Criminal Justice System

Training Model

FINAL REPORT

MAY 31, 1974

WILLARD R. FEY

HARRISON M. WADSWORTH

DONOVAN B. YOUNG

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

GEORGIA INSTITUTE OF TECHNOLOGY

ATLANTA, GEORGIA 30332

LAW ENFORCEMENT ASSISTANCE ADMINISTRATION

GRANT NO. 73-TN-04-0001 (S-1)

OMB APPROVAL NO. 43-R0525
EXPIRATION DATE 6-30-74

U. S. DEPARTMENT OF JUSTICE LAW ENFORCEMENT ASSISTANCE ADMINISTRATION	DISCRETIONARY GRANT PROGRESS REPORT		
GRANTEE State Crime Commission 1430 W. Peachtree St., Suite 306 Atlanta, GA 30309	LEAA GRANT NO. 73-TN-04-0001(S-1)	DATE OF REPORT 5/31/1974	REPORT NO.
IMPLEMENTING SUBGRANTEE School of Industrial and Systems Engineering Georgia Institute of Technology Atlanta, GA 30332	TYPE OF REPORT ☐ REGULAR QUARTERLY ☐ SPECIAL REQUEST ☒ FINAL REPORT		
SHORT TITLE OF PROJECT Criminal Justice System Training Model	GRANT AMOUNT $42,000.00		
REPORT IS SUBMITTED FOR THE PERIOD April 1, 1973	THROUGH March 31, 1974		
SIGNATURE OF PROJECT DIRECTOR	TYPED NAME & TITLE OF PROJECT DIRECTOR Willard R. Fey Associate Professor, School of ISyE		

COMMENCE REPORT HERE (Add continuation pages as required.)

RECEIVED BY GRANTEE STATE PLANNING AGENCY (Official)	DATE

Table of Contents

I. CRIMINAL JUSTICE SYSTEM TRAINING MODEL REPORT

A. Project Background

This project is a continuation of a project started during the summer of 1972 and funded by the Law Enforcement Assistance Administration through the Georgia State Crime Commission under Grant Number 72-TN-04-0003. The purpose of this grant, as well as the previous grants, was to develop an interactive computer model of the criminal justice system that will be suitable for use in training sessions for criminal justice planners.

The initial grant was to determine the feasibility of such a model. During that initial three-month period a preliminary model was developed and programmed for the Georgia Tech owned Univac 1108 computer. This model was then demonstrated to several representative groups. These demonstrations took place between August, 1972 and February, 1973. The purpose of the demonstrations was to show the feasibility of this approach to the training of criminal justice planners. It was the opinion of persons viewing the demonstrations that this approach was feasible and further funding was provided by LEAA to complete the training model. This is the report which concludes that development.

The training model is based on Atlanta and covers Part I crimes. However, for training purposes, the model could be thought of as a representation of any large city. The important thing to someone interested in learning how to make decisions regarding the criminal justice system is to observe relative changes in the system when various possible changes in control variables are made. However, if the trainee really wants to change the model to represent some other city or area, such as a state or region, instructions for making such changes are included in this report.

B. Activities

1. Data Gathering

Data gathered during the initial feasibility study (Summer, 1972) was of a preliminary nature and only that data that could be readily obtained by interviews, study of published reports, etc. was obtained. Much of the data needed to develop a realistic model was not obtainable by these methods. When such data was not available, the researchers were forced to make estimates of the appropriate numbers. One of the tasks of the current study has been to go back and obtain the correct data where this occurred. As a result of this step the model quite adequately represents the condition in Atlanta in 1960, the initial period of the simulation studies.

Data was also required for the years between 1960 and the present. This has been obtained and used to improve the relationships expressed by the model. It is believed that all necessary data has now been obtained to represent the Atlanta Criminal Justice System as far as it pertains to Part I crime during the period studied.

2. Model Building

The initial feasibility study developed the framework for the model. However because of the brief time available many aspects of the model were not activated at that time. In other words, if the dynamic nature of some relationship was not understood it was not made a part of the model. One of the principal tasks of this phase of the training model development was to go back and learn enough about each of these relationships to activate them in the model. In this sense then, the basic model of the system has become considerably larger and more realistic than was the previous model.

These additional relationships were added very carefully and were done only after the appropriate data was collected and sample simulations were made. The resulting simulations were described to criminal justice experts who thus helped in their final evaluation.

The final model that has been developed will be described in detail in several places in this report. First in Section II in which the structure and relationships are described by means of words and flow diagrams and by equations. This is the mathematical and philosophical basis for the model. Section III describes the model in the form of FORTRAN statements. This is the computer program which enables the trainee to use the model. Section III also contains a model builder's guide which will facilitate any later changes in the model to more closely represent any particular system. Finally Section IV consists of a trainee's manual which describes the model in qualitative terms and contains complete descriptions of the procedures recommended for using it for training purposes.

3. Programming

As changes or additions to the model equations were proposed they were translated into computer language and the original program was correspondingly modified. A number of improvements were also made in the interactive capabilities. These were made to attempt to make the use of the program more appealing to the trainee by more nearly enabling it to answer any question he might wish to ask. The computer program is discussed in detail in Section III.

Many of these changes were made following questions that had been asked during the several demonstrations discussed in the first paragraph of this report. Others were suggested by members of the advisory committee which is discussed in the next paragraph. Still others were proposed by members of the research team in weekly meetings held throughout the life of the project. While it is felt that the current version contains most of the interactive conveniences that may be put into the program, additional refinements may come up in later demonstrations. If this happens the program will be modified if possible to accommodate these improvements.

4. Advisory Committee

An ad hoc advisory committee was appointed by Mr. George Murphy, Director of the Atlanta Regional Office of LEAA. This committee consisted of the following members:

a. Dr. Gordon Waldo
   Southeastern Correctional and Criminology Research Center
   Florida State University
   Tallahassee, Fla.

b. Mr. Chet Dettlinger
   Chief, Police Planning
   Kentucky Crime Commission
   Frankfort, Kentucky
   (In September, 1973 Mr. Dettlinger changed positions to become Assistant to the Police Chief, City of Atlanta, Atlanta, Ga.)

c. Judge Reed Merritt
   Gwinnett County Superior Court
   Lawrenceville, Georgia

d. Ms. Cheryl Purvis
   Office of State Crime Commission
   1430 W. Peachtree Street
   Atlanta, Ga.

The above committee represented the respective areas of corrections, police, courts, and overall systems in the criminal justice field. Its purpose was to advise the research team on matters pertaining to model development and implementation. They met together in Atlanta on many occasions during the life of the project and contributed their expertise to the project.

Some of the specific things they did were to correct some of the definitions of terms used by the research team, propose other variables and relationships that should be considered, prod the research team to more adequately define variables used in the model, etc.

It is uniformly agreed by all persons associated with the project that the assistance and advice given by the advisory committee was invaluable and, in fact, essential to successful culmination of the project. These expert practitioners in the criminal justice field added a great many years of practical experience to the project that could not have been obtained in any other reasonable way.

5. Presentation to CJS Groups

The model has not been demonstrated to members of any criminal justice groups since the start of the present improvement phase. However the use of the Trainees Manual, included in this report, will greatly facilitate such presentations in the future. In addition to the Trainees Manual, suitable training aids, such as slides, charts, etc. will be prepared for future presentations and training programs. The next task facing the research team is the development of such training aids. This will be accomplished in time to conduct training programs specified by LEAA personnel.

6. LEAA Personnel Trained to Administer Training

No personnel, other than members of the research team, are currently prepared to administer training sessions using this model. However, others can be instructed very easily if desired by LEAA. It is hoped that in the near future such persons will be designated and this instruction may be given.

II. THE CRIMINAL JUSTICE SYSTEM MODEL

The criminal justice system model is a group of mathematical expressions (equations) and numbers. The equations represent the forces and influences between conditions and decisions in a selected criminal justice system. The numbers correspond to the values of unchanging system characteristics (parameters) and the values of certain necessary quantities at a time chosen as the beginning of the study. The equations and numbers, taken together and iteratively solved on a large computer, recreate the time histories of variables such as crime rate, number of police officers, corrections budget, etc. in the criminal justice system starting at a pre-selected time. For this model the starting time is 1960 and the subject system is the City of Atlanta.

The model is designed to be used as a training aid. It is felt that a person's understanding of how the parts of a criminal justice system are related and what might be the consequences of changing decisions or budgets will be improved by his involvement with this model. This means examining the flow diagrams and descriptions contained in the attached training manual and observing and interpreting the model's time histories under different conditions. Since the model is for training, rather than decision making, the scope of the project is somewhat limited and the predicted time histories generated by the model should not be used for planning or policy making purposes.

A. Model Structure

This CJS model has six major parts or sectors designated criminal flows, police, courts, corrections, financial, and community. Within each sector there are several variables that are mutually dependent in ways that create the changes in their values through time. In addition, certain conditions in each sector will influence activities in other sectors as, for example, when efforts by prosecutors in the courts sector create a flow of criminals from court to prison in the criminal flow sector. The major sectors and their connections are shown in Figure 1. A somewhat more detailed view particularly of the flows in the criminal flow sector is shown in Figure 2. Finally, the chart of all the variables is given in Figure 3. These three diagrams will be used extensively in the following detailed description of the relationships.

Any person in a criminal justice jurisdiction can be associated with one and only one of six states or conditions. These are non-criminal, free active criminal, person with an unresolved arrest charge ("open case"), in prison, on probation, or on parole. Through time, people move from one condition to another. The "rates" or flows in Figure 2 represent the number of people per year who flow from the state at the tail of the arrow to the state at the arrow head. Since the existence of crime creates the need for the criminal justice system, and since people in the latter five states commit the crimes with different activity for each state, the flows between these states determine whether crime incidence increases, decreases, or remains the same. How these criminal flows and states create pressures on the criminal justice system and how the CJS adjusts its resources and responds to influence the criminal flows is the key to an understanding of why the important variables change the way they do. The description begins with the way the sectors of the CJS adjust their resources.

In the model the three parts of the CJS (police, courts, and corrections) have similar representations, so they will be described together.

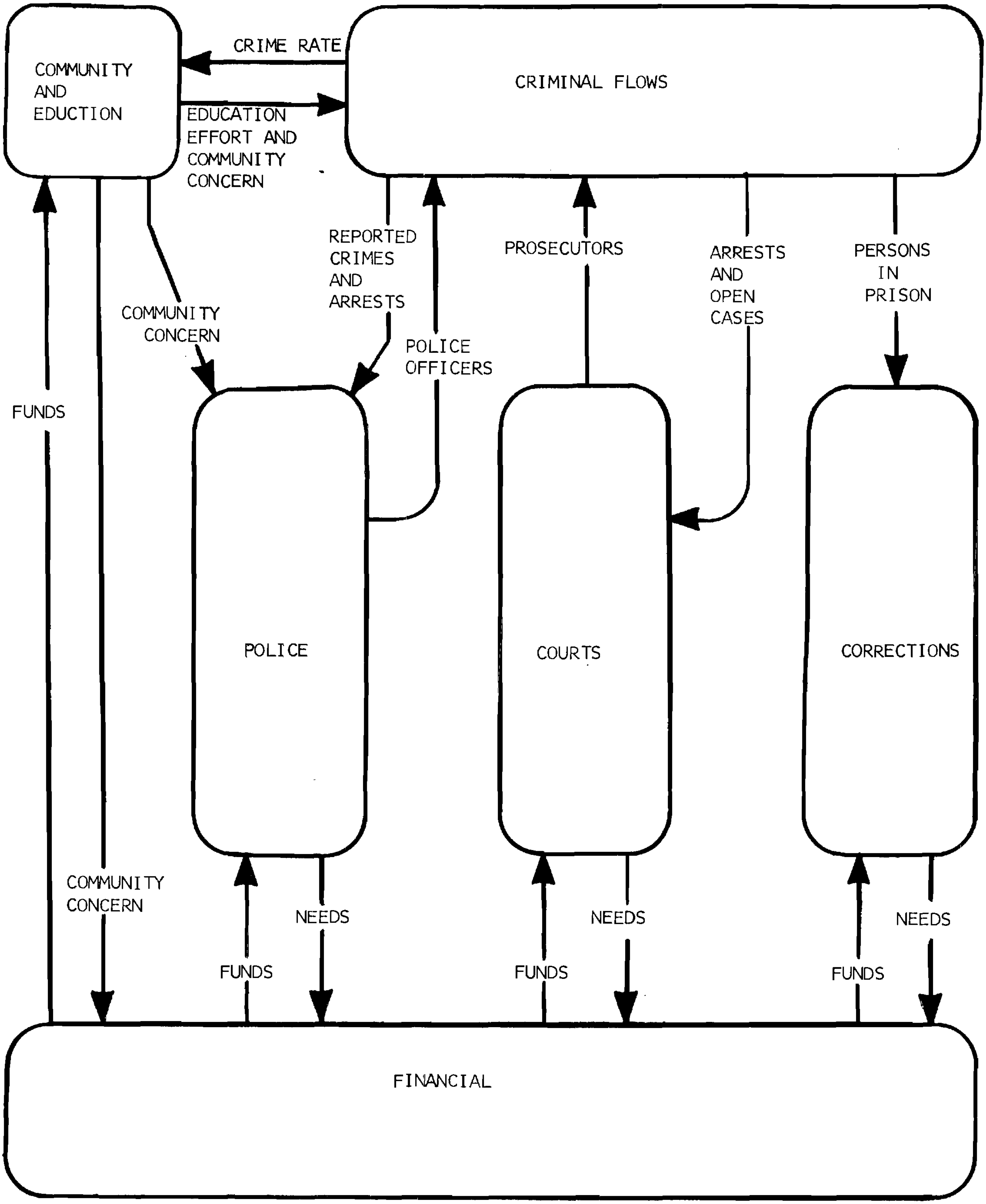


FIGURE 1. CRIMINAL JUSTICE MODEL SECTORS AND INTERACTIONS.

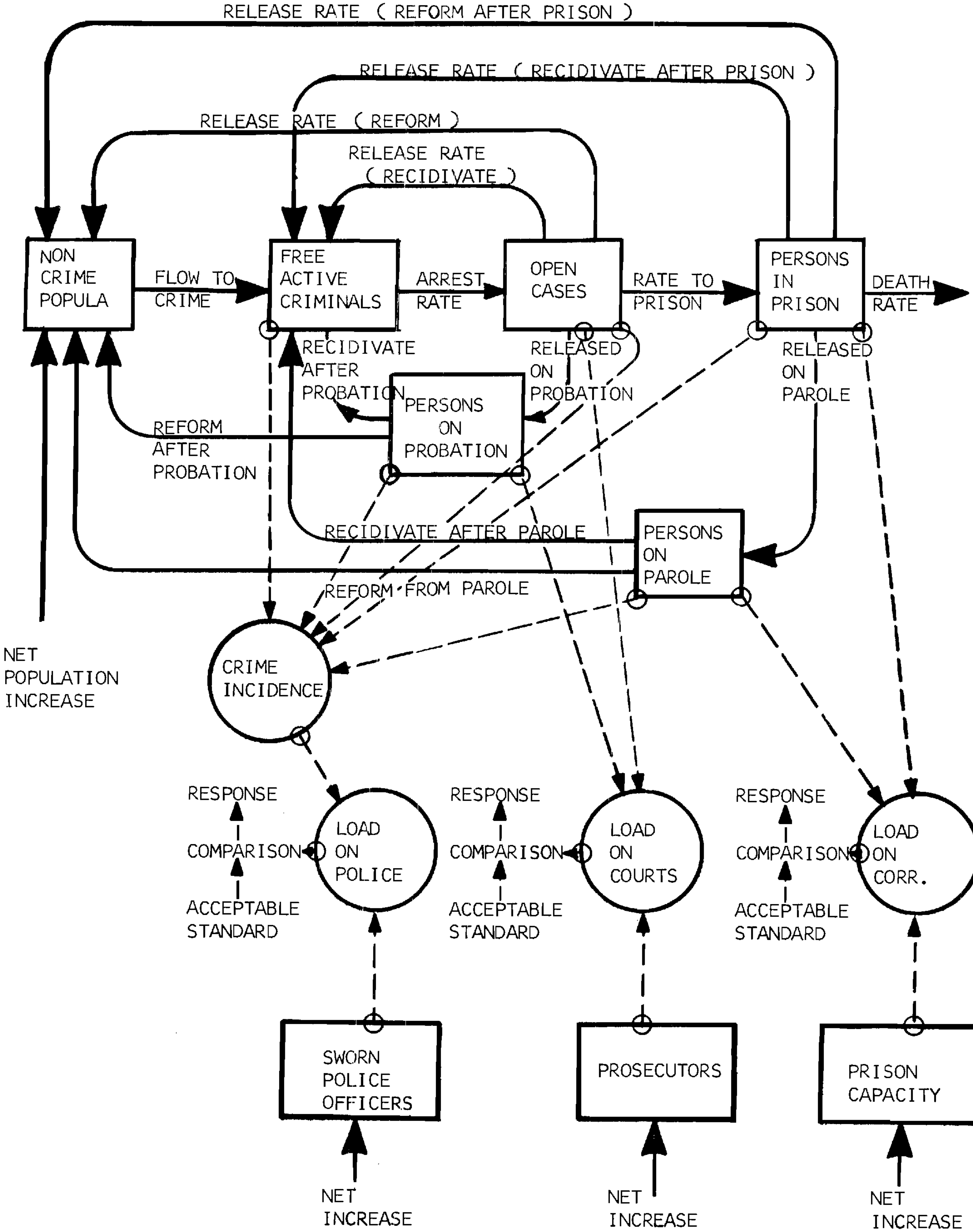
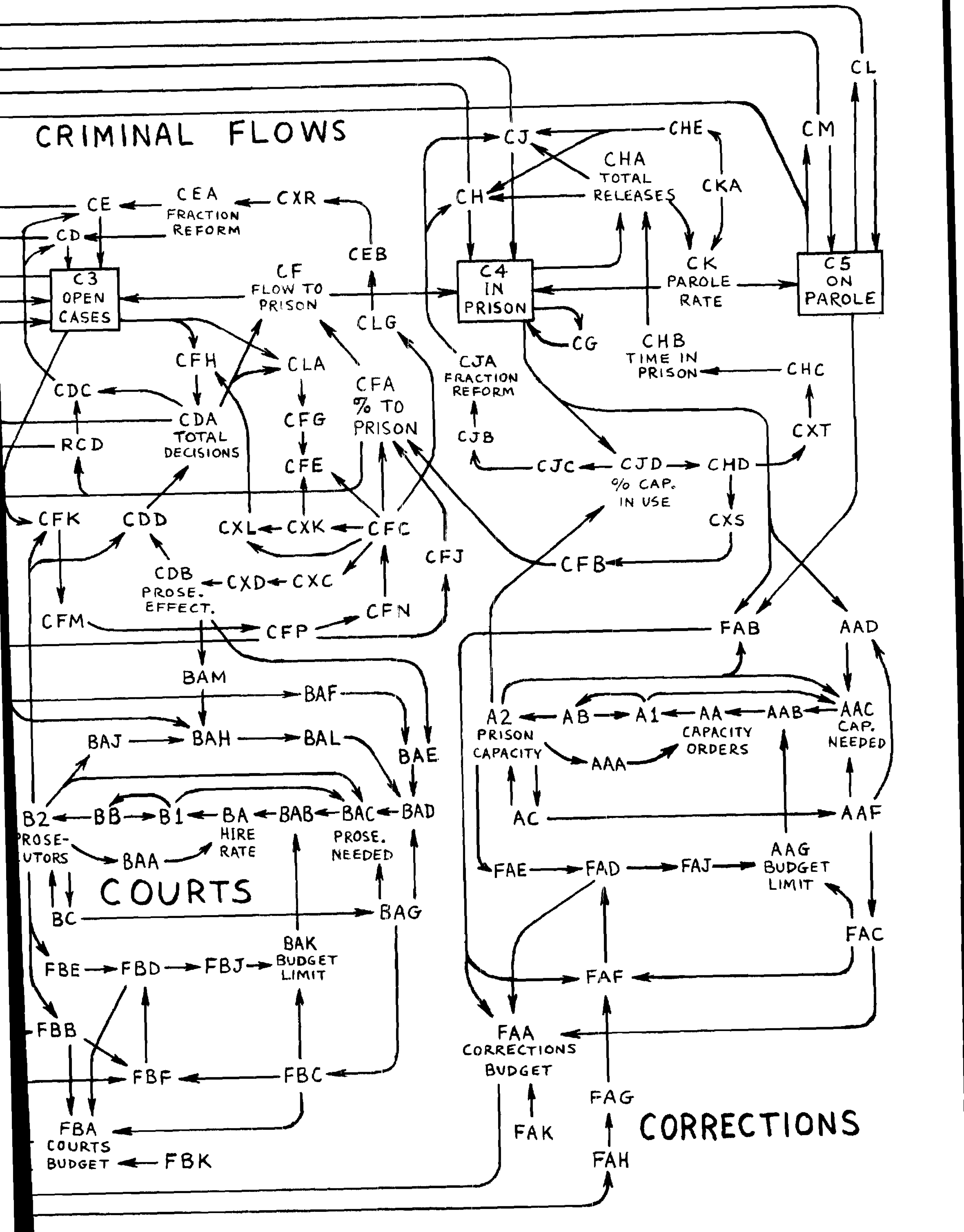


FIGURE 2. FLOWS REPRESENTED IN THE CRIMINAL FLOWS SECTION OF THE CRIMINAL JUSTICE MODEL.

Each sector has one basic resource: sworn officers for the police, prosecutors for the courts, and prison capacity for corrections. These are the factors that directly relate to the criminals and their activities. For financial accounting purposes, each resource stands for the total of all existing resources in its sector including employees, equipment, and buildings. Each resource has a natural attrition rate (the rate at which resources disappear through voluntary leaving, retirement, wear out, obsolescence, etc.), an acquisition decision that may be positive for acquisition or negative for divestment, and a delay between the decision and the actual acquisition or divestment actions. These are shown in Figure 3.

