value</u> <u>Expressed in</u>
A . . .	10^{-3}	. . .	thousandths
B . . .	10^9	. . .	billions
C . . .	10^{27}	. . .	octillions
D . . .	10^{33}	. . .	decillions
E . . .	10^{-6}	. . .	millionths
F . . .	10^{-9}	. . .	billionths
G . . .	10^{-12}	. . .	trillionths
H . . .	10^{-15}	. . .	quadrillionths
J . . .	10^{-18}	. . .	quintillionths
K . . .	$< 10^{-30}$. . .	(off scale)
L . . .	10^{-21}	. . .	sextillionths
M . . .	10^6	. . .	millions
N . . .	10^{30}	. . .	nonillions
P . . .	10^{24}	. . .	septillions
Q . . .	10^{15}	. . .	quadrillions
R . . .	10^{12}	. . .	trillions
S . . .	10^{21}	. . .	sextillions
T . . .	10^3	. . .	thousands
U . . .	10^{-24}	. . .	septillionths
V . . .	10^{18}	. . .	quintillions
W . . .	10^{-27}	. . .	octillionths
X . . .	1	. . .	units
Y . . .	10^{-30}	. . .	nonillionths
Z . . .	$> 10^{33}$. . .	(off scale)

Appendix B

EQUATION FORMS

Specifications

A quantity name consists of 5 or fewer alphabetic or numeric characters, the first of which is alphabetic.

A numerical value consists of 8 or fewer decimal digits with or without a decimal point. This number may be multiplied by some power of 10 by immediately following it with the letter E and the (positive or negative) power of 10. In the equation forms

V	=	quantity name, on left of equality sign
N	}	= numerical values
N1		
N2		
P-U	}	= quantity name or numerical value
W-Z		
C	}	= given or computed constant or numerical value
C1		
C2		

Table B-1 Subscript Table

Quantity Type on Left of Equation		Subscript on Left	Subscripts on Quantities on Right if Quantity is					
			L	A	R	S	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
S	Supplementary	K	K	K	JK	K	none	none
C	Constant	none	—	—	—	—	—	—
N	{ iNitial value or computed constant	none	—	—	—	—	none	none

A ± in an equation form indicates that either sign is acceptable. After any of = (/ or , the + sign can be dropped.

Letters Following Equation Form Number

A or R	may be used to define only auxiliaries or rates.
B	may be used to define only boxcar trains. Also, no other equation types may be used to define boxcar trains.
L	may be used to define only levels. Furthermore, levels may be defined only by so-marked equation forms.
N	may be used to define only initial values.
R	may be used to define only rates.
no letter	may be used to define auxiliaries, rates, computed constants, initial values, or supplementaries; that is, any quantity type except level or boxcar train.

Table B-1 Equation Forms

<u>Form Number</u>	<u>General Form</u>	<u>Exact Punching Format</u>
	<u>Level Equations</u>	
1L	$V = V + (DT)(P + Q)$	$V.K=V.J+(DT)(\pm P\pm Q)$
52L	$V = V + (DT)(P + Q + R + S)$	$V.K=V.J+(DT)(\pm P\pm Q\pm R\pm S)$
2L	$V = V + (DT)(P + Q + R + S + T + U)$	$V.K=V.J+(DT)(\pm P\pm Q\pm R\pm S\pm T\pm U)$
3L	$V = V + (DT)\left(\frac{P+Q}{Y}\right)$	$V.K=V.J+(DT)(1/\pm Y)(\pm P\pm Q)$
4L	$V = V + (DT)\left(\frac{P+Q+R+S+T+U}{Y}\right)$	$V.K=V.J+(DT)(1/\pm Y)(\pm P\pm Q\pm R\pm S\pm T\pm U)$
5L	$V = V + (DT)\left(\frac{P+Q}{Y} + \frac{R+S}{Z}\right)$	$V.K=V.J+(DT)((1/\pm Y)(\pm P\pm Q)+ (1/\pm Z)(\pm R\pm S))$
<u>Sums, Products, and Products with Sums</u>		
6	$V = P$	$V=\pm P$
7	$V = P + Q$	$V=\pm P\pm Q$
8	$V = P + Q + R$	$V=\pm P\pm Q\pm R$
9	$V = P + Q + R + S$	$V=\pm P\pm Q\pm R\pm S$
10	$V = P + Q + R + S + T + U$	$V=\pm P\pm Q\pm R\pm S\pm T\pm U$
11	$V = P + Q + R + S + T + U + W + X$	$V=\pm P\pm Q\pm R\pm S\pm T\pm U\pm W\pm X$
12	$V = (P)(Q)$	$V=(\pm P)(\pm Q)$
13	$V = (P)(Q)(R)$	$V=(\pm P)(\pm Q)(\pm R)$
14	$V = P + (Q)(R)$	$V=\pm P+(\pm Q)(\pm R)$
15	$V = (P)(Q) + (R)(S)$	$V=(\pm P)(\pm Q)+(\pm R)(\pm S)$
16	$V = (P)(Q) + (R)(S) + (T)(U) + (W)(X)$	$V=(\pm P)(\pm Q)+(\pm R)(\pm S)+(\pm T)(\pm U)+(\pm W)(\pm X)$
17	$V = (P)(Q)(R) + (S)(T)(U) + (W)(X)(Y)$	$V=(\pm P)(\pm Q)(\pm R)+(\pm S)(\pm T)(\pm U)+(\pm W)(\pm X)(\pm Y)$
18	$V = (P)(Q + R)$	$V=(\pm P)(\pm Q\pm R)$
19	$V = (P)(Q + R + S + T)$	$V=(\pm P)(\pm Q\pm R\pm S\pm T)$