In each case the acquisition decision is based on three things: replacement of resources lost through attrition, an attempt to adjust the resources to the number needed to respond properly to the pressures created in the criminal flow sector, and the availability of money to support the existing and needed resources. The acquisition decision is the smaller of the rate needed based on pressures and the rate provided for by the budget. The acquisition budgets come from the financial sector to be discussed next. The pressures on the three sectors are these. Prisoners form the primary pressure on corrections; the case load exerts the pressure on the courts; and crimes reported create police pressures.

The financial sector generates budgets for the police, courts, corrections, and education. Educational expenditures are a fraction of the budgets for the other three sectors. The fraction is directly dependent upon community concern. The police, courts, and corrections budgets are all similar in the sense that each involves funds for operating expenditures, replacement of resources, acquisition of new resources, and funds from outside



Criminal Justice System Training Model.

sources. The budget for new resources is the smaller of the rate at which the sector can absorb new resources and the percentage increase the community will tolerate based on its concern about crime.

The community sector is designed to evaluate the crime rate through time and determine how concerned the community is about crime. This concern then determines how much financial support the people will permit for the criminal justice functions and extent to which the people will cooperate with police and law enforcement officials.

The remaining aspects of the model are the factors that control the flows of people in the criminal flow sector. These flows, as shown in Figure 2, include the "Net Population Increase" that builds up the "Non Crime Population" and the "Flow to Crime" that shifts people to the "Free Active Criminals" state. Some of these criminals become associated with the criminal justice system by being arrested (Arrest Rate) and becoming an "Open Case" in the judicial system. Open cases must be resolved in one of three ways: a prison sentence (Rate to Prison), "Release on Probation," or free release which results in a return to crime (Release Rate-Recidivate) or reform and return to "Non Crime Population" (Release Rate-Reform). "Persons in Prison" eventually leave prison by dying (Death Rate), being "Released on Parole," or finishing their sentence and being released to return to crime (Release Rate-Recidivate After Prison) or return to non-crime. The "Persons on Parole" and "Persons on Probation" also return either to crime or non-crime when their trial period is over.

The "Net Population Increase" at the far left in Figure 2 (CA in Figure 3) is based on the average empirically observed percentage increase per year in population in Atlanta.

"Flow to Crime" (CB in Figure 3) is the net number of people per year who first become criminals. It is formulated as a normal constant percentage of the "Non Crime Population" with deviations from normal being produced by changes in six factors. These are 1) the fraction of the total population that is criminal (called the peer pressure influence, CBB), 2) the financial reward of crime, CBD, 3) police visibility and arrest effectiveness, CBF, 4) probability of punishment as measured by the fraction of those arrested who are sent to prison, CBH, 5) influence of effort to educate the population to the problems of crime, CBJ, and 6) an empirical function of time, CBS, that represents the rate at which youngsters reach the crime susceptible teen age years.

The "Arrest Rate" (CC in Figure 3) is the rate in people per year at which people are arrested by the police. It is the number of police times the usual arrests per officer per year. This is modified by influences from five factors. These are 1) the budget per officer influence, CCX, that assumes that better equipped, supported, and paid officers arrest more criminals, 2) the crime load per officer influence, CCJ, assumes that over- or under-worked officers lose efficiency, 3) trial delay influence, CCP, indicates that when the trial delay gets to be too long there is not much point in arresting people, 4) influence of community concern, CCR, and 5) the influence of danger to police, CCM, that is based on the aggressiveness of criminals.

The four flows leaving the "Open Cases" box are "Rate to Prison," CF, "Released on Probation," CN, "Release Rate-Reform," CE, and "Release Rate-Recidivate," CD. These are formulated on the basis of a total disposition rate, CDA, being divided into four parts by determining the fraction sent to prison, CFA, the fraction of those remaining released on parole, CNAA, and the fraction of the free releases that reform, CEA.

Total case disposition rate is based on the number of prosecutors and their average effectiveness which varies with the backlog of "Open Cases." The prison sentence fraction is formulated to vary a small amount in response to changes in the amount of plea bargaining and prison crowding. The fraction of those released who reform also varies somewhat with plea bargaining.

A small fraction of "Persons in Prison" leave each year by dying, CG. Others leave on parole, CK, or after finishing their sentence, CH and CJ. The total release rate, CHA, is based on the average effective sentence time. The total rate is split into three parts based on the fraction paroled, CKAA, and the fraction who reform, CJA. This propensity to reform is influenced by average prison crowding.

At the end of the parole time, some parolees reform, CL, and the rest return to crime, CM. The reform proportion is constant, CLAA. At the end of the probation time, some probationers reform, CQ, and the rest return to crime, CP. This reform proportion, CQAA, is also constant.

This is an overview of the model that describes some of the influences in the criminal justice system. The detailed descriptions and the equations themselves follow at the end of the chapter. As shown in Figure 3 the model is self contained. The value of each model variable at each point in time of the simulation is derived from other model variables. Outside data time histories are not used to determine the model time histories. As shown in Figure 1 there is a great deal of mutual influence between the parts of the criminal justice system. It is this interaction that creates the patterns of change in the variables.

B. Model Performance

The model is intended to simulate the operation through time of a reasonably realistic criminal justice system for training purposes. Therefore, the time histories that the model creates must be reasonable representations of possible time histories. The time histories actually produced by the model are not and were not intended to be perfect reproductions of the subject system - Atlanta, 1960-1970 - though in the case of most variables for which we have reliable data the approximation is rather close.

The dominant characteristic of crime and criminal justice systems in the 1960's was expansion. Atlanta, like most large cities, experienced a sizable continuing increase in the number of reported crimes, the physical and financial size of the police, courts and corrections activities, and the time to try a case. The percentage increase in all of these was substantially greater than the percentage increase in population. The time histories of five model variables - crime rate, reported crimes per police officer, total criminal justice budget, open cases per prosecutor and reported crimes per police officer -- are shown in Figure 4. This has the model conditions that correspond most closely to Atlanta (1960-1970).

This simulation shows an initial increase in all five factors and then in the 1970's the rates of growth are reduced. The initial increases in the 1960's are due to the rapid flow of people from Non Crime to Active Criminal status. This flow was caused by the unavoidable social frustrations in the city; the increasing financial reward of crime; deteriorating justice in the form of increased plea bargaining, lengthening trial delays, declining fraction sentenced to prison; and the large number of youths reaching the middle teen ages when crime is an enticing experience. A

RUN	LØWER	UPPER	SYM	NAME	DESCRIPTIØN
1	.000	.500+08	C	F1	TØTAL ANNUAL CJS BUDGET
1	.000	.100+05	#	X2	CRIME RATE (PER 100,000)
1	.000	.400+02	I	CCL	REP CRIMES/PØLICE ØFFICER
1	.000	.500+03	L	CFK	ØPEN CASES/PRØSECUTØR
1	.000	.200+03	P	CJD	PERCENT PRISØN CAP IN USE

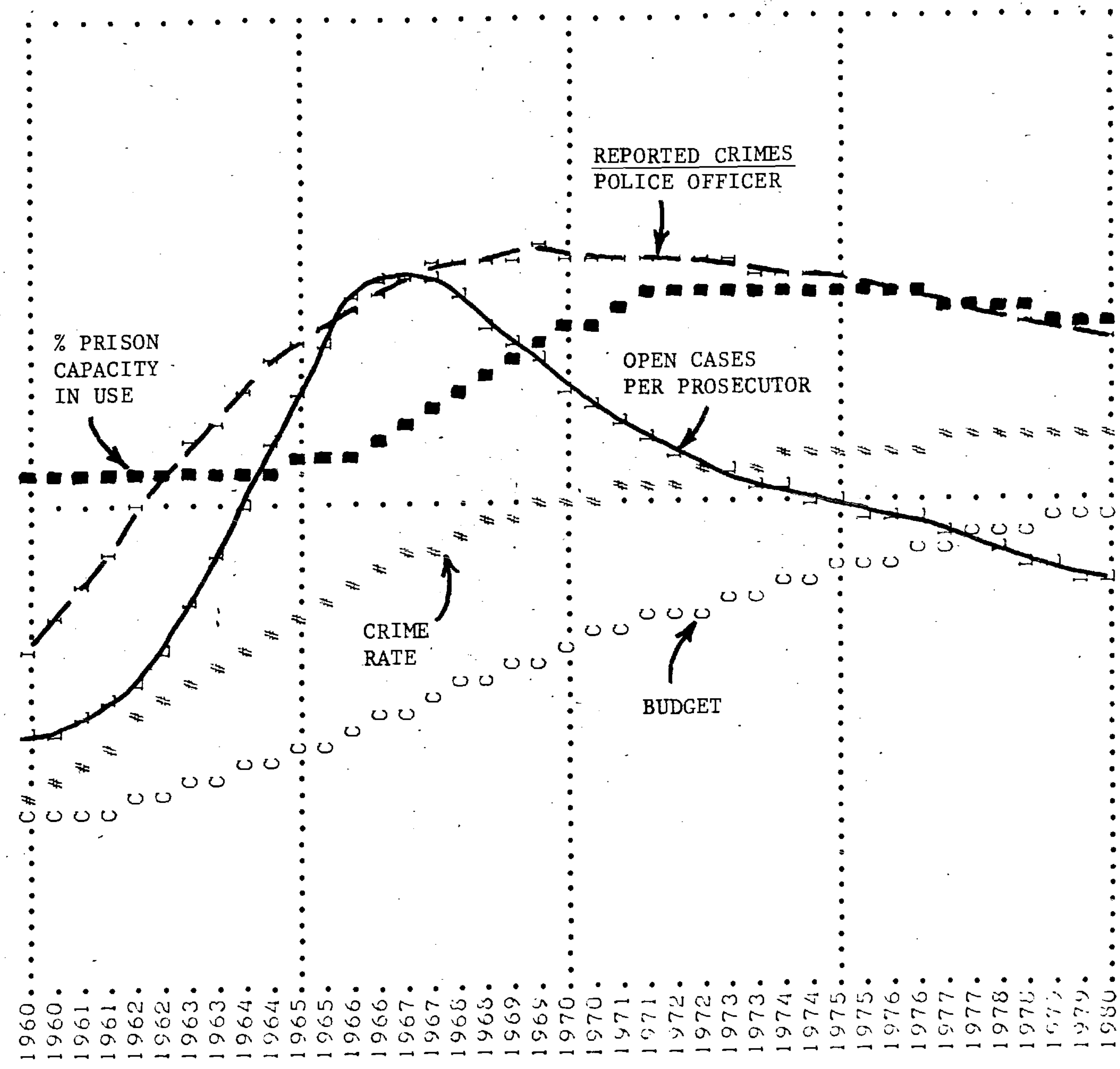


Figure 4. Simulated Time Histories Using the Standard Criminal Justice System Model.

rapid expansion in police officers, prosecutors, and budgets naturally follows the increase in crime.

In the 1970's the model shows slower growth in budget and crime rate and an eventual decline in open cases per prosecutor, reported crimes per officer and percent prison capacity in use. The crime rate slows its growth because the number of prosecutors and police catch up with the earlier crime rate expansion and slow down the flow to crime. Another signification influence is the population distribution that produces fewer youths reaching the crime susceptible ages in the 1970's than in the 1960's. When the crime rate slows its growth at the same time the criminal justice system continues to expand, the measures of system load - reported crimes per officer, open cases per prosecutor, and % prison capacity in use - stop increasing or decline. The criminal justice system resources - police officers, prosecutors, and prison facilities (not shown in the figure) - that rose 30% to 100% in the 1960's slow their rate of increase considerably in the 1970's, though they do not decline.

For training purposes the model is used to illustrate that changes or actions in one part of the criminal justice system (increased police hiring for example) will produce changes in other parts of the system (more arrests, larger case backlog, more plea bargaining, more crowded prisons, etc.). Such changes are shown by making a change in one function and observing the impact on other factors. The impact is determined by simulating the model with the new condition and then comparing the new time histories with the time histories in Figure 4.

C. Model Relationships (Equations and Constants)

Corrections Sector

A1 Prison Capacity on Order (inmate spaces)

equation: A1.K=A1.J+(DT)(AA.JK-AB.JK)

The "prison capacity on order" at any computation time K (as A1.K) equals the prison capacity on order at the previous computation time J (A1.J), plus the new prison capacity ordered between times J and K (DTxAA.JK) minus the old capacity orders that have been filled during DT (DTxAB.JK).

initial value equation: A1=(ABXA)(AAF)

The "initial prison capacity on order" (A1) is determined by both the "delay in receiving the prison capacity ordered" (ABXA) and the "initial average prison capacity loss rate" (AAF).

A2 Prison Capacity (inmate spaces)

equation: A2.K=A2.J+(DT)(AB.JK-AC.JK)

The "prison capacity" at K (A2.K) equals the "prison capacity" at the previous time J (A2.J), plus the new prison capacity created between times J and K (DTxAB.JK) minus the prison capacity retired between J and K (DTxAC.JK).

initial value equation: A2=(.95)(C4)

The "initial prison capacity" (A2) equals, in inmate spaces, 95% of the initial number of prison inmates.

AA Prison Capacity Order Rate (inmate spaces/yr.)

equation: AA.KL=MAX(AAA.K,AAB.K)

The "prison capacity order rate" (AA) is the larger of the minimum order rate (AAA) and a function (AAB) that defines what is otherwise possible and desirable.

<u>AAA</u> <u>Minimum Prison Capacity Order Rate</u> (inmate spaces/yr.)

equation: AAA.K=-A2.K/AAAA

This minimum acts as a limit on how fast (AAAA years) prisons can dispose of capacity that is not needed.

<u>AAF</u> <u>Average Prison Capacity Loss Rate</u> (inmate spaces/yr.)

equation: AAF.K=AAF.J+(DT)(1/AAFD)(AC.JK-AAF.J)

The "average prison capacity loss rate" (AAF) at time <u>K</u> is equal to its previous value at time <u>J</u> plus the difference of the "prison capacity loss rate" (AC) and the average loss rate (AAF) during the interval <u>DT</u> divided by the "averaging time of the prison capacity loss rate" (AAFD).

initial value equation: AAF=A2/ACXA

The initial average prison capacity loss rate (AAF) equals the initial prison capacity divided by the normal cap. deterioration time.

<u>AAB</u> <u>Prison Capacity Order Limit Function</u> (inmate spaces/yr.)

equation: AAB.K=MIN(AAC.K,AAG.K)

The "prison capacity order limit function" (AAB) requires that the "prison capacity order rate needed" (AAC) and the "prison capacity order maximum" (AAG) - the financial ceiling allowed capacity orders - be calculated. The smaller of the two numbers is taken and used to determine <u>AA.K</u>, the prison capacity order rate.

<u>AAC</u> <u>Prison Capacity Order Rate Needed</u> (inmate spaces/yr.)

equation: AAC.K=AAF.K+(1/AACA)(AAD.K-A2.K-A1.K+0+0+0)

The "prison capacity order rate needed" (AAC) equals the average prison capacity loss rate (AAF) plus the quotient of the prison capacity needed (AAD) minus the present prison capacity (A2) and the capacity on order

(A1) divided by the "adjustment time for prison capacity" (AACA) -

AAD Prison Capacity Needed (inmate spaces)

equation: AAD.K=(AADA)(C4.K)+(ABXA)(AAF.K)

The "prison capacity needed (AAD) equals the spaces needed in place (AADA) (C4) plus the spaces needed on order (ABXA)(AAF) to maintain the replacement flow; the product of the "prison capacity needed per prisoner factor" (AADA) and the "number of prisoners" (C4) is the space needed. The "delay in receiving prison capacity ordered" (ABXA) multiplied by the "average prison capacity loss rate" (AAF) is the pipeline need.

AAG Prison Capacity Order Maximum (inmate spaces/yr.)

equation: AAG.K=(1/FACA)(FAC.K+FAJ.K)

The "prison capacity order maximum" (AAG) equals the sum of the "annual prison replacement budget" (FAC) and the "delayed prisons capital budget" (FAJ) - the budget for facility expansion - divided by the "average price per unit capacity" (FACA). AAG is the maximum capacity has been budgeted at time K.

AB Prison Capacity Arrival Rate (inmate spaces/yr.)

equation: AB.KL=A1.K/ABXA

The "prison capacity arrival rate" (AB) equals the "prison capacity on order" (A1) divided by the "delay in receiving prison capacity ordered" (ABXA). This is the equation for a first order delay function.

AC Prison Capacity Loss Rate (inmate spaces/yr.)

equation: AC.KL=A2.K/ACXA

The "prison capacity loss rate" (AC) equals the present "prison capacity" (A2) divided by the "normal capacity deterioration time" (ACXA). This is the equation for a first order delay function.

B1.K Prosecutors Being Hired (prosecutors)

equation: B1.K=B1.J+(DT)(BA.JK-BB.JK)

The "prosecutors being hired" at time K (B1.K) equals the prosecutors being hired at time J (B1.J) plus the prosecutors hired between J and K (DTxBA.JK) minus the prosecutors hired at time J who are on the job at time K (DTxBB.JK).

initial value equation: B1=(BBXA)(BAG)

The "initial prosecutors being hired" (B1) equals the "time to acquire prosecutors" (BBXA) times the "initial average prosecutor loss rate" (BAG); in other words, B1 equals the initial number of prosecutors on the job times the ratio of the time to acquire prosecutors to the normal judge tenure.

B2.K Prosecutors (prosecutors)

equation: B2.K=B2.J+(DT)(BB.JK-BC.JK)

The number of "prosecutors" at time K (B2.K) equals the number at time J (B2.J) plus the prosecutors who arrived on the job between times J & K (DTxBB.JK) minus the prosecutors who leave the job during JK (DTxBC.JK).

initial value equation: B2=B2XJ

The "initial prosecutors" (B2) is equal the constant B2XJ=8. This number is a statistic obtained from the Fulton County Court Administrator.

BA.KL Prosecutor Order Rate (prosecutors/yr.)

equation: BA.KL=MAX(BAA.K,BAB.K)

The "prosecutor order rate" is the larger of the minimum order rate (BAA) and a function (BAB) that defines what is otherwise desirable and possible.

<u>BAA.K</u> <u>Minimum Prosecutor Order Rate</u> (prosecutors/yr.)

equation: BAA.K=-B2.K/BAAA

This minimum acts as a limit on the speed with which unneeded prosecutors can be released.

<u>BAB.K</u> <u>Prosecutors Order Limit Function</u> (prosecutors/yr.)

equation: BAB.K=MIN(BAC.K,BAK.K)

The "prosecutors order limit function" at time <u>K</u> (BAB.K) requires that the "prosecutor order rate needed" at <u>K</u> (BAC.K) and the "prosecutor order maximum" at <u>K</u> (BAK.K) - a financial constraint - be calculated. The smaller of the two numbers equals BAB.K.

<u>BAC.K</u> <u>Prosecutor Order Rate Needed</u> (prosecutors/yr.)

equation: BAC.K=BAG.K+(1/BACA)(BAD.K-B2.K-B1.K+0+0+0)

BAC.K equals the "average prosecutor loss rate" at time K (BAG.K) plus the quotient of the "judges needed" at <u>K</u> (BAD.K) minus the number of prosecutors (B2.K) minus the number of prosecutors being hired at <u>K</u> (B1.K) all divided by BACA, the "adjustment time for prosecutor hiring." The added zero merely serves to meet Dynamo equation format requirements.

<u>BAD.K</u> <u>Prosecutors Needed</u> (prosecutors)

equation: BAD.K=(BBXA)(BAG.K)+(BAE.K)(1)+(BAL.K)(1)+(0)(0)

BAD.K equals the "average prosecutor loss rate" at time K (BAG.K) times the "time to acquire prosecutors" (BBXA) plus the "prosecutors needed to handle the arrest rate" at K (BAE.K) and the "prosecutors needed for load reduction at K (BAL.K).

<u>BAE.K</u> <u>Prosecutors Needed to Handle the Arrest Rate</u> (prosecutors)

equation: BAE.K=BAF.K/CDB.K

BAE.K equals the "average arrest rate" at time K (BAF.K) divided by the "court productivity" (in persons per prosecutor-year) at K (CDB.K).

BAF.K Average Arrest Rate (persons/year)

equation: BAF.K=BAF.J+(DT)(1/BAFD)(CC.JK-BAF.J)

BAF.K equals its value at time J (BAF.J) plus the product of the computation time interval (DT), the inverse of the "averaging time for the arrest rate" (BAFD), and the difference between the actual value of the arrest rate during DT (CC.JK) and its average value at time J (BAF.K).

initial value equation: BAF=(P2)(CCAB)

The "initial average arrest rate" (BAF) equals the product of the "initial number of police" (P2) and the "normal effectiveness of police" (CCAB) - in persons arrested per officer-year.

BAG.K Average Prosecutor Loss Rate (prosecutors/year)

equation: BAG.K=BAG.J+(DT)(1/BAGD)(BC.JK-BAG.J)

BAG.K equals its value at time J (BAG.J) plus the product of the computation time interval (DT), the inverse of the "averaging time for prosecutor loss rate" (BAGD), and the difference between the actual value of the "prosecutor loss rate" (BC.JK) during DT and its average value at time J (BAG.J)

initial value equation: BAG=B2/BCXA

The "initial average prosecutor loss rate" equal the quotient of the initial prosecutors" (B2) and the "normal prosecutor tenure" (BCXA).

BAH.K Prosecutors for Load Reduction (prosecutors)

equation: BAH.K=(1/BAM.K)(C3.K-BAJ.K)

Since BAM.K is merely a dummy variable used to satisfy Dynamo format requirements, the equation should read:

BAH.K=[1/(BAMA)(CDB.K)](C3.K-BAJ.K)

BAH.K equals the product of two factors: the first term is the inverse of the product of the "time to adjust court backlog overload" (BAMA) and the "court productivity" (CDB.K) - this yields a prosecutor per persons factor for load reduction; the second term is the difference between the number of "people in the criminal justice system" at K (C3.K) and the "normal prosecutor load" at K (BAJ.K) - this supplies a direction for the change in the number of prosecutors.

BAJ.K Normal Prosecutor Load (persons)

equation: BAJ.K=(B2.K)(C3XJ)/B2XJ

BAJ.K equals the ratio of "prosecutors" at time K (B2.K) to the "initial prosecutors" (B2XJ) multiplied by the "initial persons with pending cases" (C3XJ).

BAK.K Prosecutor Order Maximum (dimensionless)

equation: BAK.K=(1/FBCA)(FBC.K+FBJ.K)

BAK.K equals the sum of "annual courts replacement budget" at time K (FBC.K) and the "delayed courts capital budget" at K (FBJ.K) divided by the "average cost of a new prosecutor" (FBCA).

BAL.K Prosecutors for Load Reduction (prosecutors)

equation: BAL.K=MAX(BAH.K,Ø)

BAL.K equals the maximum of two numbers: zero and the "prosecutors for load reduction" at K (BAH.K). BAL.K appears in the equation for prosecutors needed (BAH.K). BAL.K will not go negative; negative judges do not exist to reduce load.

BAM.K Factor for BAH.K

equation: BAM.K=(BAMA)(CDB.K)

BAM.K has no meaning as a construct. It merely serves to satisfy Dynamo equation format requirements.

BB.KL Prosecutor Arrival Rate (prosecutors/yr.)

equation: BB.KL=B1.K/BBXA

BB.KL equals the "prosecutors being hired" at K (B1.K) divided by the "time to acquire prosecutors" (BBXA).

BC.KL Prosecutor Loss Rate (prosecutors/yr.)

equation: BC.KL=B2.K/BCXA

BC.KL equals the "prosecutors" at time K (B2.K) divided by the "normal prosecutor tenure" (BCXA).

Criminal Flow Sector

C1 Non-Criminal Population (persons)

equation: C1.K=C1.J+(DT)(CA.JK+CE.JK+CJ.JK-CB.JK+CL.JK+CQ.JK)

The "non-criminal population" at time K (C1.K) equals its value at time J (C1.J) plus the inflows minus the outflow during (DT). The inflows are the net population growth (DTxCA.JK), the persons who reform after experiencing the CJS (CTxCE.JK), the persons who reform after leaving prison (DTxCJ.JK), the persons who reform after being paroled (DTxCL.JK), and the persons who reform after being placed on probation (DTxCQ.JK). The single outflow is the number of non-criminals who become potential active criminals (DTxCB.JK).

initial value equation: C1=C1IA-C2-C3-C4-C5-C6

The "initial non-criminal population" (C1) equals the "initial population" (C1IA) minus the "initial potential active criminals" (C2), the "initial persons with pending cases" (C3), the "initial persons in prison" (C4), the "initial persons on parole" (C5), and the "initial persons on probation" (C6).

C2 Potential Active Criminals (persons)

equation: C2.K=C2.J+(DT)(CB.JK+CD.JK+CH.JK-CC.JK+CM.JK+CP.JK)

The "potential active criminals" at time K (C2.J) equals its value at time J (C2.J) plus the inflows minus the outflow during (DT). The inflows include the non-criminals who enter the C2 group (DTxCB.JK), the persons who do not reform after experience with the CJS (DTxCD.JK), the persons who return to criminal activity after prison (DTxCH.JK), the persons who repeat after parole (CTxCM.JK), and the persons who repeat after probation (DTxCP.JK). The only outflow is the people arrested (DTxCC.JK).

initial value equation: C2=(CCAI)(C2AI)+(CCBA)(-C3)+(CQAA)(-C6)+(CLAA)(-C5)

The "initial potential active criminals" (C2) equals the total active criminals (CCAI)(C2AI) minus the active criminals on bond (CCBA)(C3), on probation (CQAA)(C6), and on parole (CLAA)(C5).

<u>C3</u> <u>Persons in CJS</u> (persons)

equation: C3.K=C3.J+(DT)(CC.JK-CD.JK-CE.JK-CF.JK-CN.JK+0)

The "persons in CJS" at time K (C3.K) equals its value at time J (C3.J) plus the single inflow minus the outflows during (DT). The inflow is the number of people arrested (DTxCC.JK). The outflows are the persons leaving C3 and returning to C2 (DTxCD.JK), those persons returning to C1 (DTxCE.JK), the people sent to prison (DTxCF.JK), and the persons placed on probation (DTxCN.JK). The zero is added so that the equation conforms to the rules of Dynamo.

initial equation: C3=C3XJ

The "initial persons with a pending case" (C3) equals an initial value constant, C3XJ=1000 persons.

<u>C4</u> <u>Persons in Prison</u> (persons)

equation: C4.K=C4.J+(DT)(CF.JK-CG.JK-CH.JK-CJ.JK-CK.JK+0)

"Persons in prison" at time K (C4.K) equals its value at time J (C4.J) plus a single inflow minus four outflows. Added to C4.J are the persons sent to prison (DTxCF.JK). The outflows are the prisoners who die during DT (DTxCG.JK), those prisoners released who return to crime after prison (DTxCH.JK), as well as those who reform (DTxCJ.JK), and lastly the prisoners paroled between times J and K (DTxCK.JK).

initial value equation: C4=(C4IA)(C4IB)/(CGA+CHCA)

"Initial persons in prison" (C4) is equal to the initial flow to prison (C4IA) times a factor (C4IB)/(CGA+CHCA) that represents the average time spent in prison. Since inmates leave through death and release both the average lifetime in prison (CGA) and average sentence (CHCA) must be considered.

<u>C5</u> <u>Persons on Parole</u> (persons)

equation: C5.K=C5.J+(DT)(CK.JK-CL.JK-CM.JK+0)

"Persons on parole" at time K (C5.K) equals its value at time J (C5.J) plus the persons paroled between J and K (DTxCK.JK) minus the persons who reform (DTxCL.JK) and return to criminal activity (DTxCM.JK) after parole.

initial value equation: C5=(C4IA)(CLAD)(CKAA)/((CHCA)(1)(1))

<u>C6</u> <u>Persons on Probation</u> (persons)

equation: C6.K=C6.J+(DT)(CN.JK-CP.JK-CQ.JK+0)

"Persons on probation" at time <u>K</u> (C6.K) equals the value of C6 at time <u>J</u> plus the people placed on probation between <u>J</u> & <u>K</u> (DTxCN.JK) minus the persons who return to crime (DTxCP.JK) and those who reform (DTxCQ.JK) after probation.

initial value equation: C6=(CQAD)(CNAA)(C6IB)

The "initial persons on probation" (C6) equals the total initial case disposition rate (C6IB) times the fraction sentenced to probation (CNAA) times the average time on probation (CQAD).

<u>CA</u> <u>Net Population Growth Rate</u> (persons/yr.)

equation: CA.KL=(CAA.K)(CABA)

The "net population growth rate" between times <u>K</u> & <u>L</u> equals a percentage of the reproductive population (CAA.K), that is, multiplication of CAA.K by a "reproduction factor" (CABA).

CAA Reproductive Population (persons)

equation: CAA.K=CAA.J+(DT)(1/CAAA)(C1.J+C2.J+C3.J+C5.J+C6.J-CAA.J)

The "reproductive population" at time K (CAA.K) is equal to its value at time J (CAA.J) plus the change over (DT) in the number of people in the reproductive population. Notice that persons in prison (C4.J) are excluded from the reproductive population.

initial value equation: CAA=C1+PNA

The "initial reproductive population" equals the initial non-criminal population (C1) plus the initial potentially active criminals.

CB Net Flow to Crime (persons/yr.)

equation: CB.KL=(C1.K)(CBA.K)(CBN.K)

(CBA.K) and (CBN.K) are merely dummy parameters used to meet Dynamo equation format requirements. By substituting their values into the above equation, we obtain a meaningful expression for CB.KL.

CB.KL=(C1.K)(CBR.K)(CBB.K)(CBD.K)(CBF.K)(CBH.K)(CBJ.K)

The "net flow to crime" between times K & L equals the non-criminal population at time K (C1.K) times the normal fraction to crime based on population age (CBR.K) times several influences on the flow to crime: peer pressure (CBB.K), crime reward (CBD.K), police effectiveness (CBF.K), the average fraction of cases sent to prison (CBH.K), and the delayed education budget allocation (CBJ.K).

CBA.K Factor for CB.KL (1/years)

equation: CBA.K=(CBR.K)(CBB.K)(CBD.K)

CBA.K is a dummy parameter used to satisfy Dynamo equation format requirements.

CBB.K Influence of Peer Pressure on Flow to Crime (dimensionless)

equation: CBB.K=TABHL(TCBD,CBE.K,0,20000,4000)

The "influence of peer pressure on flow to crime" at time K (CBB.K) is a function of "delayed criminal-non-criminal interaction" (CBE.K) and the relationship is expressed by the table function TCBD.

(see following page)

CBC.K Delayed Criminal-Non-Criminal Interaction (dimensionless)

equation: CBC.K=CBC.J+(DT)(1/CBCA)(CBP.J-CBC.J)

This is the equation for a first-order information delay function. The "delayed criminal-non-criminal interaction" at time K equals its value at time J plus the product of the ratio of the computation interval (DT) to the "delay time in peer influence on to crime flow" (CBCA) multiplied by the difference between the actual "percent criminals to noncriminals" at time J (CBP.J) and the delayed value at time J (CBC.J).

initial value equation: CBC=(100)(PNA)/C1

The "initial delayed criminal-non-criminal interaction" (CBC) is the percentage of "initial active criminals" (PNA) to the "initial non-criminal population" (C1).

CBD.K Influence of Crime Reward on to Crime (dimensionless)

equation: CBD.K=TABHL(TCBD,CBE.K,0,20000,4000)

The "influence of crime reward on to crime" (CBD.K) is a function of "delayed crime reward information"(CBE.K) and the relationship is described by the table function TCBD.

CBE.K Delayed Crime Reward Information ($/criminal/yr.)

equation: CBE.K=CBE.J+(DT)(1/CBEA)(CBQ.J-CBE.J)

This is the equation for a first-order information delay function. The "delayed crime reward information" at time K (CBE.K) equals its value at time J (CBE.J) plus the ratio of DT to the "delay in crime reward information" (CBEA) times the difference between the actual value of "crime reward" at the J and its delayed value at that time (CBE.J).

initial value equation: CBE=(CBEB)(CCCA)

The "initial delayed crime reward information" equals the product of the "average return per crime" (CBEB) and the "normal reported crimes per criminal" (CCCA).

CBF.K Influen... ...n to Crime (dimensionless)

2.5,.5)

s on to crime" (CBF.K) is a func-

' (CBG.K). The function is TCBF,

C.

J-CBG.J)

...ormation delay. The "delayed

...s factor" at time K (CBG.K) equals its value at time J plus the product of the ratio of computation time (DT) to the delay time for police effectiveness information (CBGA) multiplied by the difference between the actual value of the "fraction of criminals arrested" at time J (CLE.J) and its delayed value at time J (CBG.J).

initial value equation: CBG=1.

CBH.K Influence of Average Fraction of Cases Sent to Prison on Flow to Crime (dimensionless)

equation: CBH.K=TABHL(TCBH,CCE.K,0,.25,.5)

CBH.K is a function of the "average fraction of cases sent to prison" (CCE.K). The relationship is represented as a table function, TCBH.

CBJ.K Influence of Delayed Education Budget Allocation on Flow to Crime (dimensionless)

equation: CBJ.K=TABHL(TCBJ,CBL.K,0,10,2)

CBJ.K is a function of the "delayed budget allocation for education" (CBL.K). The table function, TCBJ, relates these two variables.

CBL.K Delayed Budget Allocation for Education ($/person)

equation: CBL.K=CBL.J+(DT)(1/CBLA)(CLF.J-CBL.J)

CBL.K (at time K) equals its value at time J (CBL.J) plus the ratio of computation time (DT) to the information delay time for education budget allocation (CBLA) multiplied by the difference between the change in the CJS budget allocation for education at time J (CLF.J) and its delayed value at that time (CBL.J).

initial value equation: CBL=3E5/(C1+PNA)

Initial annual crime education budget per citizen equals $300,000.00 divided by the non crime population C1 plus the free criminal population (PNA).

CBM.K Dollars Per Crime ($/crime)

equation: CBM.K=(CBEB)(CCD.K)(CCD.K)/((CCDA)(CCDA))

Crime reward equals normal crime reward (CBEB) times a factor that increases as the square of the criminals' aggressiveness (CCD).

CBN.K Factor For Net Flow to Crime (CB.KL) (dimensionless)

equation: CBN.K=(CBF.K)(CBH.K)(CBJ.K)

CBN.K is a dummy parameter used to satisfy Dynamo equation format requirements.

CBP.K Percent Criminals At Large to Non-Criminals (dimensionless)

equation: CBP.K=(100)(CCB.K)/C1.K

CBP.K equals the percentage: (100) times the ratio of "criminals at large" (CCB.K) to the "non-criminal population" (C1.K).

CBQ.K Crime Reward ($/criminal/yr.)

equation: CBQ.K=(CBM.K)(CCC.K)(CBQA)

CBQ.K equals the product of "dollars per crime" (CBM.K), "reported crimes per criminal per year" (CCC.K), and the "inverse of the fraction of total crimes reported" (CBQA). The product of CCC.K and CBQA yields, of course, the total number of crimes committed per year.

CBR.K Normal Fraction of Non-Criminals to Crime Based on Population Age (1/years)

equation: CBR.K=(CBAA)(CBS.K)

CBR.K equals the "normal fraction of the non-criminal population to crime per year" (CBAA) times the "influence of population age on the normal fraction to crime" (CBS.K).

CBS.K Influence of Population Age on the Normal Fraction to Crime (dimensionless)

equation: CBS.K=TABHL(TCBS,TIME.K,0,14,2)

CBS.K is necessarily a function of TIME.K and is related to it by the table function, TCBS.

CC.KL Arrest Rate (persons/yr.)

equation: CC.KL=(P2.K)(CCAB)(CCW.K)

The "arrest rate" from times K to L (CC.KL) equals the number of police officers times their normal effectiveness (CCAB) in criminals per officer per year times the "influence on police arrest effectiveness factor" (CCW.K).

initial equation: CC=(P2)(CCAB)

The "initial arrest rate" equals the "initial number of police" (P2) multiplied by the "normal police effectiveness" (CCAB).

CCA.K Reported Crimes Per Year (crimes/yr.)

equation: CCA.K=(CCB.K)(CCC.K)

CCA.K equals the product of "criminals at large" (CCB.K) and the "reported crimes per criminal per year" (CCC.K).

CCB.K Criminals At Large (persons)

equation: CCB.K=(C2.K)(1)+(CCBA)(C3.K)+(CQAA)(C6.K)+(CLAA)(C5.K)

CCB.K equals the sum of the following terms: the "potential active criminals" (C2.K)(1) - (1) occurs because of equation format requirements; the fraction free on bond (CCBA) times the person in CJS (C3.K); the fraction that return to crime after probation (CQAA) times the number of persons on probation (C6.K); and the fraction that return to crime after parole (CLAA) times the number of persons on parole (C5.K).

CCC.K Reported Crimes Per Criminal Per Year (crimes/person/yr.)

equation: CCC.K=(CCCA)(CXJ.K)

CCC.K is equal to the product of the "normal reported crimes per criminal per year" (CCCA) and the "influence of criminal aggressiveness on reported crimes per criminal per year" (CXJ.K).

<u>CCD.K</u> <u>Aggressiveness of Criminals</u> (crimes/year)

equation: CCD.K=(CCDA)(CXG.K)(CCV.K)

CCD.K equals the "normal aggressiveness of criminals" (CCDA) times the influence of delayed police load on criminal aggressiveness (CCV.K) times the "influence on criminal aggression of the average fraction of cases sent to prison" (CXG.K).

<u>CCE.K</u> <u>Average Fraction of Cases Sent to Prison</u> (dimensionless)

equation: CCE.K=CCE.J+(DT)(1/CCEA)(CFA.J-CCE.J)

CCE.K equals its value at time <u>J</u> (CCE.J) plus the ratio of computation time to the "average time for flow to prison" (CCEA) multiplied by the difference the "fraction of total cases sent to prison" (CFA.J) and the average value at time <u>J</u> (CCE.J).

initial value equation: CCE=CFJA+(CFBA)(CFCA)

The "initial average fraction of cases sent to prison" (CCE) equals the "fraction of total disposal to prison by plea" (CFJA) plus the product of the "normal fraction of trial cases to prison" (CFBA) and the normal fraction of trials (CFCA).

<u>CCF.K</u> <u>Delayed Current Multiple of Normal Police Load</u> (dimensionless)

equation: CCF.K=CCF.J+(DT)(1/CCFA)(CCG.J-CCF.J)

CCF.K equals its value at time <u>J</u> (CCF.J) plus the product of the ratio of computation time (DT) to the "delay time for load on police" (CCFA) and the difference between the "comparison of current police load to norm" (CCG.J) at time <u>J</u> and its delayed value at J.

initial value equation: CCF=1

The initial (1960) value for the load on police is considered to be the "normal" load.

<u>CCG.K</u> <u>Comparison of Current Police Load to Normal</u> (dimensionless)

equation: CCG.K=(CCB.K)(P2IA)(1)/((P2.K)(PNA)(1))

The equation is written in this manner to fulfil format requirements. It can be rewritten as:

CCG.K=(CCB.K/P2.K)/(PNA/P2IA)

This indicates that CCG.K equals the ratio of "criminals at large" at <u>K</u> (CCB.K) to "police officers" at K (P2.K) divided by the ratio of "initial criminal at large" (PNA) to "initial police officers" (P2IA). The second ratio is defined as the normal police load.

<u>CCH.K</u> <u>Factor for CCW.K</u> (dimensionless)

equation: CCH.K=(CCM.K)(CCP.K)(CCR.K)

CCH.K is a dummy parameter used to satisfy Dynamo equation format requirements.

<u>CCJ.K</u> <u>Influence of Ability to Arrest on Police Effectiveness</u> (dimensionless)

equation: CCJ.K=TABHL(TCCJ,CCK.K,15,40,5)

CCJ.K is a function of "delayed crimes per year per officer" (CCK.K). The relationship is expressed as a table function, TCCJ.

<u>CCK.K</u> <u>Delayed Crimes Per Year Per Officer</u> (crimes/yr./officer)

equation: CCK.K=CCK.J+(DT)(1/CCKA)(CCL.J-CCK.J)

CCK.K equals its value at time <u>J</u> (CCK.J) plus the product of the ratio (DT) to the "information delay time for crimes reported per year per officer" (CCKA) and the difference between the actual value of "crimes per year per officer" at time <u>J</u> (CCL.J) and its delayed value at that time (CCK.J).

initial value equation: CCK=PAE/P2

The "initial delayed crimes per year per officer" equals the "initial average crimes reported per year" (PAE) divided by the "initial police officers" (P2).

CCL.K Crimes Per Year Per Police Officer (crimes/yr./officer)

equation: CCL.K=CCA.K/P2.K

CCL.K equals the quotient of "reported crimes per year" (CCA.K) and "police officers" at time K (P2.K).

CCM.K Influence of Criminal Aggressiveness on the Fraction Arrested (dimensionless)

equation: CCM.K=TABHL(TCCM,CCN.K,0,10,2)

CCM.K is a function of "delayed aggressiveness of criminals information" (CCN.K). The table function relating the two variables is TCCM.

CCN.K Delayed Aggressiveness of Criminals (crimes/yr.)

equation: CCN.K=CCN.J+(DT)(1/CCNA)(CCD.J-CCN.J)

CCN.K equals its value at time J (CCN.J) plus the product of the ratio of (DT) to (CCNA), the delay time for criminal aggressiveness, and the difference of the actual value of "criminal aggressiveness" at time J (CCD.J) and its delayed value at time J (CCN.J).

initial value equation: CCN=CCDA

CCN is equal to CCDA=5, defined as the normal aggressiveness of criminals in crimes per year.

CCP.K Influence of Current Court Load on Fraction of Criminals Arrested (dimensionless)

equation: CCP.K=TABHL(TCCP,CCQ.K,1,5,1)

CCP.K is related to the independent variable CCQ.K - "the delayed comparison of current court load to the normal load" - by the table function TCCP.

CCQ.K Delayed Comparison of Current Court Load to the Normal Load (dimensionless)

equation: CCQ.K=CCQ.J+(DT)(1/CCQA)(CFP.J-CCQ.J)

CCQ.K equals its value at time J (CCQ.J) plus the product of the ratio of (DT) to the "delay time for the load on the courts" (CCQA) and the difference of the actual value at time J of the "comparison of current court load to the normal load" (CFP.J) and its delayed value at J (CCQ.J).

initial value equation: CCQ=1

The initial value for the "delayed comparison of current court load to normal load" equals one because the initial court load is defined as the normal load.

CCR.K Influence of Community Concern on Fraction of Criminals Arrested (dimensionless)

equation: CCR.K=TABHL(TCCR,CCS.K,0,10,2)

CCR.K is a function of "delayed community concern" and is related by the table function TCCR.

CCS.K Delayed Community Concern (dimensionless)

equation: CCS.K=CCS.J+(DT)(1/CCSA)(S1.J-CCS.J)

CCS.K equals its value at time J plus the product of the ratio of (DT) to the "delay time for community concern" (CCSA) and the difference between the actual value of "community concern" at time J (S1.J) and its delayed value at time J (CCS.J).

initial value equation: CCS=S1

"Initial delayed community concern" (CCS) equals "initial community concern" (S1).

<u>CCV.K</u> <u>Influence of Delayed Multiple of Normal Police Load on Criminal Aggressiveness</u> (dimensionless)

equation: CCV.K=TABHL(TCCF,CCF.K,1,5,1)

CCV.K is dependent on the variable CCF.K "delayed current multiple of normal police load," and is related to CCF.K by the table function TCCF.