Ratios

$$V = \frac{P}{Q}$$

$$V = \frac{(P)(Q)}{R}$$

$$V = \frac{P}{(Q)(R)}$$

$$V = \frac{(P)(Q)(R)}{(S)(T)(U)}$$

$$V = \frac{P+Q}{Y}$$

$$V = \frac{P+Q+R+S+T+U}{Y}$$

$$V = \frac{(P)(Q)+(R)(S)}{Y}$$

$$V = \frac{P+Q}{Y} + \frac{R+S}{Z}$$

$$V = \frac{P}{Q} + R$$

$$V = P + \frac{R+S}{Q}$$

$$V = P + \frac{R+S+T+U+W+X}{Q}$$

$$V = \frac{P}{Q+R}$$

$$V = \frac{(P)(Q)}{R+S}$$

$$V = \frac{P+Q+R}{S+T+U}$$

20 $V = \pm P / \pm Q$

44 $V = (\pm P)(\pm Q) / \pm R$

42 $V = \pm P / ((\pm Q)(\pm R))$

46 $V = (\pm P)(\pm Q)(\pm R) / ((\pm S)(\pm T)(\pm U))$

21 $V = (1/\pm Y)(\pm P \pm Q)$

24 $V = (1/\pm Y)(\pm P \pm Q \pm R \pm S \pm T \pm U)$

22 $V = (1/\pm Y)((\pm P)(\pm Q) + (\pm R)(\pm S))$

23 $V = (1/\pm Y)(\pm P \pm Q) + (1/\pm Z)(\pm R \pm S)$

27 $V = (\pm P / \pm Q) \pm R$

40 $V = \pm P + (1/\pm Q)(\pm R \pm S)$

25 $V = \pm P + (1/\pm Q)(\pm R \pm S \pm T \pm U \pm W \pm X)$

48 $V = \pm P / (\pm Q \pm R)$

50 $V = (\pm P)(\pm Q) / (\pm R \pm S)$

26 $V = (\pm P \pm Q \pm R) / (\pm S \pm T \pm U)$

<u>General Form</u>	<u>Form Number</u>	<u>Exact Punching Format</u>
<u>Special Functions</u>		
$V = Pe^Q$ (exponential) (Q must be < 88)	28	V=(±P)EXP(±Q)
$V = P \ln Q$ (natural logarithm) (±Q must be > 0)	29	V=(±P)LOGN(±Q)
$V = P\sqrt{Q}$ (square root) (±Q must be ≥ 0)	30	V=(±P)SQRT(±Q)
$V = P \sin\left(\frac{2\pi Q}{R}\right)$ (2πQ/R must be < 68.7 billion)	31	V=(±P)SIN((2PI)(±Q)/±R)
$V = P \cos\left(\frac{2\pi Q}{R}\right)$ (2πQ/R must be < 68.7 billion)	32	V=(±P)COS((2PI)(±Q)/±R)
Random numbers uniformly distributed between $-\frac{P}{2}$ and $+\frac{P}{2}$	33	V=(P)NOISE
P times random numbers normally distributed with mean ±Q and standard deviation R	34	V=(±P)NORMRN(±Q,R)
Third-order delay: V is the output rate of third-order delay of length C, with input P.	39R	V=DELAY3(±P,C)
V is a train of pulses of height P and width DT. The first pulse appears at time Q, and succeeding pulses, at intervals R.	41(A or R)	V=PULSE(±P,Q,R)
$V = \begin{cases} \text{initial value} & \text{if TIME} < Q \\ P & \text{if TIME} \geq Q \end{cases}$	45(A or R)	V=STEP(±P,Q)
$V = \begin{cases} \text{initial value} & \text{if TIME} \leq Q \\ \text{initial value} + \sum_{Q+DT} P(DT) & \text{if TIME} > Q \end{cases}$	47(A or R)	V=RAMP(±P,Q)

V is set equal to P at sampling intervals Q and is not changed in between.

$$V = \text{MAX}(P, Q)$$

$$V = \text{MIN}(P, Q)$$

$$V = \begin{cases} P & \text{if } R \geq S \\ Q & \text{if } R < S \end{cases}$$

$$V = \begin{cases} P & \text{if } R = 0 \\ Q & \text{if } R \neq 0 \end{cases}$$

V is function of P given by the table, NAME.