<u>CCW.K</u> <u>Influence on Police Arrest Effectiveness Factor</u> (dimensionless)

equation: (CCH.K)(CCX.K)(CCJ.K)

Since CCH.K is a dummy parameter, the equation should read: CCW.K= (CCM.K)(CCP.K)(CCR.K)(CCX.K)(CCJ.K).

CCW.K equals the product of five influences on police arrest effectiveness: criminal aggressiveness (CCM.K), current court load (CCP.K), community concern (CCR.K), dollars budgetted per officer (CCX), and crimes per year per officer (CCJ).

<u>CCX.K</u> <u>Influence of Delayed Dollars Budgetted Per Officer on Fraction Criminals Arrested</u> (dimensionless)

equation: CCX.K=TABHL(TCCX,CCZ.K,7000,17000,2000)

CCX.K is a function of "delayed dollars budgetted per police officer" (CCZ.K). They are related by the table function TCCX.

<u>CCY.K</u> <u>Dollars Budgetted Per Officer</u> ($/officer)

equation: CCY.K=FPA.K/P2.K

CCY.K equals the "total annual police budget" (FPA.K) divided by the number of "police officers" (P2.K).

<u>CCZ.K</u> <u>Delayed Dollars Budgetted Per Officer</u> ($/officer)

equation: CCZ.K=CCZ.J+(DT)(1/CCZD)(CCY.J-CCZ.J)

CCZ.K equals its value at time <u>J</u> (CCZ.J) plus the product of the computation time (DT), the inverse of the "delay time for the operation budget per officer" (CCZD), and the difference between the actual value for the "dollars budgetted per officer" at time J (CCY.J) and its delayed value at J (CCZ.J).

initial value equation: CCZ=(1.1)(YCZ)

CD.KL Persons Repeating After CJS (person)

equation: CD.KL=(CDC.K)(1)+(CDC.K)(-CEA.K) or CD.KL=(CDC.K)(1-CEA.K)

CD.KL equals the "number of cases not sent to prison or probated" at time K (CDC.K) multiplied by one minus the "fraction of cases dropped who reform" at K (CEA.K) - in other words, CDC.K times the fraction of cases dropped who repeat at time K.

CDA.K Total Cases Decided (cases/yr.)

equation: CDA.K=MIN(CDD.K,CFH.K)

CDA.K equals the smaller of two numbers: the "maximum disposal rate due to prosecutors" at time K (CDD.K) and the "administrative limit on disposal rate due to load" at K (CFH.K).

CDB.K Court Productivity (person/prosecutor/yr.)

equation: CDB.K=(CDBA)(CXD.K)

CDB.K equals the product of the "normal effectiveness of prosecutors" (CDBA) and a weighting factor, "the influence of current load on court productivity" at time K (CXD.K).

CDC.K Cases Dropped (cases/yr.)

equation: CDC.K=(CDA.K)(RCD.K)(1)+(CDA.K)(RCD.K)(-CNA.K)+(Ø)(Ø)(Ø)

or CDC.K=(CDA.K)(RCD.K)(1-CNA.K)

CDC.K equals the product of the "total cases decided" at time K (CDA.K), "the fraction of cases not sent to prison" at K (RCD.K), and the fraction of cases not sent to prison that are not probated at K (1-CNA.K).

CDD.K Maximum Disposal Rate Due to Prosecutors (persons/yr.)

equation: CDD.K=(B2.K)(CDB.K)

CDD.K equals the product of the "number of prosecutors" at time K (B2.K) and the "court productivity" at K (CDB.K). The units of court productivity are persons per prosecutor per year.

CE.KL Flow to Reform After CJS (persons/yr.)

equation: CE.KL=(CDC.K)(CEA.K)

CE.KL equals the product of the "cases dropped" at time K (CDC.K) and the "fraction of cases dropped who reform" at K (CEA.K).

CEA.K Fraction of Cases Dropped Who Reform (dimensionless)

equation: CEA.K=(CEAA)(CXRK)

CEA.K equals the product of the "normal fraction of cases dropped who reform" and a weighting factor, "the influence of delayed load information on the fraction of reform after CJS" at time K (CXR.K).

CEB.K Delayed Influence of Current Load on Fraction Trials

equation: CEB.K=CEB.J+(DT)(1/CEBA)(CLG.J-CEB.J)

CEB.K equals its value at time J (CEB.J) plus the product of the computation time (DT), the inverse of the "delay time for influence of court load on fraction trials" (CEBA), and the difference between the actual value of the "influence of current load on fraction of trials" at time J (CLG.J) and its delayed value at time J (CEB.J).

initial value equation: CEB=1

The "initial delayed influence of current load on fraction trials" is one. The initial load is the normal load and its influence on trials is one - there is no initial effect.

CF.KL Rate Sent to Prison (persons/yr.)

equation: CF.KL=(CDA.K)(CFA.K)

CF.KL equals the product of the "total cases decided" at time K (CDA.K) and the "fraction of total cases sent to prison" at K (CFA.K).

CFA.K Fraction of Total Cases Sent to Prison (dimensionless)

equation: CFA.K=(CFJA)(CFJ.K)+(CFB.K)(CFC.K)

CFA.K equals the sum of two products: the first is the product of the "fraction of total cases decided to prison by plea" (CFJ.K); the second is the product of the "fraction of trial cases sent to prison" at K (CFB.K) and the "fraction of trials" at K (CFC.K).

CFB.K Fraction of Trial Cases to Prison (dimensionless)

equation: CFB.K=(CFBA)(CXS.K)

CFB.K equals "normal fraction of trial cases to prison" (CFBA) multiplied by the "influence of prison capacity in use on fraction trial cases sent to prison" at time K (CXS.K).

CFC.K Fraction of Trials (dimensionless)

equation: CFC.K=(CFCA)(CFN.K)

CFC.K equals the "normal fraction of cases that go to trial" (CFCA) multiplied by the "influence of current load on the fraction of trials" at time K (CFN.K).

CFE.K Average Trial Delay (years)

equation: CFE.K=(1/CFC.K)(CFG.K-CXK.K)

CFE.K equals the difference of the "average criminal justice system delay" at K (CFG.K) and the "criminal justice system delay due to non-trial cases" at K (CXK.K), and that difference is divided by the "fraction of trials" at K (CFC.K).

CFG.K Average CJS Delay (years)

equation: CFG.K=CFG.J+(DT)(1/CFGA)(CLA.J-CFG.J)

CFG.K equals its value at time J plus the computation time (DT) multiplied by the inverse of the "average time for case disposition delay" (CFGA) times the difference between the "case disposition delay" (CLA.J) at J and the value of the "average CJS delay" at time J.

initial value equation: CFG=MAX(CNA,.1)

The "initial average CJS delay "equals the maximum of the initial CJS delay (CNA) as measured by the open cases (C3) and the disposition rate and the minimum practical delay of .1 year.

CFH.K Administrative Limit on Disposal Rate Due to Load (persons/yr.)

equation: CFH.K=C3.K/CXL.K

CFH.K equals the "persons in the criminal justice system" at time K (C3.K) divided by the "average criminal justice delay" at K (CXL.K).

CFJ.K Fraction of Pleas Sent to Prison (dimensionless)

equation: CFJ.K=TABHL(TCFJ,CFP.K,0,5,1)

CFJ.K is a function of the "comparison of current prosecutor load to the normal (1960) load" at time K (CFP.K). The relationship is expressed as a table function, TCFJ.

CFK.K Cases Per Prosecutor (Current Load) (persons/prosecutor)

equation: CFK.K=C3.K/B2.K

CFK.K equals the "persons in the criminal justice system" at time K

(C3.K) divided by the number of "prosecutors" at K (B2.K).

CFL.K Average Rate of Persons Sent to Prison (persons/year)

equation: CFL.K=CFL.J+(DT)(1/CFLD)(CF.JK-CFL.J)

CFL.K equals its value at time J (CFL.J) plus the product of the computation time (DT), the inverse of the "average time for flow to prison" (CFLD), and the difference between the "rate of persons sent to prison" between times J and K (CF.JK) and its average value at J (CFL.J). CFL.K does not appear in other equations; however, it is a bit of information which is frequently wanted.

initial value equation: CFL=(.2)(B2)(CDBA)

The "initial average rate sent to prison" (CFL) equals two tenths of the initial persons (cases) processed by the prosecutorial staff in 1960: that is, (.2) times the "initial number of prosecutors" (B2) times the "normal effectiveness of prosecutors" (CDBA).

CFM.K Average Cases Per Prosecutor (Current Load) (persons/prosecutor)

equation: CFM.K=CFM.J+(DT)(1/CFMD)(CFK.J-CFM.J)

CFM.K equals its value at time J plus the product of the computation time (DT), the inverse of the "average time for court load" (CFMD), and the difference between the current load at time J (CFK.J) and its average value (CFM.J) at time J.

initial value equation: CFM=C3/B2

The "initial average cases per prosecutor" equals the quotient of the "initial persons in the criminal justice system" (C3) and the "initial prosecutors" (B2).

CFN.K Influence of Current Load on Fraction Trials (dimensionless)

equation: CFN.K=TABHL(TCFC,CFP.K,0,5,1)

CFN.K is a function of the "comparison of current load to normal load" at time K (CFP.K). The relationship is defined by the table function, TCFC.

CFP.K Comparison of Current Load to Normal Load (dimensionless)

equation: CFP.K=(CFM.K)(B2XJ)/C3XJ or CFP.K=CFM.K/[C3XJ/B2XJ]

CFP.K equals the "average cases per prosecutor" at time K CFM.K divided by the ratio of the "initial people (cases) in the CJS" (C3XJ) to the "initial prosecutors" (B2XJ). This ratio (the divisor) is the normal cases per prosecutor on the normal load.

CG.KL Prison Death Rate (persons/yr.)

equation: CG.KL=C4.K/CGA

CG.KL equals the "persons in prison" at time K (C4.K) divided by the "average lifetime in prison" (CGA).

CH.KL Repeat Rate After Prison (persons/yr.)

equation: CH.KL=(CHA.K)(CHE.K)(1)+(CHA.K)(CHE.K)(-CJA.K)+(0)(0)(0)

or CH.KL=(CHA.K)(CHE.K)(1-CJA.K)

The "rate of persons who return to or repeat criminal activities after release from prison" between computation times K & L (CH.KL) equals the product of the "average number of prisoners released per year" at K (CHA.K), the "normal fraction of prisoners released free without parole restrictions" at K (CHE.K), and one minus the "fraction of inmates who reform after prison" at K (CJA.K) (i.e., (1-CJA.K) equals the fraction of freed prisoners who do not reform after prison).

CHA.K Average Prisoners Released Per Year (prisoners/yr.)

equation: CHA.K=C4.K/CHB.K

CHA.K equals the number of "persons in prison" at time K (C4.K) divided by the "delayed average sentence time" at K (CHB.K). A delayed time is used because while the average sentence time might change (probably decrease), the effect of prison release rate is delayed.

CHB.K Delayed Average Sentence Time (years)

equation: CHB.K=CHB.J+(DT)(1/CHBA)(CHC.J-CHB.J)

CHB.K equals its value at time J plus the product of (DT), the inverse of the "delay time for average sentence time" (CHBA), and the difference between the "average sentence time" at J (CHC.J) and its delayed value at J (CHB.J).

initial value equation: CHB-CHCA

The "initial delayed average sentence time" (CHB) equals the "normal average sentence time" (CHCA).

CHC.K Average Sentence Time (years)

equation: CHC.K=(CHCA)(CXT.K)

CHC.K equals the product of the "normal average sentence time" and the "influence of delayed percent prison capacity in use on sentence time" at time K (CXT.K).

CHD.K Delayed Percent Prison Capacity in Use (dimensionless)

equation: CHD.K=CHD.J+(DT)(1/CHDA)(CJD.J-CHD.J)

CHD.K equals its value at time J (CHD.J) plus the product of (DT), the inverse of the "delay time for the percent of prison capacity in use" (CHDA) and the difference of the "percent prison capacity in use" at time J (CJD.J) and its delayed value at J (CHD.J).

initial value equation: CHD=(100)(C4)/A2

The "initial delayed percent prison capacity in use" (CHD) equals (100) times the ratio of "initial persons in prison" (C4) to "initial prison capacity" (A2) (or initial inmate spaces).

CHE.K Normal Fraction People Released Free From Prison (dimensionless)

equation: CHE.K=1-CHA.K

CHE.K equals one minus "the normal fraction of people released from prison on parole "at time K (CKA.K).

CJ.KL Rate of Reform After Prison (persons/yr.)

equation: CJ.KL=(CJA.K)(CHA.K)(CHE.K)

CJ.KL equals the product of the "average prisoners released per year" at time K (CHA.K), the "normal fraction prisoners released free - not on parole" at K (CHE.K), and the "fraction of freed inmates who reform after prison" at K (CJA.K).

CJA.K Fraction Freed Inmates Who Reform After Prison (dimensionless)

equation: CJA.K=(CJAA)(CJB.K)

CJA.K equals the product of the "normal fraction of people who reform from prison" (CJAA) and the "influence of prison conditions on reform" at time K (CJB.K).

CJB.K Influence of Prison Conditions on Reform (dimensionless)

equation: CJB.K=TABHL(TCJB,CJC.K,60,140,20)

CJB.K is a function of the "average percent prison capacity in use" at time K (CJC.K). The relationship is described by a table function, TCJB.

CJC.K Average Percent Prison Capacity in Use (dimensionless)

equation: CJC.K=CJC.J+(DT)(1/CJCA)(CJD.J-CJC.J)

CJC.K equals its value at time J (CJC.J) plus the product of (DT), the inverse of the "average time for percent prison capacity in use" (CJCA), and the difference between the "percent prison capacity in use" at J (CJD.J) and its average value at time J (CJC.J).

initial value equation: CJC=CHD

The "initial average percent prison capacity in use" (CJC) equals the "initial delayed percent prison capacity in use" (CHD).

CJD.K Percent Prison Capacity in Use (dimensionless)

equation: CJD.K=(100)(C4.K)/A2.K

CJD.K equals the ratio of "persons in prison" at time K (C4.K) to "prison capacity" at K (A2.K - in units of inmate spaces) multiplied by (100).

CK.KL Parole Rate (persons/yr.)

equation: CK.KL=(CHA.K)(CKA.K)

CK.KL equals the "average prisoners released per year" at time K (CHA.K) multiplied by the "normal fraction of prisoners released on parole" at K (CKA.K).

CKA.K Normal Fraction of Prisoners Released on Parole (dimensionless)

equation: CKA.K=CKAA

CKA.K is a constant, CKAA.

CL.KL Rate of Reform After Parole (persons/yr.)

equation: CL.KL=(C5.K)(CLH.K)/CLAD

CL.KL equals the number of "persons on parole" at time K (C5.K) times the "fraction persons who reform after parole" at K (CLH.K) divided by the "average time on parole" (CLAD). That is, CL.KL equals the number of people who reform after parole multiplied by the fraction of people who go off parole per year (1/CLAD).

CLA.K Case Disposition Delay (years)

equation: CLA.K=C3.K/CDA.K

CLA.K equals the number of "persons (cases) in the CJS" at time K (C3.K) divided by the "total cases decided per year" at K (CDA.K).

CLC.K Factor for CLE.K

equation: CLC.K=(CCP.K)(CCM.K)

CLC.K is a dummy variable required to satisfy Dynamo equation format rules.

CLD.K Factor for CLE.K

equation: CLD.K=(CCR.K)(CCJ.K)

CLD.K is a dummy variable required to satisfy Dynamo equation format rules.

CLE.K Fraction Criminals Arrested (dimensionless)

equation: CLE.K=(CLD.K)(CLC.K)(CCX.K)

or CLE.K=(CCR.K)(CCJ.K)(CCP.K)(CCM.K)(CCX.K)

CLE.K equals the product of five terms: "the influence of community concern on the fraction of criminals arrested" at time K (CCR.K), "the ability to arrest (influence of delayed crimes per officer per year)" at K (CCJ.K), "the influence of current court load on the fraction of criminals arrested" at K (CCP.K), "the influence of criminal aggressiveness on the fraction arrested" at K (CCM.K), and "the influence on police arrest effectiveness factor" at K (CCW.K).

CLF.K Dollar Change in CJS Budget per Person For Education ($/person)

equation: CLF.K=FEA.K/(C1.K+CCB.K)

CLF.K equals the "total annual CJS education budget" at time K (FEA.K) divided by the sum of the "non-criminals" at K (C1.K) and the "criminals at large" at K (CCB.K) - i.e., the entire population.

CLG.K Influence of Current Load on Fraction of Trials (dimensionless)

equation: CLG.K=CFC.K/CFCA=CFN.K

See CFN for description.

CLH.K Fraction Persons Who Reform After Parole (dimensionless)

equation: CLH.K=1-CLAA

CLH.K equals a constant, (1-CLAA), or one minus "the fraction who report after parole" (CLAA).

CM.KL Repeat Rate After Parole (persons/yr.)

equation: CM.KL=(1/CLAD)((C5.K)(1)+(C5.K)(-CLH.K))

or CM.KL=(1/CLAD)((C5.K)(1-CLH.K))

CM.KL equals the number of "people on parole" at time K (C5.K) times one minus the "fraction who reform after parole" at K (CLH.K) multiplied by the inverse of the "average time on parole" (CLAD). In other words, CM.KL equals the number of people who reform after parole times the fraction of parolees released per year.

CN.KL Probation Rate (persons/yr.)

equation: CN.KL=(CDA.K)(RCD.K)(CNA.K)

CN.KL equals the product of three terms: "the total cases decided per year" at time K (CDA.K), "the fraction of total cases not sent to prison" at K (RCD.K), and "the fraction of cases not sent to prison but probated" at K (CNA.K).

CNA.K Fraction Cases Not Sent to Prison but Probated (dimensionless)

equation: CNA.K=CNAA

CNA.K equals a constant, CNAA - "the fraction of people released on probation."

CP.KL Repeat Rate After Probation (persons/yr.)

equation: CP.KL=(1/CQAD)((C6.K)(1)+(C6.K)(-CQA.K))

or CP.KL=(1/CQAD)((C6.K)(1-CQA.K))

CP.KL equals the product of the number of "people on probation" at time K (C6.K) and one minus the fraction of people who reform after probation" at K (CQA.K) divided by the "average time on probation" (CQAD). That is, CP.KL equals the number of people who repeat after probation times the fraction of those people released per year.

CQ.KL Reform Rate After Probation (persons/yr.)

equation: CQ.KL=(C6.K)(CQA.K)/CQAD

CQ.KL equals the product of the number of "persons on probation" at time K (C6.K) and "the fraction of persons who reform after probation" at K (CQA.K) multiplied by the fraction of people released from probation per year (1/CQAD).

CQA.K Fraction of People Who Reform After Probation (dimensionless)

equation: CQA.K=1-CQAA

CQA.K equals one minus a constant, CQAA - "the fraction who repeat after probation."

CXC.K Influence of Current Load on Fraction of Trials (dimensionless)

equation: CXC.K=CFC.K/CFCA=CFN.K

See CFN for description.

CXD.K Influences of Current Load on Court Productivity (dimensionless)

equation: CXD.K=TABHL(TCDB,CXC.K,0,1.4,.2)

CXD.K is a function of CXC.K - "the influence of current load on the fraction trials." The relationship is expressed as a table function, TCDB.

CXG.K Influence of Average Fraction Cases Sent to Prison On Criminal Aggressiveness (dimensionless)

equation: CXG.K=TABHL(TCCD,CCE.K,0,.25,.05)

CXG.K is a function of the "average fraction of cases sent to prison" at time K (CCE.K). The relationship is expressed as a table function, TCCD.

CXJ.K Influence of Criminal Aggressiveness on the Crimes Committed Per Criminal (dimensionless)

equation: CXJ.K=TABHL(TCCC,CCD.K,0,10,2)

CXJ.K is a function of the "aggressiveness of criminals" at time K (CCD.K). The relationship is expressed as a table function, TCCC.

CXK.K CJS Delay Due to Non-Trial Cases (years)

equation: CXK.K=CFEA+(CFEA)(-CFC.K)

or CXK.K=CFEA(1-CFC.K)

CXK.K equals the product of the "normal delay for non-trial cases" (CFEA) and one minus the "fraction of trials" at K (CFC.K); or, CXK.K equals the delay for non-trial cases times the fraction of non-trial cases.

CXL.K Average CJS Delay (years)

equation: CXL.K=CXK.K+(CFFA)(CFC.K)

CXL.K equals the "CJS delay due to non-trial cases" plus the CJS delay due to trial cases - i.e., the product of "the normal delay for trial cases" (CFFA) and "the fraction of trials" at time K (CFC.K).

CXR.K Influence of Delayed Load Information on the Fraction Who Reform After CJS (dimensionless)

equation: CXR.K=TABHL(TCEA,CEB.K,0,2.5,.5)

CXR.K is a function of "the delayed influence of current load on the fraction of trials" at time K (CEB.K). The relationship is expressed by the table function, TCEA.

CXS.K Influence of Prison Capacity in Use on Fraction of Trials to Prison (dimensionless)

equation: CXS.K=TABHL(TCFB,CHD.K,60,140,20)

CXS.K is a function of the "delayed percent of prison capacity in use" at time K (CHD.K). The relationship is expressed by the table function, TCFB.

CXT.K Influence of Prison Capacity in Use on Length of Sentence Time (dimensionless)

equation: CXT.K=TABHL(TCHC,CHD.K,60,140,20)

CXT.K is a function of the "delayed percent of prison capacity in use" at time K (CHD.K). The relationship is expressed by the table function, TCHC.

RCD.K Fraction of Total Cases Not Sent to Prison (dimensionless)

equation: RCD.K=1-CFA.K

RCD.K equals one minus "the fraction of total cases sent to prison" at time K (CFA.K).

Financial Sector

F1 Total Annual CJS Budget (dollars/yr.)

equation: F1.K=FPA.K+FAA.K+FBA.K+FEA.K

The "total annual CJS budget" (F1) equals the sum of the "total annual police budget" (FPA), the total annual corrections budget (FAA), the total annual courts budget (FBA), and the total annual CJS education budget.

FAA Total Annual Corrections Budget (dollars/yr.)

equation: FAA.K=FAB.K+FAC.K+FAD.K+FAK.K

The "total annual corrections budget" (FAA) equals the sum of the "annual corrections operating budget" (FAB), the "annual prison replacement budget" (FAC), the "annual prisons capital budget" (FAD), and the "annual outside support for corrections" (FAK).

FAB Annual Corrections Operating Budget (dollars/yr.)

equation: FAB.K=(FABA)(A2.K)+(FABC)(C4.K)+(FABE)(C5.K)+(0)(0)

The "annual corrections operating budget" (FAB) equals the sum of the operating costs for the prison capacity - FABA (the annual operating cost per prison capacity unit) times A2.K (the number of prison capacity units); for the prisoners - FABC (average annual operating cost per prisoner) times C4.K (the number of prisoners); and for parolees - FABE (annual cost per parolee) times C5.K (the number of parolees).

FAC.K Annual Prison Replacement Budget (dollars/yr.)

equation: FAC.K=(FACA)(AAF.K)

FAC.K equals the "average price per unit capacity" (FACA) times the "average prison capacity loss rate" at time K (AAF.K).

FAD.K Annual Prison Capital Budget (dollars/yr.)

equation: FAD.K=MIN(FAE.K,FAF.K)

FAD.K equals the smaller of the "maximum prisons expansion rate" at time K (FAE.K) and the "maximum prison budget factor" at K (FAF.K).

FAE.J Maximum Prisons Expansion Rate (dollars/yr.)

equation: FAE.J=(FAEA)(FACA)(A2.K)

FAE.J equals the product of three terms: "the maximum prison expansion fraction" (FAEA), "the average price per unit of capacity" (FACA), and "the prison capacity" at time K (A2.K).

FAF.J Maximum Prison Budget Factor (dollars/yr.)

equation: FAF.J=(FAG.K)(FAB.K+FAC.K)

FAF.J equals the product of the "fraction of the corrections operating budget for expansion" at time K (FAG.K) and the operating budget at K - which includes the "annual corrections operating budget" (FAB.K) and the "annual prison replacement budget" (FAC.K).

FAG.K Fraction Operating Budget for Expansion (dimensionless)

equation: FAG.K=TABHL(FAGT,FAH.K,0,10,2)

FAG.K is a function of "average community concern" at time K (FAH.K). The relationship is expressed as a table function, FAGT.

FAH.K Average Community Concern - Prisons (dimensionless)

equation: FAH.K=FAH.J+(DT)(1/FAHD)(S1.J-FAH.J)

FAH.K equals its value at time J (FAH.J) plus the product of the computation time (DT), the inverse of the "average time for community concern - prisons" (FAHD), and the difference between "community concern - CJS" at time J (S1.J) and its average value at J (FAH.J).

initial value equation: FAH=S1

The "initial average community concern - prisons" equals the "initial community concern - CJS."

FAJ.K Delayed Prisons Capital Budget ($/year)

equation: FAJ.K=FAJ.J+(DT)(1/FAJD)(FAD.J-FAJ.J)

FAJ.K equals its value at time J (FAJ.J) plus the product of (DT), the inverse of the "delay for corrections capital budget" (FAJD), and the difference between the "annual prisons capital budget" at J (FAD.J) and its average value at J (FAJ.J).

initial value equation: FAJ=0

The "initial delay in prisons capital budget" equals zero.

FAK.K Annual Outside Support for Corrections ($/year)

equation: FAK.K=FAKA

FAK.K equals a constant, "the annual outside support for corrections" (FAKA).

FBA.K Total Annual Courts Budget ($/yr.)

equation: FBA.K=FBB.K+FBC.K+FBD.K+FBK.K

FBA.K equals the "annual courts operation budget" at time K (FBB.K), plus the "annual courts replacement budget" at K (FBC.K), plus the "annual courts capital budget" at K (FBD.K), plus the "annual outside supports for the courts" at K (FBK.K).

FBB.K Annual Courts Operation Budget ($/yr.)

equation: FBB.K=(FBBA)(B2.K)+(FBBC)(C6.K)

FBB.K equals the product of the "annual operating cost per prosecutor" (FBBA) and the number of "prosecutors" at time K (B2.K) plus the product of the "annual cost per probation" (FBBC) and the number of "probationers" at K (C6.K).

FBC.K Annual Courts Replacement Budget ($/yr.)

equation: FBC.K=(FBCA)(BAG.K)

FBC.K equals the product of the "average cost of a new prosecutor" (FBCA) and the "average prosecutor low rate" at time K (BAG.K).

FBD.K Annual Courts Capital Budget ($/yr.)

equation: FBD.K=MIN(FBE.K,FBF.K)

FBD.K equals the smaller of the "maximum courts expansion rate" at time K (FBE.K) and the "maximum courts budget factor" at K (FBF.K).

FBE.K Maximum Courts Expansion Rate ($/yr.)

equation: FBE.K=(FBEA)(FBCA)(B2.K)

FBE.K is defined by a product of three terms: the "maximum courts expansion factor" (FBEA), the "average cost of a new prosecutor" (FBCA), and the number of "prosecutors" at time K (B2.K).

FBF.K Maximum Courts Budget Factor ($/yr.)

equation: FBF.K=(FBG.K)(FBB.K+FBC.K)

FBF.K equals the sum of the "annual courts operation budget at time K (FBB.K) and the "annual courts replacement budget" at K (FBC.K) multiplied by the "fraction of the operation budget for expansion" at K (FBG.K).

FBG.K Fraction of the Operation Budget for Expansion (dimensionless)

equation: FBG.K= TABHL(FBGT,FBH.K,0,10,2)

FBG.K is a function of "average community concern - courts" at time K (FBH.K). The relationship is defined by the table function, FBGT.

FBH.K Average Community Concern - Courts (dimensionless)

equation: FBH.K=FBH.J+(DT)(1/FBHD)(S1.J-FBH.J)

FBH.K equals its value at time J (FBH.J) plus the product of (DT), the inverse of the "average time for community concern - courts" (FBHD), and the difference between the "community concern - CJS" at time J and its average value at J (FBH.J).

initial value equation: FBH=S1

"Initial average community concern - courts" (FBH) equals the "initial average community concern" (S1).

FBJ.K Delayed Courts Capital Budget ($/yr.)

equation: FBJ.K=FBJ.J+(DT)(1/FBJD)(FBD.J-FBJ.J)

FBJ.K equals its value at time J (FBJ.J) plus the product of (DT), the inverse of the "delay for the courts capital budget" (FBJD), and the difference of the "annual courts capital budget" at time J (FBD.J) and its average value at time J (FBJ.J).

initial value equation: FBJ=0

Initial delayed courts capital budget equals zero.

FBK.K Annual Outside Support for Courts ($/yr.)

equation: FBK.K=FBKA

FBK.K equals the "annual outside support for courts," FBKA - a constant.

FEA.K Total Annual CJS Education Budget ($/yr.)

equation: FEA.K=(FEB.K)(FPA.K+FAA.K+FBA.K+0)

FEA.K equals the sum of the "total annual police budget" at time K (FPA.K), the "total annual corrections budget" at K (FAA.K), and the "total annual courts budget" at K (FBA.K) multiplied by the "fraction CJS budget for education" at time K (FEB.K).

FEB.K Fraction CJS Budget for Education (dimensionless)

equation: FEB.K=TABHL(TFEB,FEC.K,0,10,2)

FEB.K is a function of "average community concern - education" (FEC.K). The relationship is defined by the table function, TFEB.

FEC.K Average Community Concern - Education (dimensionless)

equation: FEC.K=FEC.J+(DT)(1/FECD)(S1.J-FEC.J)

FEC.K equals its value at time J (FEC.J) plus the product of (DT), the inverse of the "average time for community concern" (FECD), and the difference of the "community concern - CJS" at time J (S1.J) and its average value at J (FEC.J).

initial value equation: FEC=S1

FEC equals the "initial community concern - CJS" (S1).

FPA.K Total Annual Police Budget ($/yr.)

equation: FPA.K=FPB.K+FPC.K+FPD.K+FPE.K

FPA.K equals the sum of the "annual police operation budget" at time K (FPB.K), the "annual police replacement budget" at K (FPC.K), the "annual outside support for police" at K (FPD.K), and the "annual police capital budget" at K (FPE.K).

FPB.K Annual Police Operation Budget ($/yr.)

equation: FPB.K=(FPP.K)(P2.K)

FPB.K equals the product of the "cost per officer per year" at time K (FPP.K) and the number of "police officers" at K (P2.K).

FPC.K Annual Police Replacement Budget ($/yr.)

equation: FPC.K=(PEB)(PAF.K)

FPC.K equals the product of the "normal acquisition cost per officer" (PEB) and the "average police loss rate" at time K (PAF.K).

FPD.K Annual Outside Support for Police ($/yr.)

equation: FPD.K=FPDA

FPD.K equals a constant, "the annual outside support for police" (FPDA).

FPE.K Annual Police Capital Budget ($/yr.)

equation: FPE.K=MIN(FPH.K,FPK.K)

FPE.K equals the smaller of the "maximum police expansion rate" at time K (FPH.K) and the "maximum police budget factor" at K (FPK.K).

FPH.K Maximum Police Expansion Rate ($/yr.)

equation: FPH.K=(FPHA)(PEB)(P2.K)

FPH.K equals the product of three terms: the "maximum police expansion factor" (FPHA), the "normal acquisition cost per officer" (PEB), and the number of "police officers" at time K (P2.K).

FPJ.K Cost of Needed Police Orders ($)

equation: FPJ.K=(PEB)(PAC.K)

FPJ.K equals the product of the "normal acquisition cost per officer" (PEB) and the "police order rate needed" at time K (PAC.K).

FPK.K Maximum Police Budget Factor

equation: FPK.K=(FPL.K)(FPB.K+FPC.K)

FPK.K equals the sum of the "annual police operation budget" at time K (FPB.K) and the "annual police replacement budget" at K (FPC.K) multiplied by the "fraction of the operation budget for expansion at time K (FPL.K).

FPL.K Fraction Operation Budget for Expansion (dimensionless)

equation: FPL.K=TABHL(TFPL,FPM.K,0,10,2)

FPL.K is a function of the "average community concern - police" at time K (FPM.K). The relationship is defined by the table function, TFPL.

FPM.K Average Community Concern - Police (dimensionless)

equation: FPM.K=FPM.J+(DT)(1/FPMD)(S1.J-FPM.J)

FPM.K equals its value J (FPM.J) plus the product of the computation time (DT), the inverse of the "average time for community concern - police" (FPMD), and the difference between "community concern - CJS" and its average value at J (FPM.J).

initial value equation: FPM=S1

"Initial average community concern - police" (FPM) equals the "initial community concern - CJS" (S1).

FPN.K Delayed Police Capital Budget ($/yr.)

equation: FPN.K=FPN.J+(DT)(1/FPND)(FPE.J-FPN.J)

FPN.K equals its value at J (FPN.J) plus the product of (DT), the inverse of the "delay for police capital budget" (FPND), and the difference of the "annual police capital budget" at J (FPE.J) and its delayed value at J (FPN.J).

initial value equation: FPN=0

"Initial delayed police capital budget" equals zero.

FPP.K Cost Per Officer Per Year ($/officer/yr.)

equation: FPP.K=(PEC)(FPPA)+(1)(FPQ.K)+(-FPPA)(FPQ.K)+(0)(0)

or FPP.K=(PEC)(FPPA)+(FPQ.K)(1-FPPA)

FPP.K equals the product of the "normal operating cost per officer" (PEC) and the "fraction of operation costs fixed" (FPPA) plus the product of the "variable cost per officer based on crime load" at time K (FPQ.K) and one minus the "fraction of operation costs fixed" (FPPA).

FPQ.K Variable Cost Per Officer Based on Crime Load ($/officer/yr.)

equation: FPQ.K=(PEC)(CCK.K)(P2IA)/((PNA)(CCCA)(1))

FPQ.K equals the "normal operating cost per officer per year" (PEC) multiplied by the ratio of the present crime load on police to the normal load: the present load is CCK.K - the "average crimes reported per officer per year" at time K; the normal load = (PNA)(CCCA)/(P2IA) or the product of the "initial active criminals" (PNA) and the "normal reported crimes per criminal per year" (CCCA) divided by the "initial police officers" (P2IA). Thus the present load and the normal load have the same units: crimes per officer per year.

Police Sector

P1.K Police Officers Being Hired (police)

equation: P1.K=P1.J+(DT)(PA.JK-PB.JK)

P1.K equals its value at time J (P1.J) plus the number of police ordered during the computation time (=(DT) times the "police order rate" between times J and K, PA.JK) minus the number of police who finish the hiring and training phase and become active policemen (=(DT) times the "new police arrival rate" between J and K, PB.JK).

initial value equation: P1=(PBAA)(PAF)

The "initial police officers being hired" equals the product of the "time to acquire and train officers" (PBAA) and "the initial average police attrition rate" (PAF). P1, then, compensates for the initial attrition rate.

P2.K Police Officers (police)

equation: P2.K=P2.J+(DT)(PB.JK-PC.JK)

P2.K equals its value at time J plus the number of police who have become active (=(DT) times the "new police arrival rate" between times J and K, PB.JK) minus the number of police who have left the force during DT (=(DT) times the "police attrition rate" between J and K, PC.JK).

initial value equation: P2=P2IA

"Initial police officers" equals a constant, P2IA - the number of officers in 1960.

PA.KL Police Order Rate (police/yr.)

equation: PA.KL=MAX(PAA.K,PAG.K)

PA.KL equals the larger of the "police order limit function" at time K (PAA.K) and the "police reduction maximum" at K (PAG.K).

PAA.K Police Order Limit Function (police/yr.)

equation: PAA.K=MIN(PAC.K,PAH.K)

PAA.K equals the smaller of the "police order rate needed" at K (PAC.K) and the "police order maximum" at K (PAH.K).

PAB.K Police Necessary to Maintain Desired Crime - Clearance (police)

equation: PAB.K=(PAE.K)(PABA)/PAJ.K

$$\text{or PAB.K} = \frac{\text{(PAE.K)(PABA)}}{\text{PAJ.K}} \left(\frac{\text{persons arrested}}{\text{crimes cleared}}\right)$$

PAB.K equals the product of the "average crimes reported per year" at time K (PAE.K) and the "fraction of crimes desired.

PAC.K Police Order Rate Needed (police/yr.)

equation: PAC.K=PAF.K+(1/PACA)(PAD.K-P2.K-P1.K+Ø+Ø+Ø)

PAC.K equals the "average police attrition rate" at K (PAF.K) plus the inverse of the "adjustment time for police officers" (PACA) multiplied by the sum of the "police needed" at K (PAD.K) minus the active "police officers" at K (P2.K) minus the "police officers being hired" at K (P1.K).

PAD.K Police Needed (police)

equation: PAD.K=PAB.K+(PBAA)(PAF.K)

PAD.K equals the "police needed to maintain desired crime clearance" at K (PAB.K) plus the product of the "time to acquire and train police officers" (PBAA) and the "average police attrition rate" at K (PAF.K). So PAD.K contains two factors: the first is related to crime rate; the second is a replacement factor.

PAE.K Average Crimes Reported Per Year (crimes/yr.)

equation: PAE.K=PAE.J+(DT)(1/PAEA)(CCA.J-PAE.J)

PAE.K equals its value at time J (PAE.J) plus the product of the computation time (DT), the inverse of the "average time for reported crimes" (PAEA), and the difference between the "reported crimes per year" at time J (CCA.J) and its average value at time J (PAE.J).

initial value equation: PAE=(CCCA)(PNA)

"Initial average reported crimes per year" (PAE) equals the product of the "normal reported crimes per criminal per year" (CCCA) and the "initial active criminals" (PNA).

PAF.K Average Police Attrition Rate (police/yr.)

equation: PAF.K=PAF.J+(DT)(1/PAFA)(PC.JK-PAF.J)

PAF.K equals its value at time J plus the product of the computation time (DT), the inverse of the "average time for police attrition (PAFA), and the difference between the "police attrition rate" between J and K (PC.JK) and its average value at J (PAF.J).

initial value equation: PAF=P2/PCAA

The "initial average police attrition rate" equals the initial number of "police officers" (P2) divided by the "average time on the police force" (PCAA).

PAG.K Police Force Reduction Maximum (police/yr.)

equation: PAG.K=-P2.K/PAGA

PAG.K equals the negative of the ratio of "police officers" at K (P2.K) to the "shortest time for police reduction" (PAGA). PAG.K is the maximum rate at which the police force would be allowed the decrease in size if the decrease became policy.

PAH.K Police Order Maximum (police/yr.)

equation: PAH.K=(1/PEB)(FPC.K+FPN.K)

PAH.K equals the sum of the "annual police replacement budget" at K (FPC.K) and the "delayed police capital budget" at K (FPN.K) divided by the "normal acquisition cost per officer" (PEB). PAH.K is a financial upper bound.

PAJ.K Persons Arrested Per Officer (persons/officer/yr.)

equation: PAJ.K=PAJ.J+(DT)((1/P2.J)(CC.JK+0)+(1/1)(-PAJ.J+0))

or PAJ.K=PAJ.J+(DT)(1/1 year)((1/P2.J)(CC.JK)+(1/1)(-PAJ.J))

PAJ.K equals its value at time J (PAJ.J) plus the product of the computation time (DT), the inverse of the average time for persons arrested per officer (1 year), and a third term: it equals the "arrest rate" between times J and K (CC.JK) divided by the number of "police officers" at J (P2.J) plus the negative of the average value at J, PAJ.J.

initial value equation: PAJ=CCAB

The "initial persons arrested per officer" (PAJ) equals a constant, CCAB.

PB.KL New Police Arrival Rate (police/yr.)

equation: PB.KL=P1.K/PBAA

PB.KL equals the number of "police officers being hired" at time K (P1.K) divided by the "time to acquire and train officers" (PBAA).

PC.KL Police Attrition Rate (police/yr.)

equation: PC.KL=P2.K/PCAA

PC.KL equals of the ratio of the number of "police officers" at time K (P2.K) to the "average time on the police force" (PCAA).

Community Sector

S1.K Community Concern - CJS (dimensionless)

equation: S1.K=S1.J+(DT)(SA.JK)

S1.K equals its value at time J (S1.J) plus the product of the computation time (DT) and the "rate of change in community concern" between times J and K (SA.JK).

initial value equation: S1=S1XJ

"Initial community concern" (S1) equals a constant, S1XJ.

SA.KL Rate of Change in Community Concern (dimensionless)

equation: SA.KL=(1/SAXD)(SAA.K-S1.K)

SA.KL equals the difference between the "percent change indicated in community concern" at time K (SAA.K) and the "community concern - CJS" at K (S1.K) divided by the "delay in public concern development" (SAXD).

SAA.K Percent Change Indicated in Community Concern (dimensionless)

equation: SAA.K=TABHL(SAAT,SAB.K,Ø,2Ø,4)

SAA.K is a function of "delayed concern over crime rate" at time K (SAB.K). The relationship is defined by the table function, SAAT.

SAB.K Delayed Concern Over Crime Rate (dimensionless)

equation: SAB.K=SAB.J+(DT)(1/SABD)(SAC.J-SAB.J)

SAB.K equals its value at time J (SAB.J) plus the product of the computation time (DT), the inverse of the "delay time for crimes per person" (SABD), and the difference between the value for "crimes per noncriminal per year" at time J (SAC.J) and the value of "delayed concern over crime rate" at J (SAB.J).

initial value equation: SAB=1

"Initial delayed concern over crime rate" equals a constant, 1.

SAC.K Perceived Severity of Crime Problem (dimensionless)

equation: SAC.K=(CCA.K)(C1JB)(1)/((C1.K)(PNA)(CCCA))

$$\text{or SAC.K}=\left(\frac{\text{CCA.K}}{\text{C1.K}}\right)\Bigg/\left(\frac{\text{(PNA)(CCCA)}}{\text{C1JB}}\right)$$

SAC.K equals the ratio of crimes reported per noncriminal per year at time K to its initial value. Crimes reported per noncriminal per year at K is obtained by dividing "reported crimes per year" at K (CCA.K) by "noncriminals" at K (C1.K). The initial value is obtained by multiplying the "initial active criminals" (PNA) by the "normal reported crimes per criminal per year" (CCCA) - yielding initial crimes per year - and dividing by the "initial noncriminal population" (C1JB).

Supplementary System Evaluation Variables

X1.K Total Population (persons)

equation: X1.K=C1.K+C2.K+C3.K+C4.K+C5.K+C6.K

X1.K equals the sum of the "non-criminal population" at K (C1.K), the "potential active criminals" at K (C2.K), the "persons in CJS" at K (C3.K), the "persons in prison" at K (C4.K), the "persons on parole" at K (C5.K), and the "persons on probation" at K (C6.K).

X2.K Crime Rate (Per 100,000) (crimes/person/yr.)

equation: X2.K=(1ØØØØØ)(CCA.K)/X1.K or $(CCA.K) \Big/ \frac{(X1.K)}{1ØØØØØ}$

X2.K equals the "reported crimes per year" at K (CCA.K) divided by the ratio of the "total population" at K (X1.K) to (1ØØØØØ).

X4.K Fraction Reported Crimes Cleared by Arrest (dimensionless)

equation: X4.K=(BAFB)(BAF.K)/PAE.K

X4.K equals the "average arrest rate" at time K (BAF.K) times the "crimes cleared per person arrested" (BAFB) divided by the "average crimes reported per year" at time K (PAE.K).