V is function of P given by the table, NAME. If range of P exceeds N1 and N2, extreme value of table is used.

N1 = first value of P

N2 = last value of P

N3 = interval between successive values of P

Boxcars and Related Functions

V is a train of N boxcars shifted cyclically at shifting intervals P.

V is a train of N boxcars shifted linearly at shifting intervals P.

V is to be loaded initially with the product of P and Q.

$$V = \sum_{i=1}^N P_i$$

$$V = \sum_{i=1}^N P_i Q_i$$

$$V = \sum_{i=1}^N P_i Q_i R_i$$

$$V = \text{SAMPLE}(\pm P, Q)$$

$$V = \text{MAX}(\pm P, \pm Q)$$

$$V = \text{MIN}(\pm P, \pm Q)$$

$$V = \text{CLIP}(\pm P, \pm Q, \pm R, \pm S)$$

$$V = \text{SWITCH}(\pm P, \pm Q, R)$$

$$V = \text{TABLE}(\text{NAME}, P, \pm N1, \pm N2, N3)$$

$$V = \text{TABHL}(\text{NAME}, P, \pm N1, \pm N2, N3)$$

43(A or R)

56

54

51

49

59

58

$$V = \text{BOXCYC}(N, P)$$

35B

$$V = \text{BOXLIN}(N, P)$$

37B

$$V = \text{BOXLOAD}(\pm P, \pm Q)$$

36N

$$V = \text{SUM1}(N, P)$$

53

$$V = \text{SUM2}(N, P, Q)$$

55

$$V = \text{SUM3}(N, P, Q, R)$$

57

INDEX TO CHAPTER 2

- Action-time, variable, 41-42
- Auxiliary, definition of, 20-21
- Boxcar, 32-34
 - subscripts on, 23
- Boxcar Train, definition of, 21
 - initial value for, 27
- BOXLOAD function, 33-34
- Card, comments on, 26
 - constant, 26-27
 - continuation, 25-26
 - direction, 27-32
 - equation, 20-27
 - identification, 27-28
 - NOISE, 31-32
 - NOTE, 31
 - PLOT, 30-31
 - PRINT, 29-30
 - RUN, 28
 - SPEC, 28-29
- Card punching, 25-26
- Cards, order of, 32
- Clip function, 38
- Comments on cards, 26
- Computer time required, 43
- Constant, computed, definition of, 21
 - given, definition of, 21
- Constant card, 26-27
- Continuation card, 25-26
- Cosine function, 35
- Delay function, third-order, 36
- Direction cards, 27-32
 - See also the particular type card
- DT, 24
 - choice of, 42-43
- Equation card, 20-27
- Equation forms, 21-23
- Exponential function, 34
- Final space, rule of the, 25
- Identification card, 27-28
- Initial values, for auxiliary, 25
 - for boxcar train, 27, 33-34
 - for level, 24
 - for rate, 25
 - for supplementary, 24
 - for third-order delay, 25, 36
- Level, definition of, 20
- Logarithmic function, 34
- Maximum function, 38
- Minimum function, 38
- NOISE card, 31-32
- NOTE card, 31
- Numerical values, 26
- Order of cards, 32
- PLOT card, 30-31
 - in reruns, 32
- PRINT card, 29-30
 - in reruns, 32
- Pulse function, 36-37
- Quantities, permissible defining, 23-24
- Quantity name, 21
- Quantity type, 20-21
- Ramp function, 37
- Random numbers, 35-36
- Rate, definition of, 21
- Reruns, 32
- Rule of the final space, 25
- RUN card, 28
- Sample function, 37
- Scaling in printed results, 30
- Sine function, 35
- SPEC card, 28-29
- Square-root function, 35
- Step function, 37
- Subscripts, 23-24
- Summing functions, 40-41
- Supplementary, definition of, 21
- Switch function, 38
- TABLE functions, 38-40
- Tables, 26
- Third-order delay function, 36
- TIME, 24
 - comparisons with, 41
 - computer, required, 43
- Type, quantity, 20-21
 - variable, 20
- Variable action time, 41-42
- Variable type, 20





651.26.P978D2.C1R



651.26
P978d2

DYNAMO user's manual. main
651.26P978d2



3 1262 03314 4519