X5.K Fraction Reported Crimes Punished (dimensionless)

equation: X5.K=(CFLB)(CFL.K)/PAE.K

X5.K equals the "average rate sent to prison" at time K (CFL.K) times the "crimes punished per person sent to prison (CFLB) divided by the "average crimes reported per year" at K (PAE.K).

X6.K Criminals at Large Per Person Imprisoned (criminals/person)

equation: X6.K=CCB.K/C4.K

X6.K equals the "criminals at large" at K (CCB.K) divided by the number of "people in prison" at K (C4.K).

X8.K Arrests Per Officer Per Year (persons/officer/yr.)

equation: X8.K=CC.JK/P2.K

X8.K equals the "arrest rate" between times J and K (CC.JK) divided by the number of "police officers" at time K (P2.K).

List of Constants

Symbol	Value	Units	Meaning
AAAA	15	years	Estimated minimum time to dispose of unneeded capacity.
AACA	6	years	Estimated correction time for a capacity deficiency.
AADA	1.1	dimensionless	Estimated prison capacity needed per prisoner (safety factor) used in capacity planning.
AAFD	2	years	Estimated time to average capacity loss for replacement decision.
ABXA	2	years	Estimated time after decision to acquire new prison capacity.
ACXA	15	years	Average useful lifetime of prison capacity.
BAAA	15	years	Estimated minimum time to release unneeded prosecutors.
BACA	2	years	Estimated correction time for a prosecutor deficiency.
BAFB	1	crimes/person	Average crimes cleared per person arrested. Estimated by police officials.
BAFD	1	years	Estimated time to average arrest rate for prosecutor hiring decision.
BAGD	3	years	Averaging time for prosecutor attrition rate.
BAMA	1.5	years	Estimated desired time to correct excess case backlog.
BBXA	2	years	Estimated hiring and training delay for prosecutors.
BCXA	10	years	Estimated average time prosecutors remain with court.
CAAA	1	years	Estimated delay in population reproduction.

Symbol	Value	Units	Meaning
CABA	.01	1/years	Fraction population increase per year from census.
CBAA	.003	1/years	Normal annual fraction of non-crime pop. flowing to crime. Estimated by various CJS personnel and model tuning.
CBCA	1	year	Estimated time for peer contact to result in flow to crime.
CBEA	1	year	Estimated time for crime reward awareness to influence flow to crime.
CBEB	400	dollars/crime	Estimated average dollar amount realized per crime.
CBGA	2	years	Estimated time for police effectiveness to influence flow to crime.
CBLA	8	years	Estimated time before education expenditures influence flow to crime.
CBQA	5	dimensionless	Reciprocal of the estimated fraction of total crimes reported.
CCAB	3.3	persons/officer/year	Normal police arresting effectiveness estimated from Atlanta arrest data, 1960.
CCBA	.8	dimensionless	Estimated fraction of persons with open cases who are free on bond.
CCCA	2	crimes/criminal/year	Average annual reported crimes committed per criminal estimated by various police officials.
CCDA	5	crimes/year	Estimated normal (1960) aggressiveness of criminals on 0 to 10 scale.
CCEA	1	year	Estimated time for awareness of fraction sent to prison to influence flow to crime.
CCFA	1	year	Estimated time for police per criminal to affect aggressiveness.
CCKA	1	year	Estimated time for police load to affect arrest effectiveness.
CCNA	1	year	Estimated time for criminals aggressiveness to affect arrest effectiveness.
CCQA	1	year	Estimated time for case load to affect arrest effectiveness.
CCSA	5	years	Estimated time for community concern to affect arrest effectiveness.

Symbol	Value	Units	Meaning
CCZD	2	years	Estimated time for budget per officer to affect arrest effectiveness.
CDBA	275	persons/prosecutor/ year	Normal effectiveness of prosecutors estimated from 1960 data.
CEAA	.35	dimensionless	Normal fraction of people released who reform. Estimates varied widely.
CEBA	2	years	Estimated time for case load to affect the fraction of cases tried.
CFBA	.5	dimensionless	Estimated normal fraction of trial cases receiving a prison sentence.
CFCA	.2	dimensionless	Estimated normal fraction of cases that go to trial.
CFEA	.1	years	Estimated average delay for nontrial cases (1960).
CFFA	.25	years	Estimated average delay for trial cases (1960).
CFGA	1	years	Estimated time for case disposition time to affect plea fraction.
CFJA	.1	dimensionless	Estimated normal fraction of total disposition to prison by plea.
CFLB	1.5	crimes/person	Estimated crimes punished per person sent to prison.
CFLD	2	years	Estimated time to average flow to prison.
CFMD	2	years	Estimated time to average court load.
CGA	40	years	Average inmate lifetime in prison.
CHBA	5	years	Estimated time for average sentence length to affect prison release rate.
CHCA	4	years	Estimated average sentence time, 1960.
CHDA	1	year	Estimated time the load on prison capacity to affect fraction sentenced to prison.
CJAA	.05	dimensionless	Normal fraction of people who reform after prison. Estimates varied widely.
CJCA	3	years	Estimated time for load on person capa- city to affect fraction reform after prison.

Symbol	Value	Units	Meaning
CKAA	.9	dimensionless	Average fraction of people released on parole estimated by court officials and release data.
CLAA	.9	dimensionless	Estimated fraction of people who repeat criminal activity after parole. Estimates varied widely.
CLAD	3	years	Estimated average time on parole.
CNAA	.3	dimensionless	Estimated average fraction of people released on probation.
CQAA	.8	dimensionless	Estimated average fraction of people who return to criminal activity after probation. Estimates varied widely.
CQAD	1	year	Estimated average time on probation.
FABA	300	\$/capacity units/year	Estimated annual operating cost per prison capacity unit.
FABC	500	\$/person/year	Estimated average annual cost per prisoner.
FABE	50	\$/person/year	Estimated annual cost per parolee.
FACA	2000	\$/capacity unit	Estimated average price per unit of prison capacity.
FAEA	.2	1/years	Estimated maximum prison expansion fraction - 20% per year.
FAHD	2	years	Estimated time for community concern to influence prison budget.
FAJD	.5	years	Estimated time for corrections capital budget to affect purchasing.
FAKA	0	\$/year	Test input for annual external support for corrections.
FBBA	250,000	\$/prosecutor/year	Estimated annual operating cost per prosecutor.
FBBC	150	\$/person/year	Average annual cost per probationer estimated by court officials.
FBCA	200,000	\$/prosecutor	Average cost of hiring a new prosecutor including all associated support and facilities.
FBEA	.2	1/years	Estimated maximum annual courts expansion fraction - 20% of prosecutors.

Symbol	Value	Units	Meaning
FBHD	2	years	Estimated time for community concern to affect courts budget.
FBJD	.5	years	Estimated time for courts capital budget to affect hiring and purchasing.
FBKA	0	$/year	Test input of annual external support for courts.
FECD	3	years	Estimated time for community concern to affect education budget.
FPDA	0	$/year	Test input of annual external support for police.
FPHA	.2	1/year	Estimated maximum police expansion fraction - 20% of officers.
FPMD	2	years	Estimated time for community concern to affect police budget.
FPND	.5	years	Estimated time for police capital budget to influence hiring and purchasing.
FPPA	.3	dimensionless	Estimated fraction of police operating costs not dependent on number of officers.
PABA	.25	dimensionless	Fraction of crimes reported desired cleared by arrest.
PACA	6	years	Estimated time to correct deficiency in police officers.
PAEA	2	years	Estimated time to average reported crimes for police hiring decision.
PAFA	2	years	Estimated time to average police attrition for hiring decision.
PAGA	15	years	Estimated minimum time for release of unneeded police resources.
PBAA	1	year	Estimated time to acquire and train officers.
PCAA	10	years	Estimated average tenure on police force.
PEB	5000	$/officer	Estimated average cost to acquire, train, equip and support a police officer.
PEC	5000	$/officer/year	Estimated normal annual operating cost per officer.
SAXD	2	years	Estimated time for community concern to respond to crime rate changes.

Symbol	Value	Units	Meaning
SABD	2	years	Estimated time for public perception of changes in crime rate.
TCBB	.85/1/ 1/1/ 1.1/1.3	dimensionless	Values for peer pressure influence (CBB) on flow to crime.
TCBD	.9/1/ 1.05/ 1.2/ 1.4/ 1.7	dimensionless	Values for crime reward influence (CBD) on flow to crime.
TCBF	1.15/ 1.05/1/ .95/.9/ .85	dimensionless	Values for police effectiveness influence (CBF) on flow to crime.
TCBH	1.3/ 1.2/ 1.1/1/ 1/.95	dimensionless	Values for fraction sent to prison influence (CBH) on flow to crime.
TCBJ	1/1/ .95/.9/ .82/.7	dimensionless	Values for delayed education budget influence (CBJ) on flow to crime.
TCBS	1/1.05/ 1.15/ 1.3/ 1.35/ 1.25/ 1.1/1	dimensionless	Values for population age distribution influence (CBS) on flow to crime.
TCDB	2.5/2/ 1.7/ 1.45/ 1.2/1/ .9/.7	dimensionless	Values for court productivity as influenced by current load. Values for case load influence (CXD) on prosecutor effectiveness.
TCCD	1.5/ 1.3/ 1.1/1/ 1/.8	dimensionless	Values for fraction sent to prison influence (CXG) on criminals' aggressiveness.
TCCC	.75/.9/ 1/1/ 1.3/ 1.6	dimensionless	Values for criminals' aggressiveness influence (CXJ) on crimes per criminal per year.
TCCF	1/1.05/ 1.2/ 1.4/ 1.6	dimensionless	Values for delayed case load influence on criminals' aggressiveness.

Symbol	Value	Units	Meaning
TCCJ	1/1.2/ 1.4/ 1.5/ 1.7/ 1.8	dimensionless	Values for police load influence (CCJ) on arrest effectiveness.
TCCM	1.2/ 1.05/ 1/1/.9/ .7	dimensionless	Values for criminals' aggressiveness influence (CCM) on arrest effectiveness.
TCCP	1/.95/ .9/.8/ .7	dimensionless	Values for case load influence (CCP) on arrest effectiveness.
TCCR	.7/.95/ 1/1.05/ 1.2/1.4	dimensionless	Values for community concern influence (CCR) on arrest effectiveness.
TCCX	1/1.05/ 1.15/ 1.2/ 1.25/ 1.3	dimensionless	Values for budget per officer influence on arrest effectiveness.
TCEA	1/1/1/ 1/1/1	dimensionless	Values for case load influence (CXR) on reform fraction after CJS.
TCFB	1.2/ 1.1/1/ 1/.9	dimensionless	Values for prison load influence (CXS) on fraction of trial cases sent to prison.
TCHC	1.2/ 1.05/1/ .95/.8	dimensionless	Values for prison load influence (CXT) on average sentence time.
TCFC	1/1/ .95/ .85/ .75/.6	dimensionless	Values for case load influence (CFN) on fraction of cases tried.
TCFJ	1/1/ .95/.9/ .85/.7	dimensionless	Values for case load influence (CFJ) plea fraction.
TCJB	1.1/1/ 1/.95/ .9	dimensionless	Values for prison load influence (CJB) on reform fraction after prison.
FAGT	.015/ .03/ .06/ .12/ .25/.5	dimensionless	Values for community concern influence (FAG) on corrections capital budget.

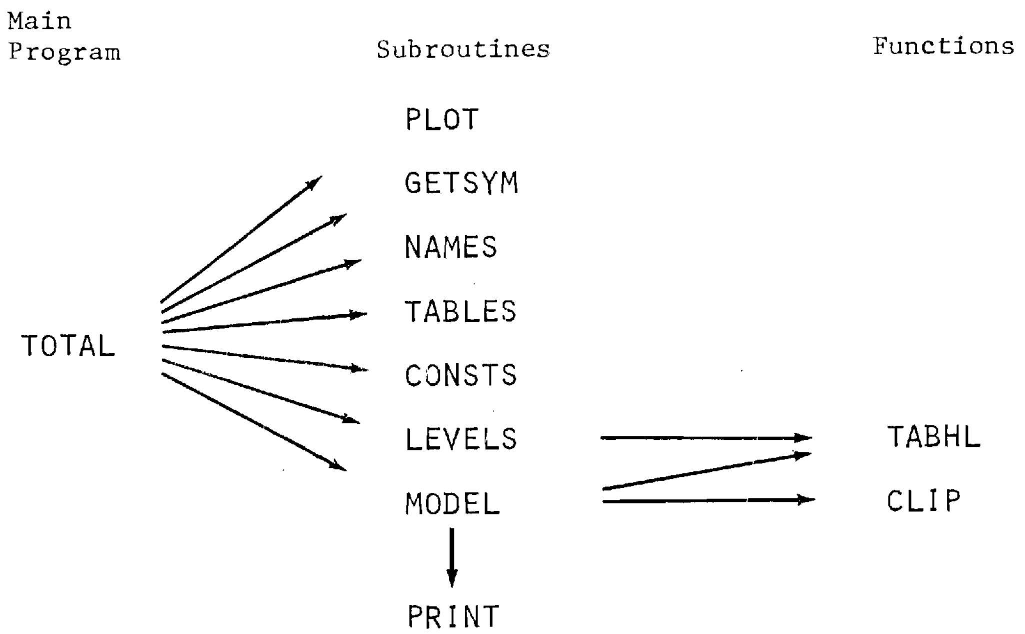
Symbol	Value	Units	Meaning
FBGT	.04/ .08/ .12/ 125/.4/ .6	dimensionless	Values for community concern influence (FBG) on courts capital budget.
TFEB	0/.015/ .03/ .05/ .075/ .125	dimensionless	Values for community concern influence (FEB) on crime education budget.
TFPL	.02/ .05/.1/ .2/.4/ .7	dimensionless	Values for community concern influence (FPL) on police capital budget.
SAAT	4/6/ 7.5/ 8.7/ 9.5/10	1/years	Values for crime rate influence (SAA) on changes in community concern.

III. THE FORTRAN PROGRAM

The Criminal Justice System Training Model exists in the form of a program written in Univac FORTRAN V programming language. On any Univac 1100-series computer system (or, with modifications, on any large computer system), the program can be run interactively through any remote computer terminal that provides hard-copy 80-character output.

A. Program Documentation

The FORTRAN program consists of a main program, eight subroutines and two function routines, related as follows (A→B means that routine A calls routine B):



TOTAL is the main program; it controls the execution of the simulation.

PLOT is used to obtain a graphical representation of the variables of interest, plotted against time.

GETSYM interprets the instructions given by the user.

NAMES contains a list of the definitions of the plot variables and the variable names of all constants and levels.

TABLES defines the numerical values of the table functions used in the model.

CONSTS defines the numerical values of the constants used in the model.

LEVELS defines the initial values of the levels used in the model.

MODEL performs the actual simulation of the model.

PRINT is used is used to print the computed values of the rates, levels and auxiliaries.

TABHL finds a numerical value interpolating from the tables.

CLIP compares two numbers, returning an indicator that identifies which is larger.

In addition to TABHL and CLIP, several standard FORTRAN library functions are called by several of the subroutines.

Instructions

There are 13 interactive instructions: GO, STOP, SET, STORE ON, STORE OFF, LIST, TIMES, PAUSE, PRINT, TABLE, PLOT, SUMMARY PLOT and SCALE. The user causes the program to perform simulations and make reports by typing in these commands and their modifying arguments. The syntax and effect of each command is described in detail in section IV, the Trainer's Manual. Briefly, GO causes a simulation run, STOP ends a series of runs, SET allows the uses to change the value of a constant in the model, STORE ON causes storage of results of runs for later comparison, STORE OFF causes discarding of unwanted previously stored results, LIST and TIMES cause chosen variables to be presented at given time intervals whether or not the chosen variables are plotted, PAUSE and TIMES cause the output to stop temporarily, PRINT causes values of constants to be displayed, TABLE enables the user to change functional relationships in the model that are stored as a table rather than as equations, PLOT causes a graphical display approximate time history of up to five variables to be output, SUMMARY PLOT enables the user to compare the results of previews runs on a single graph, and SCALE enables the user to control reference marks on the plot.

Model, Input and Output

The model itself (as opposed to provisions for interaction between the program and the user) is contained mainly in subroutines MODELS, TABLES, CONSTANTS and LEVELS. MODEL contains equations that describe how each variable depends on (a) constants, (b) values of other variables and (c) its own past values. TABLES contains relationships between variables and constants that are stored as a pair list rather than as an equation. CONSTANTS contains values of parameters (constants) of the model. LEVELS contains the initial value of certain variables called levels.

Subroutine GETSYM is concerned with input only and contains no part of the model.

Subroutine PLOT is concerned with output only and contains no part of the model.

Except for dimensions, data statements and common statements, subroutine PRINT does not contain any parts of the model. It simply stores results for plotting as printing.

Model Building

The model is a Forrester-type deterministis simulation model that is intended to represent the relationships among various parts of a criminal justice system ones time. It is a typical model of the kind often programmed in DYNAMO language, but the interactive FORTAN implementation allows much more convenient user interventions and changes than would be possible in a DYNAMO implementation.

The user may make certain basic changes in the model without programming. Those functional relationships that are expressed as tables may be changed interactively via the instruction TABLE. Parameters may be altered interactively via the instruction SET. But in order to add variables or to change functional relationships that are expressed as equations, the program itself must be altered.

There are two levels of program alteration (model building): minor alteration and major alteration.

Minor Alteration: When the model is to be modified by changing the number of level, auxiliary or rate equations, or the number of tables, or when the model builder wants to make a minor change in the output specifications, the programmer should proceed according to the following MODEL BUILDER'S GUIDE. Statement numbers in this guide refer to the numbers printed on the left of the full program listing.

Major Alteration: When the model is to be altered so extensively that it is more convenient to start from scratch than to revise the existing model, the programmer may find it useful to refer to the skeleton program listing, which contains those statements that are independent of the model. Computer card decks of both the full and the skeleton program are available from the authors. The numbers on the right of the skeleton listing correspond to the numbers on the left of the full listing as shown by the full listing, which contains both numbers for each statement common to both listings.

One should not attempt major alteration (as defined here) without previous experience in programming Forrestorian models in DYNAMO, or withour previous FORTRAN experience. However, the MODEL BUILDER'S GUIDE is sufficient for use by a non-FORTRAN programming to do minor alterations to the model.

Generality of the FORTRAN Program

Interactive programs cannot be written in ANSI standard FORTRAN (FORTRAN IV). Certain, but not all, features of the input and output portions of the program are compatible with the advanced FORTRAN available on large computers other than the Univac 1108. If the program is to be implemented or any computer other than a Univac U-1108, transletion of many FORTRAN statements will be necessary (this is a commonly encountered task for FORTRAN programmers).

B. Model Builder's Guide

Introduction

Every Forrestorian simulation contains various types of equations. A level equation is a first-order difference equation that calculates the value of a variable called a level; each level is equal to the old value of the same level plus a term that includes the time increment as a multiplicative factor. An auxiliary equation is an algebraic or logic equation based on levels and/or other auxiliary variables, defining an auxiliary variable. A supplementary equation defines a supplementary variable, which is similar to an auxiliary variable except that it is used for output only and is never used to calculate any other kind of variable, i.e., a supplemetary variable never appears on the right-hand side of any equation. A rate equation defines a rate variable, which expresses the influence of one level variable on another. A table is a set of pairs of values that defines one variable in terms of another by means of a table look-up (with linear interpolation) rather than by means of an equation. A plot variable is any variable chosen by the programmer to be able to be displayed to the user on a graphical time plot. A constant is any parameter whose value will not be allowed to change with time during a single simulation. This nomenclature has been chosen to coincide with that of the Forrestorian simulation language DYNAMO, since the previous existence of a DYNAMO-type model is assumed.

The following is a list of the step-by-step procedures required to make a minor alteration to the existing criminal justice training model. We will separately consider (A) changes in the model itself and (B) changes in the output characteristics. The following minor changes are contemplated:

(A) Model changes

1. Add (delete) a level equation
2. Add (delete) an auxiliary equation
3. Add (delete) a rate equation
4. Add (delete) a constant
5. Add (delete) a table

(B) Output changes

6. Plot a variable not currently plotted
7. Change the plot specifications of the plot obtained when the user does not intervene
8. Change the simulation specifications

The statements that need to be updated are referred to in this Model Builder's Guide by the subprogram name in which they appear and by the line number within the subprogram. These numbers appear in the leftmost column of the full program listing.

(A) MODEL CHANGES

1. To add (delete) a level equation, the model builder must update:

1.1 Number of levels. This parameter appears in the following places:

1.1.1 TOTAL:L4. The number in parenthesis in PNAMES(xxx) is the sum of the number of rate, level and auxiliary variables.

1.1.2 TOTAL:L9. The number in parenthesis in ILEV(xxx) is the number of levels.

1.1.3 TOTAL:L15. NOLEVS is the number of levels.

1.1.4 TOTAL:L25. The numbers in parenthesis in IVALS(xxx) and ILEVS(xxx) are the number of levels.

1.1.5 TOTAL:L78. The number of iterations in the DO statement is the number of levels.

1.1.6 TOTAL:L139. The number of iterations in the DO statement is the number of levels.

1.1.7 PRINT:L2. The first number in parenthesis in F(xxx,20) is the number of levels.

1.1.8 PRINT:L3. The number in parenthesis in IF(xxx) is the number of levels.

1.1.9 PRINT:L6. The number in parenthesis is the sum of the number of rate, level and auxiliary variables.

1.1.10 PRINT:L255. The number of iterations in the DO statement is the number of levels.

1.1.11 NAMES:L4. The number in parenthesis in ILEVELS(xx) is the number of levels.

1.1.12 NAMES:L5. The number in parenthesis is the number of levels.

1.1.13 NAMES:L62. The number of iterations in the DO statement is the number of levels.

1.2 Array of level names. This array is a list of the names of all the levels, appearing in the following places:

1.2.1 PRINT:L20-L25. The array in this COMMON statement is a list of the names of all levels, separated by commas. The list starts in column 7 and ends in column 72 (on an 80-column computer card). If the list is too long for one card (line), then it continues on the next card, which is identified as a continuation by having a non-zero character in column 6.

1.2.2 PRINT:L40-L45. The array in this DATA statement is a list of the names of all the levels, enclosed by single quation marks and separated by commas. If the new level is added at the end, the slash (/) must be restored at the end.

1.2.3 PRINT:L201-L246. In these lines, each level is given a serial number i in one of the statements F(i,J)=yyy, where yyy is the name of the level.

1.2.4 NAMES:L49-L53. The array in this DATA statement is a list of the names of all the levels, enclosed by single quatation marks and separated by commas, with continuations as indicated above and with an ending slash as mentioned in 1.2.2 above.

1.2.5 LEVELS:L9-L12. The array in this COMMON statement is a list of the names of all the levels.

1.2.6 MODEL:L16-L21. The array in this COMMON statement is a list of the names of all the levels.

1.2.7 MODEL:L49-L53. The array in this COMMON statement is a list of the names of all the levels, except that each name has a prefix I .

1.2.8 MODEL:L56-L60. The array in this REAL statement is a list of the names of all the levels, except that each name has a prefix I.

1.2.9. MODEL:L62-L107. A statement of the form NAME=INAME, where NAME is the name of a level and INAME is the I-prefixed name of the same level, is given for each level.

1.2.10 MODEL:L312-L317. A list of the names of all the levels, separated by commas, is given in these continuation cards (lines).

1.3 <u>List of all level equations</u>. This list appears in the following places:

1.3.1 LEVELS:L23-L80. This list contains the equations that define the initial values of each level. A given level should not appear on the right-hand side of a level equation unless it was defined (left-hand side of a level equation) in an earlier equation. Thus the order of level equations is important, and a new level equation must be put in the proper place.

1.3.2 MODEL:L342-L387. This list contains the equations that define the levels. Here again the order of definition is important, and the new equation must be placed so that the variables on the right-hand side have been previously defined.

<u>2. To add (delete) an auxiliary or supplementary equation, the model builder must update</u>:

2.1 <u>Number of auxiliary variables</u>. This parameter appears in the following places:

2.1.1 TOTAL:L4. The number in parenthesis in PNAMES(xxx) is the sum of the number of rate, level and auxiliary variables.

2.1.2 TOTAL:L9. The number in parenthesis in IAUX(xxx) is the number of auxiliary variables.

2.1.3 TOTAL:L80. The number of iterations in the DO statement is the number of auxiliary variables.

2.1.4 TOTAL:L135. The number of iterations in the DO statement is the number of auxiliary variables.

2.1.5 PRINT:L2. The number of auxiliary variables is xxx in A(xxx,20).

2.1.6 PRINT:L3. The number of auxiliary variables is xxx in IA(xxx).

2.1.7 PRINT:L249. The number of iterations in the DO statement is the number of auxiliary variables.

2.2 <u>Array of Auxiliary variable names</u>. This array appears in the following places:

2.2.1 PRINT:L9-L17.

2.2.2 PRINT:L25-L38.

2.2.3 PRINT:L50-L175. In these lines, each auxiliary variable is given a serial number in one of the statements A(i,J)=yyy, where i is the serial number and yyy is the name of the auxiliary variable.

2.2.4 MODEL:L5-L13. The array in this COMMON statement is a list of all the names of the auxiliary variables. It also contains a list of names of supplementary variables, except that each name has the prefix D .

2.2.5 MODEL:L301-L309.

2.3 <u>List of auxiliary equations</u>. This list appears in the following places:

2.3.1 MODEL:L115-L241. The list of level equations must be ordered so that the variables that appear on the right-hand side of an equation have been defined.

<u>3. To add (delete) a rate equation, the model builder must update</u>:

3.1 <u>Number of rate variables</u>. This parameter appears in the following places:

3.1.1 TOTAL:L4. The number in parenthesis in PNAMES(xxx) is the sum of the number of rate, level and auxiliary variables.

3.1.2 TOTAL:L9. The number in parenthesis in IRAT(xxx) is the number of rate variables.

3.1.3 TOTAL:L137. The number of iterations in the DO statement is the number of rate variables.

3.1.4 TOTAL:L140. The number in parenthesis in PNAMES(I+xxx) has xxx as the sum of the number of auxiliary variables plus the number of rate variables.

3.1.5 PRINT:L2. In R(xxx,20), xxx is the number of rate variables.

3.1.6 PRINT:L3. In IR(xxx), xxx is the number of rate variables.

3.1.7 PRINT:L6. In NAMES(xxz), xxz is the sum of the number of rate, level and auxiliary variables.

3.1.8 PRINT:L252. The number of iterations in the DO statement is the number of rate variables.

3.2 <u>Array of rate variable names</u>. This array appears in the following places:

3.2.1 PRINT:L18-L19. The array in this COMMON statement contains the names of all rate variables.

3.2.2 PRINT:L38-L40. The array in this DATA statement contains the names, in single quotes, of all rate variables.

3.2.3 PRINT:L176-L200. In these lines, each rate is given a serial number i in one of the statements R(i,J)=yyy , where yyy is the name of the rate.

3.2.4 MODEL:L14-L15. The array in this COMMON statement is a list of the names of all rate variables.

3.2.5 MODEL:L310-L311.

3.3 <u>List of rate equations</u>. This list appears in the following place:

3.3.1 MODEL:L243-L268. The list of rate equations must be ordered so that the variables that appear on the right-hand side of an equation have been previously defined.

4. <u>To add (delete) a constant, the model builder must update</u>:

4.1 <u>Number of constants</u>. This parameter appears in the following places:

4.1.1 TOTAL:L14. NOCONS is the number of constants.

4.1.2 TOTAL:L23. The numbers in parenthesis in Y(xxx) and CONST(xxx) are both the number of constants.

4.1.3 NAMES:L2. In CONST(xxx), xxx is the number of constants.

4.1.4 NAMES:L4. In ICONS(xxx), xxx is the number of constants.

4.1.5 NAMES:L60. The number of iterations in the DO statement is the number of constants.

4.2 <u>Array of Constant names</u>. This array appears in the following places:

4.2.1 NAMES:L35-L48. The array in this DATA statement is a list of the names of all the constants, enclosed in single quotes and separated by commas.

4.2.2 LEVELS:L13-L22.

4.2.3 MODEL:L39-L48.

4.2.4 CONSTS:L2-L11.

4.3 List of Constants. This list appears in the following place:

4.3.1 CONSTS:L12-L107. In these lines, the numerical value of each constant is defined in statements of the form nnn=uuu, where nnn is the name of the constant and uuu is its numerical value. The order of statements is immaterial.

5. To add (delete) a table, the model builder must update:

5.1 Number of tables. This parameter appears in the following places:

5.1.1 TOTAL:L18. The number in parenthesis in TNAME (xxx) is the number of tables.

5.1.2 TOTAL:L15g. The number of iterations in the DO statement is the number of tables.

5.1.3 TOTAL:L387. The number of iterations in the DO statement is the number of tables.

5.2 Array of table names. This array appears in the program in the following places:

5.2.1 TOTAL:L27-L30. The array in this DIMENSION statement is a list of all the tables with the number of values defined for each table in parenthesis, separated by a comma. The format is DIMENSION list, using continuation cards if necessary.

5.2.2 TOTAL:L31-L33. The array in this COMMON statement is a list of all the tables separated by a comma, punched as indicated before. The formate is COMMON/TABLE/list.

5.2.3 TOTAL:L162-L187. A set of statements, one for each table, with the format if (I.EQ.J) write (6,6498) INSYM, NAME, where J goes from one to the number of tables and NAME refers to the given name for each table in the order they appear in the COMMON statement, must be given here.

5.2.4 TOTAL:L392-L417. A set of statements, one for each table, with the format if (I.EQ.L) NAME(J)=XNUM, where L goes from one to the number of tables and NAME refers to the given name for each table in the order they appear in the COMMON statement, must be given here.

5.2.5 LEVELS:L2-L5. The array in this DIMENSION statement is a list of all the tables with the number of values defined for each table in parenthesis, separated by commas. The format is DIMENSION list, using continuation cards if necessary.

5.2.6 LEVELS:L6-L8. The array in this COMMON statement is a list of all the tables separated by a comma, punched as indicated before. The format is COMMON/TABLE/LIST.

5.2.7 MODEL:L23-L26. The array in this DIMENSION statement is a list of all the tables with the number of values defined for each table in parenthesis, separated by a comma. The format is DIMENSION list, using continuation cards if necessary.

5.2.8 MODEL:L27-L29. The array in this COMMON statment is a list of all the tables separated by a comma, punched as indicated before. The format is COMMON/TABLE/LIST.

5.2.9 TABLES:L2-L5. The array in this DIMENSION statement is a list of all the tables with the number of values defined for each table in parenthesis, separated by commas. The format is DIMENSION list, using continuation cards if necessary.

5.2.10 TABLES:L6-L9. The array in this DIMENSION statement is a list of all the tables with the number of values defined for each table in parenthesis. The name of the tables in this array should be changed so that it starts with a D instead of T as in the previous dimension statement.

5.2.11 TABLES:L12-L14. The array in this COMMON statement contains the names of all the "T-Tables."

5.2.12 TABLES:L15-L18. The array in this DATA statemnt contains the names, in single quotes, of all the "D-Tables."

5.2.13 TABLES:L19-L44. The array in this data statement contains the values chosen for each "D-Table." The format is /xxx,yyy,zzz/ where xxx, yyy and zzz are the values which establish the functional relationship.

5.3 Storage of table values.

5.3.1 TABLES:L45-L78. The numerical values for the "T-Tables" are defined in a series of iterations for each subset of tables with the same number of values.

(B) OUTPUT CHANGES

6. To PLOT an additional variable, the model builder must update:

6.1 Number of plot variables. This parameter appears in the following places:

6.1.1 TOTAL:L16. NOPLTS is the number of variables that can be plotted.

6.1.2 TOTAL:L22. The number in parenthesis in NAME (6,xxx) must be set equal to the number of variables to be plotted.

6.1.3 NAMES:L2. The number in parenthesis in NAME (6,xxx) must be set equal to the number of variables to be plotted.

6.1.4 NAMES:L3. Since the DATA statement may have at most 18 continuation cards, the array of variables to be plotted must be stored in more than one array if more than 18 variables are to be plotted. To plot between 19 and 36 variables, the number in parenthesis in NAME 2 (6,xxx) must be set equal to the number of variables to be plotted minus 18, so that the complete array is kept in NAME 1 and NAME 2. If more than 36 variables are to be plotted, a third array NAME 3 would be used proceeding as before.

6.1.5 NAMES:L55 and L57. The number of iterations in these DO statements must be set up in the same way as the DATA statements. The purpose of this procedure is to store all the plot variables in a single array called NAME.

6.1.6 MODEL:L22. The number in parenthesis in X(xxx) must be set up equal to the number of variables to be plotted.

6.2 Array of plot variable names. This array appears in the following places:

6.2.1 TOTAL:L41-L45. The array in this DATA statement is a list of the scale specifications for the five variables to be plotted without the user intervention. The format is DATA VLIST/xxx.,o.,o.,o.,aaa., followed by four continuation cards with the format +yyy.,o.,o.,o.,bbb., where xxx. and yyy. represent a number in the array NAME corresponding to the variables to be plotted, and aaa. and bbb., are the desired upper limits of the plot variables.

6.2.2 NAMES:L8-L25 and L27-L34. The arrays in these DATA statements are a list of the variables to be plotted with their meaning. The format of this list is + 'NAME', "MEANING', where NAME stands for the variable name, and MEANING refers to its significance in the model. The format of the data statement is DATA NAME 1/list/,DATA NAME 2/list/.

6.2.3 MODEL:L32-L38. The array in this EQUIVALENCE statement is a list of the elements of the vector X and the plot variables that are to be in a one to one correspondence. The format is EQUIVALENCE list, where the elements of this list are ordered pairs where the first component is X(i), and the second one of the plot variables, enclosed in parenthesis and separated by a comma. The order of the plot variables in this list must be the same as in the array NAME defined previously.

6.2.4 MODEL:L270-L295. In this set of statements a variable of the format DNAME is defined equal to NAME where NAME refers to the given name for the variable to be plotted, for each plot variable.

7. To change the plot specifications of the plot obtained when the user does not intervene, the model builder must update:

7.1 Scale of the time plot. This parameter appears in the following place:

7.1.1 TOTAL:L41-L45. The array in this DATA statement is a list of the scale specifications for the five variables to be plotted. The format is /xxx.,o., o., o.,aaa.,/ where xxx. is a number that indicates which of the variables in the array NAME is to be plotted and aaa. is the upper limit of the scale.

7.2 Years to be plotted. This parameter appears in the following place:

7.2.1 TOTAL:L53. PLTPD is the plot period.

8. To change the simulation specifications the model builder must update:

8.1 Time increment. This parameter is currently .25 years appearing in the following place:

8.1.1 TOTAL:L55. DT is the time increment.

8.2 Length of the simulation run. This parameter is currently 20 years appearing in the following place:

8.2.1 TOTAL:L53. NOYRS is the number of years.

8.3 Initiation of the simulation. This parameter is currently 1960 appearing in the following place:

8.3.1 TOTAL:L53. TIMEO is the year in which the simulation starts.

C. Complete Program Listing Including Model

MAIN PROGRAM TOTAL

```
      DIMENSION ISMX(5)
      DIMENSION ULIST(5,5)
      LSTORE=0
      DIMENSION LINE(60)
      INTEGER PNAMES
      COMMON /PNAMS/ PNAMES
      DATA ILIST/-LIST-/,JTIME/-TIMES-/
      COMMON /PTVRS/ IAUX,IRAT,ILEV
      DIMENSION ITIMES(20)
      COMMON /TIMES/ ITIMES,KTIME
C----------- -
C----------- -
      INTEGER TNAME
      COMMON /TNAME/ TNAME
      DATA ITABLE/-TABLE-/
      DIMENSION VLIST(5,5),PAUSE(6)
      DIMENSION SZ(10,254,5),IVAR(10,5),KRUN(5)
      COMMON /BLOCK1/ INSYM,ITYPE,XNUM,IDEL,NEXTO,NEXT
      COMMON /Y/ Y
      COMMON /I/ IVALS
      COMMON /BUG/ JDEBUG,KDEBUG
      REAL IVALS
      INTEGER YES,CONST,SYC,SPACES,HALF,ONE,DOT
      INTEGER PAUSE,STAR,PLUS,BLANK
      DATA ISM/-C-,-)-,-I-,-L-,-P-/
      DATA NO/-NO-/,DOT/-.-/,BLANK/- -/,ISPACE/-SPACE-/
      DATA IAPPND/-A-/
      DATA ISCALE/-SCALE-/,IHALF/-HALF-/,IONE/-ONE-/,SPACES*10/
      DATA ISET/-SET-/,ISTART/-START-/,IPLTPD/-PLTPD-/,NYRS,-NOYRS-/
```

```
      DATA IPA/-[illegible]-/,I[illegible]/-[illegible]-/,IA[illegible]/-[illegible]-/,I[illegible]/-[illegible]-/
      DATA I[illegible]/-[illegible]-/,ISTO[illegible]/-STO[illegible]-/,ION/-ON-/,IOFF/-OFF-/
      DATA TI[illegible]/1[illegible]./,PLTPD/.5/,[illegible]/25/
      DATA STAR/-*-/,[illegible]/-1-/,IPLOT/-PLOT-/,IG[illegible]/-GO-/
      DATA HALF/-.-/,[illegible]/-.-/,I[illegible]T/-[illegible]-/,JT/.25/,IPAUSE/-PAUSE-/
      DATA PAUSE/9[illegible]99,[illegible]/
      DATA IVARBL/-VARIA-/,ICONS/-CONST-/,IPRINT/-PRINT-/
      DATA YES/-YES-/,IRESET/-RESET-/,ILEVS/-LEVEL-/
      DATA IDEBUG/-DEBUG-/,JDEBUG/0/,KDEBUG/0/
      CALL NAMES(NAME,CONST,ILEVLS)
      DO 7589 I=1,5
 7589 KRUN(I)=1
      DO 1 I=1,5
      ISMX(I)=ISM(I)
      DO 1 J=1,5
    1 ULIST(I,J)=VLIST(I,J)
      IRUN=0
      JSTORE=0
 1000 CALL CONSTS
      CALL LEVELS
      CALL TABLES
      DO 2 I=1,6
    2 PAUSE(I)=0
      DO 4 I=1,20
    4 ITIMES(I)=0
    5 IRAT(I)=0
    6 ILEV(I)=0
    7 IAUX(I)=0
      KTIME=0
      DO 3587 I=1,5
      ISM(I)=ISMX(I)
      DO 3587 J=1,5
```

```
 3987 VLIST(I,J)=PLIST(I,J)
      DO 3 I=1,20
    3 ITIMES(I)=0.
      NTIMES=0
C***INSTRUCTIONS
   39 WRITE(6,40)
   40 FORMAT(- INSTRUCTIONS0-)
 7090 NEXT=73
      ITYPE=0
      CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9998
      IF(INSYM.EQ.IPLOT) GO TO 7100
      IF(INSYM.EQ.ISTORE) GO TO 6351
      IF(INSYM.EQ.ISUMS) GO TO 7600
      IF(INSYM.EQ.ISET) GO TO 6201
      IF(INSYM.EQ.ITABLE) GO TO 7430
      IF(INSYM.EQ.IPRINT) GO TO 88
      IF(INSYM.EQ.IPAUSE) GO TO 6310
      IF(INSYM.EQ.IGO) GO TO 7055
      IF(INSYM.EQ.ISCALE) GO TO 6301
      IF(INSYM.EQ.IRESET) GO TO 7550
      IF(INSYM.EQ.IDEBUG) GO TO 6767
      IF(INSYM.EQ.JTIME) GO TO 4530
      IF(INSYM.EQ.ILIST) GO TO 4531
 9998 DO 9997 I=1,60
 9997 LINE(I)=BLANK
      LINE(NEXTO)=IARROW
      LINE(NEXT-2)=IARROW
      J=NEXTO+1
      K=NEXT-3
      DO 9995 I=J,K
 9995 LINE(I)=DOT
```

```
      WRITE(6,9996) LINE
 9996 FORMAT(6 A1)
      WRITE(6,7098)
 7098 FORMAT(- SOMETHING IS WRONG BETWEEN THE ARROWS-,/,
     +- RETYPE THE LINE-)
      GO TO 7099
C***DEBUG
 4530 J=0
      KTIME=1
 4532 CALL GETSYM
      J=J+1
      ITIMES(J)=4.*(XNUM-1960.)+1.
      IF(IDEL.EQ.1) GO TO 7099
      GO TO 4532
 4531 ITYPE=-1
      CALL GETSYM
      ITYPE=0
 4535 IF(INSYM.EQ.PNAMES(I)) IAUX(I)=1
      IF(IDEL.NE.1) GO TO 4531
      GO TO 7099
 6767 CALL GETSYM
      JDEBUG=XNUM
      CALL GETSYM
      KDEBUG=XNUM
      GO TO 7099
C***PRINT MODEL INFORMATION
   88 ITYPE=-1
      CALL GETSYM
      ITYPE=0
      IF(INSYM.EQ.IVARBL) GO TO 6490
      IF(INSYM.EQ.ICONS) GO TO 6491
      IF(INSYM.EQ.ILEVS) GO TO 6492
```

```
      DO 97 I=1,NOCONS
   97 IF(INSYM.EQ.CONST(I)) GO TO 6493
      DO 98 I=1,NOLEVS
   98 IF(INSYM.EQ.ILEVLS(I)) GO TO 6494
 6496 IF(INSYM.EQ.INAME(I)) GO TO 6497
      GO TO 9998
 6498 FORMAT(1X,A6,5(5E10.3,/,7X))
      GO TO 6489
 6490 WRITE(6,95)
   95 FORMAT(- NO-,5X,-NAME-,2X,-DESCRIPTION-)
      DO 91 I=1,NOPLTS
      WRITE(6,92)I,(NAME(J,I),J=1,6)
   92 FORMAT(I3,5X,6A6)
   91 CONTINUE
      GO TO 6489
 6491 WRITE(6,142)
  142 FORMAT(- NO-,5X,-CONST.-,5X,-PRESENT VALUE-)
      DO 143 I=1,NOCONS
  143 WRITE(6,144) I,CONST(I),Y(I)
  144 FORMAT(I3,5X,A6,5X,E10.5)
      GO TO 6489
 6492 WRITE(6,94)
   94 FORMAT(- NO-,5X,-LEVEL -,5X,-PRESENT VALUE-)
      DO 96 I=1,NOLEVS
   96 WRITE(6,144) I,ILEVLS(I),IVALS(I)
      GO TO 6489
 6493 WRITE(6,99) CONST(I),Y(I)
   99 FORMAT(1X,A6,-= -,E10.5)
      GO TO 6489
 6494 WRITE(6,99) ILEVLS(I),IVALS(I)
 6489 DO 145 I=1,2
  145 WRITE(6,16)
```

```
      IF(IDEL.EQ.1) GO TO 39
      GO TO 88
C***PLOT DATA
 7100 DO 7101 I=1,5
      DO 7101 J=1,5
 7101 VLIST(I,J)=0.
      J=1
 7102 ITYPE=-1
      JJ=1
      CALL GETSYM
      IF(IDEL.EQ.1) GO TO 7099
      DO 5436 I=1,NDPLTS
 5436 IF(INSYM.EQ.NAME(1,I)) GO TO 5437
      GO TO 9998
 5437 JJ=I
      VLIST(1,J)=I
      CALL GETSYM
      ISM(J)=INSYM
      ITYPE=0
      IF(IDEL.NE.5) GO TO 7103
      J=J+1
      GO TO 7102
 7103 CALL GETSYM
      IF(ITYPE.EQ.1) GO TO 7104
      VLIST(3,JJ)=XNUM
      GO TO 7105
 7104 IF(INSYM.NE.STAR .AND. INSYM.NE.POUND) GO TO 9998
      IF(INSYM.EQ.STAR) VLIST(2,JJ)=JJ
      IF(INSYM.EQ.POUND) VLIST(2,JJ)=-JJ
      VLIST(3,JJ)=999999.
 7105 CALL GETSYM
      IF(ITYPE.EQ.1) GO TO 7106
```

```
      VLIST(5,J0)=X[illegible]
      GO TO 7107
 7106 IF(INSYM.NE.STAR .AND. INSYM.NE.POUND) GO TO 9998
      IF(INSYM.EQ.STAR) VLIST(4,J0)=J0
      IF(INSYM.EQ.POUND) VLIST(4,J0)=-J0
      VLIST(5,J0)=-999999.
 7107 DO 7108 K=J0,J
      DO 7108 L=2,5
 7108 VLIST(L,K)=VLIST(L,J0)
      IF(J.EQ.5) GO TO 7099
      J=J+1
      J0=J
      NEXT=NEXT+1
      GO TO 7102
C***SUMMARY PLOT
 7600 IF(JSTORE.EQ.0) GO TO 9998
      CALL GETSYM
      IF(INSYM.NE.IPLOT) GO TO 9998
      DO 7611 J=1,5
      DO 7611 I=1,254
 7611 Z(I,J)=0.
      DO 7601 I=1,5
      SWC(I)=0
      Z(3,I)=999999.
      Z(4,I)=-999999.
      DO 7601 J=1,5
 7601 VLIST(I,J)=0
      J=1
      J0=1
 7602 ITYPE=-1
      CALL GETSYM
      IF(IDEL.EQ.1) GO TO 32
```

```
      DO 5567 I=1,NOPLTS
 5567 IF(INSYM.EQ.NAM(1,I)) GO TO 5568
      GO TO 9998
 5568 JJ=I
      SWC(J)=JJ
      ITYPE=0
      CALL GETSYM
      IF(ITYPE.NE.2) GO TO 9998
      IRN=XNUM
      NEXT=NEXT+1
      ITYPE=-1
      CALL GETSYM
      ISM(J)=INSYM
      ITYPE=0
      VLIST(1,J)=JJ
      KRUN(J)=IRN
      DO 7613 K=1,5
 7613 IF(IVAR(IRN,K).EQ.JJ) GO TO 7614
      GO TO 9998
 7614 KK=SZ(IRN,1,K)+4
      DO 7615 I=1,KK
      IF(I.EQ.3 .OR. I.EQ.4) GO TO 7615
      Z(I,J)=SZ(IRN,I,K)
      IF(I.LE.4) GO TO 7615
      Z(3,JJ)=AMIN1(Z(3,JJ),Z(I,J))
      Z(4,JJ)=AMAX1(Z(4,JJ),Z(I,J))
 7615 CONTINUE
      IF(IDEL.EQ.2) GO TO 7603
      J=J+1
      GO TO 7602
 7603 CALL GETSYM
      IF(ITYPE.EQ.1) GO TO 7604
```

```
      Z(3,JJ)=XNUM
      GO TO 7605
 7604 IF(INSYM.NE.STAR .AND. INSYM.NE.POUND) GO TO 9998
      IF(INSYM.EQ.STAR) VLIST(2,JJ)=JJ
      IF(INSYM.EQ.POUND) VLIST(2,JJ)=-JJ
 7605 CALL GETSYM
      IF(ITYPE.EQ.1) GO TO 7606
      Z(4,JJ)=XNUM
      GO TO 7607
 7606 IF(INSYM.NE.STAR .AND. INSYM.NE.POUND) GO TO 9998
      IF(INSYM.EQ.STAR) VLIST(4,JJ)=JJ
      IF(INSYM.EQ.POUND) VLIST(4,JJ)=-JJ
 7607 DO 7608 K=JJ,J
      Z(3,K)=Z(3,JJ)
      Z(4,K)=Z(4,JJ)
      DO 7608 L=2,5
 7608 VLIST(L,K)=VLIST(L,JJ)
      IF(J.EQ.5) GO TO 32
      J=J+1
      JJ=J
      NEXT=NEXT+1
      GO TO 7602
C***RESET VALUES
 7550 CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9998
      IF(INSYM.EQ.ICONS) GO TO 7551
      IF(INSYM.EQ.ILEVS) GO TO 7552
      GO TO 9998
 7551 CALL CONSTS
      GO TO 7553
 7552 CALL LEVELS
 7553 IF(IDEL.EQ.1) GO TO 7099
```

```
      GO TO 7099
C***STORE SETUP
 6351 CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9998
      IF(INSYM.EQ.ION) GO TO 6352
      IF(INSYM.EQ.IOFF) GO TO 6353
      GO TO 9998
 6352 JSTORE=1
      IRUN=0
      GO TO 7099
 6353 JSTORE=0
      IRUN=0
      DO 6354 J=1,5
 6354 KRUN(J)=1
      GO TO 7099
C***SCALE DATA
 6301 CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9998
      INAME=INSYM
      CALL GETSYM
      IF(ITYPE.EQ.2) GO TO 6302
      IF(INAME.EQ.IHALF .AND. INSYM.EQ.NO) HALF=BLANK
      IF(INAME.EQ.IHALF .AND. INSYM.EQ.YES) HALF=DOT
      IF(INAME.EQ.IONE .AND. INSYM.EQ.NO) ONE=BLANK
      IF(INAME.EQ.IONE .AND. INSYM.EQ.YES) ONE=DOT
      GO TO 6303
 6302 IF(INAME.NE.ISPACE) GO TO 9998
      SPACES=XNUM
 6303 IF(IDEL.EQ.1) GO TO 7099
      GO TO 6301
C***PAUSE DATA
 6310 IPASS=0
```

```
      DO 6311 I=1,6
      IF(IPASS.EQ.1) GO TO 6312
      CALL GETSYM
      IF(ITYPE.NE.2) GO TO 9998
      PAUSE(I)=XNUM
      IF(IDEL.EQ.1 .OR. I.EQ.5) IPASS=1
      GO TO 6311
 6312 PAUSE(I)=999999
 6311 CONTINUE
      GO TO 7099
C***CHANGE TABLES
 7430 CALL GETSYM
      J=0
 7431 IF(INSYM.EQ.TNAME(I)) GO TO 7432
      GO TO 9998
 7432 J=J+1
      CALL GETSYM
      IF(IDEL.EQ.8)GO TO 7099
      GO TO 7432
C***MISC INPUT DATA
 6201 CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9998
      INAME=INSYM
      CALL GETSYM
      IF(ITYPE.NE.2) GO TO 9998
      INSYM=INUM L
      DO 6205 I=1,NOCONS
 6205 IF(INSYM.EQ.CONST(I)) GO TO 6206
      DO 6207 I=1,NOLEVS
 6207 IF(INSYM.EQ.ILEVELS(I)) GO TO 6208
      IF(JSTOWL.EQ.1) GO TO 6221
      IF(INSYM.EQ.IOT) GO TO 6209
```

```
      IF(INSYM.EQ.ISTRT) GO TO 6202
      IF(INSYM.EQ.IPLTPD) GO TO 6203
      IF(INSYM.EQ.NYRS) GO TO 6204
      GO TO 9998
 6221 WRITE(6,6222) INSYM
 6222 FORMAT(1X.A6.- CANNOT BE CHANGED WHILE STORE IS ON.-)
      GO TO 6210
 6202 TIMEO=XNUM
      GO TO 6210
 6203 PLTPD=XNUM
      GO TO 6210
 6204 NOYRS=XNUM
      GO TO 6210
 6206 Y(I)=XNUM
      GO TO 6210
 6208 IVALS(I)=XNUM
      GO TO 6210
 6209 DT=XNUM
 6210 IF(IDEL.EQ.1) GO TO 7099
      GO TO 6201
 7055 DO 7056 I=1,5
      Z(3,I)=VLIST(3,I)
      Z(4,I)=VLIST(5,I)
 7056 SPC(I)=VLIST(1,I)
C***INITIAL VALUES
   16 FORMAT( )
      INP=1
      IF(PLTPD.LE.DT) GO TO 6241
      INP=PLTPD/DT
      IF(MOD(PLTPD,DT).GT.0) INP=INP+1
 6241 MAX=(NOYRS/DT)+1
      IF(MOD(NOYRS,DT).GT.0) MAX=MAX+1
```

```
      TMS=INPRNT
      IFLG=0
      DO 562 I=1,5
      NPTS=((MAX-1)/INP)+1
      Z(1,I)=NPTS
  562 Z(2,I)=TMS
      DO 61 I=1,2
   61 WRITE(6,16)
      TIME=TIMES
      CALL MODEL(TIME,DT,TMS,SWC,VLIST,Z,MAX)
   32 DO 72 I=1,5
      J0=ABS(VLIST(2,I))
      IF(J0.NE.I) GO TO 73
      DO 75 J=J0,5
      IF(IFIX(ABS(VLIST(2,J))).NE.J0) GO TO 75
      Z(3,J)=Z(3,J0)
   75 CONTINUE
   73 J0=ABS(VLIST(4,I))
      IF(J0.NE.I) GO TO 72
      DO 81 J=J0,5
      IF(IFIX(ABS(VLIST(4,J))).NE.J0) GO TO 81
      Z(4,J)=Z(4,J0)
   81 CONTINUE
   72 CONTINUE
      WRITE(6,33)
   33 FORMAT(1X,-RUN-,4X,-LOWER-,4X,-UPPER-,3X,-SYM-,3X,
     +-NAME-,2X,-DESCRIPTION-)
      DO 34 K=1,5
      IF(SWC(K).EQ.0) GO TO 34
      KM=SWC(K)
      WRITE(6,35) KRUN(K),Z(3,K),Z(4,K),ISM(K),(NAME(J,KM),J=1,6)
   35 FORMAT(1X,I3,2E9.3,4X,A2,3X,6A6)
```

```
   34 CONTINUE
      DO 62 L=1,2
   62 WRITE(6,16)
      IF(IFLG.EQ.1) GO TO 6000
      IFLG=1
      ICOUNT=0
      DO 6101 I=2,4,2
      IF(I.EQ.2)KK=3
      IF(I.EQ.4)KK=4
      DO 6102 J=1,5
      IF(VLIST(I,J).GT.-.5) GO TO 6102
      IF(-IFIX(VLIST(I,J)).NE.J) GO TO 6102
      K=VLIST(1,J)
      IF(I.EQ.2) WRITE(6,6103) K
 6103 FORMAT(- NEW LOWER-,I3,-0-)
      IF(I.EQ.4) WRITE(6,6104) K
 6104 FORMAT(- NEW UPPER-,I3,-0-)
      NEXT=73
      CALL GETSYM
      IF(ITYPE.EQ.0) GO TO 6102
      ICOUNT=1
      Z(KK,J)=XNUM
 6102 CONTINUE
 6101 CONTINUE
      IF(ICOUNT.EQ.0) GO TO 6000
      DO 83 II=1,2
   83 WRITE(6,16)
      GO TO 32
 6000 CONTINUE
      IFLG=0
      TIME=TIM[illegible]
      CALL PLOT(TIME,SUC,Z,ISM,HALF,CNL,SPACES,PAUSE)
```

```
      DO 63 I=1,2
   63 WRITE(6,16)
      IF(JSTORE.EQ.0) GO TO 1000
C***STORE RUN DATA
 7109 IRUN=IRUN+1
      IF(IRUN.GT.10) GO TO 9998
      DO 7110 J=1,5
      KRUN(J)=IRUN+1
      IVAR(IRUN,J)=VLIST(1,J)
      DO 7110 I=1,254
 7110 SZ(IRUN,I,J)=Z(I,J)
      GO TO 1000
      END
      SUBROUTINE PLOT(TIME,SWC,Z,ISM,HALF,ONE,SPACES,PAUSE)
      DIMENSION SWC(5),Z(254,5),ISM(5)
      DIMENSION LINE(60)
      DIMENSION PAUSE(6),ISUM(5),IPLT(5)
      INTEGER BLANK,DOT,SWC,NO,HALF,ONE,SPACES,PAUSE
      DATA EPS/.000001/
      DATA BLANK/- -/
      DATA DOT/-.-/
      DATA NO/-NO-/
      DO 610 I=1,5
  610 ISUM(I)=0
      NPTS=Z(1,1)+4
      TMS=Z(2,1)
      STIME=PAUSE(1)
      IJ=1
      TIME=TIME-TMS
      DO 10 II=5,NPTS
      DO 612 I=1,5
  612 IPLT(I)=0
```

```
      TIME=TIME+TMS
      JJ=II-5
      IF(MOD(JJ,SPACES).NE.0) GO TO 13
      DO 12 I=1,60
   12 LINE(I)=DOT
      GO TO 14
   13 DO 1 I=1,60
    1 LINE(I)=BLANK
      LINE(30)=HALF
      LINE(60)=ONE
   14 DO 1000 J=1,5
      MAX=-1
      DO 611 K=1,5
      IF(IPLT(K).EQ.1) GO TO 611
      IF(MAX.GE.ISUM(K)) GO TO 611
      MAX=ISUM(K)
      I=K
  611 CONTINUE
      IPLT(I)=1
      IF(SWC(I).EQ.0) GO TO 1000
      IF(Z(4,I)-Z(3,I).LE.EPS) GO TO 1000
      IF(Z(II,I).LT.Z(3,I)) GO TO 1000
      IF(Z(II,I).GT.Z(4,I)) GO TO 1000
      K=((Z(II,I)-Z(3,I))/(Z(4,I)-Z(3,I)))*60.0+1
      IF(K.GT.60) K=60
      IF(K.LT.1) K=1
  400 IF(LINE(K).NE.BLANK) GO TO 402
  401 LINE(K)=ISM(I)
      ISUM(I)=0
      GO TO 1000
  402 IF(LINE(K).EQ.DOT) GO TO 401
      ISUM(I)=ISUM(I)+1
```

```
 1000 CONTINUE
      PTIME=IFIX(TIME)
      WRITE(6,9001) PTIME,LINE
      IF(TIME.LT.STIME) GO TO 10
      IF((TIME-TBS).LT.PAUSE(IJ) .AND. PAUSE(IJ).LE.TIME) GO TO 55
      GO TO 10
   55 READ(5,111)IS3
  111 FORMAT(A1)
      IJ=IJ+1
   10 CONTINUE
      RETURN
 9001 FORMAT(1X,F5.0,60A1)
      END
      SUBROUTINE PRINT(TIME,ISWC)
      DIMENSION PRNT(20)
      COMMON /PTVRS/ IA,IR,IF
      COMMON /PNAMS/ NAMES
      COMMON /PRNT/
      DATA NAMES/
      IF(ISWC.EQ.1) GO TO 1
      IF(J+1.GT.20)RETURN
      J=J+1
      PRNT(J)=TIME
      RETURN
    1 WRITE(6,76)(PRNT(I),I=1,J)
      IF(IA(I).EQ.1) WRITE(6,77)NAMES(I),(A(I,K),K=1,J)
    2 CONTINUE
      IF(IR(I).EQ.1) WRITE(6,77) NAMES(I+76),(R(I,K),K=1,J)
    3 CONTINUE
      IF(IF(I).EQ.1) WRITE(6,77) NAMES(I+1 6),(F(I,K),K=1,J)
    4 CONTINUE
      RETURN
```

```
77 FORMAT(1X,[illegible])
76 FORMAT(/,[illegible](7X,[illegible]F1[illegible].2,/))
      END
      SUBROUTINE GETSYM
      DIMENSION IA(72)
      COMMON /BLOCK1/ INSYM,ITYPE,XNUM,IDEL,NEXTO,NEXT
      DATA ISTOP/-STOP-/
      LENGTH=0
      XNUM=0.0
      NEXTO=NEXT
      IF(ITYPE.NE.-1) ITYPE=0
      IDEC=-50
      ISIGN=0
      IF (NEXT .LE. 72) GO TO 10
620   NEXT=1
      NEXTO=NEXT
      READ(5,1) IA
    1 FORMAT(72A1)
10    DO 100 I=NEXT,72
      IF(ITYPE.EQ.-1) GO TO 47
      IF (IA(I) .NE. -+-) GO TO  48
      IF (ITYPE .NE. 0) GO TO 891
      ISIGN=1
      ITYPE=2
      GO TO 100
48    IF (IA(I) .NE. ---) GO TO 49
      IF (ITYPE .NE. 0) GO TO 892
      ISIGN=-1
      ITYPE=2
      GO TO 100
49    IF (IA(I) .EQ. -D-) GO TO 200
      IF (IA(I) .EQ. -I-) GO TO 200
```

```
      IF (IA(I) .EQ. -2-) GO TO 200
      IF (IA(I) .EQ. -3-) GO TO 200
      IF (IA(I) .EQ. -4-) GO TO 200
      IF (IA(I) .EQ. -5-) GO TO 200
      IF (IA(I) .EQ. -6-) GO TO 200
      IF (IA(I) .EQ. -7-) GO TO 200
      IF (IA(I) .EQ. -8-) GO TO 200
      IF (IA(I) .EQ. -9-) GO TO 200
   47 IF (IA(I) .EQ. - -) GO TO 300
      IF(IA(I) .EQ. -)-) GO TO 1200
      IF(IA(I) .EQ. -*-) GO TO 1300
      IF (IA(I) .EQ. -(-) GO TO 400
      IF(IA(I) .EQ. -.-) GO TO 501
      IF(IA(I) .EQ. -)-) GO TO 500
      IF (IA(I) .EQ. -,-) GO TO 600
      IF(IA(I) .EQ. -/-) GO TO 700
      IF(IA(I) .EQ. -=-) GO TO 1100
      IF (ITYPE .EQ. 2) GO TO 800
   50 IF (LENGTH .EQ. 5) GO TO 100
      FLD(LENGTH*6,6,INSYM) = FLD(0,6,IA(I))
      LENGTH=LENGTH+1
      IF(ITYPE.NE.-1) ITYPE=1
      GO TO 100
  200 XNUM=FLD(0,6,IA(I))-48+XNUM*10.0
      IDEC=IDEC+1
      ITYPE=2
      GO TO 100
  300 IDEL=1
      GO TO 900
  400 IDEL=2
      GO TO 900
  500 IDEL=3
```

```
      GO TO 7
 501  IDEC=
      I=I+1
      GO TO 1
 500  IDEL=5
      GO TO 9
  700 IDEL=8
      GO TO 900
 1100 IDEL=1
      GO TO 900
 1200 IDEL=12
      GO TO 900
 1300 IDEL=13
      GO TO 900
 800  WRITE(6,2) IA(I),I
 2    FORMAT(- IMPROPER CHARACTER -,A1,- IN COLUMN -,I2)
      WRITE(6,3)
 3    FORMAT(- RETYPE THE LINE-)
      IDEL=9
      GO TO 900
 691  IDEL=6
      GO TO 901
 692  IDEL=7
      GO TO 901
  100 CONTINUE
      GO TO 620
 900  I=I+1
 901  IF (ITYPE .EQ. 1) GO TO 950
      IF(ITYPE .EQ. -1) GO TO 950
      IF (IDEC .LT. 0) GO TO 520
      DO 910 J=1,IDEC
 910  X(J)=X(J)/10.0
```

```
   62   [illegible]
        [illegible]
   99   [illegible]
        [illegible]
        [illegible]
  760   CONTINUE
  999   NEXT=1
        IF(INCYM .EQ. ISTOP) STOP
        RETURN
        END
        SUBROUTINE NAMES(NAME,CONST,LEVELS)
        INTEGER CONST
        DATA NAME1/
        DATA NAME2/
        DATA ICONS/
        DATA ILEVELS/
        DO 1 I=1,6
      3 NAME(I,J)=NAME1(I,J)
      4 NAME(I,J+18)=NAME2(I,J)
      1 CONTINUE
      5 CONST(I)=ICONS(I)
      6 LEVELS(I)=ILEVLS(I)
        RETURN
        END
        FUNCTION TABLE(TABLE,POINT,P1,PI,P2)
        DIMENSION TABLE(2
        DATA EPS/.0001/
        IF(POINT.LE.P1) GO TO 1
        IF(POINT.GE.PI) GO TO 2
        PT=1+(POINT-P1)/P2
        I=PT
        IF(PT-EPS.LT.FLOAT(I)) GO TO 3
```

```
      TABHL=TABLE(L)
      GO TO 999
   30 TABHL=TABLE(I)+(TABLE(I+1)-TABLE(I))*(RT-I)
      GO TO 999
   10 TABHL=TABLE(1)
      GO TO 999
   20 MAX=1+(RM-R[illegible])/R2
      TABHL=TABLE(MAX)
  999 RETURN
      END
      FUNCTION CLIP(A,B,C,D)
      IF(C.GE.D)CLIP=A
      IF(C.LT.D)CLIP=B
      RETURN
      END
      SUBROUTINE TABLES
      COMMON /TABLE/ NAMES
      RETURN
      END
      SUBROUTINE CONSTS
      RETURN
      END
      SUBROUTINE LEVELS
      RETURN
      END
      SUBROUTINE MODEL(TIME,DT,[illegible],LC,ZLIST,Z,MAX)
      DIMENSION ITIMES(20)
      COMMON /TIMES/ ITIMES,RTI[illegible]
      COMMON /PRINT/
      INTEGER S,C
      COMMON /RUNS/ [illegible]
C***INITIAL VALUES
```

```
C***WE [illegible] TWIT [illegible]
      II [illegible]
      KK=4
      DO [illegible] INDEX=1,NX
      II [illegible]
C***AUXILIARY EQUATIONS
C***SURFACE [illegible]
C***RATE EQUATIONS
C***STORE PLOT DATA
      IF(KTIM[illegible]) GO TO 4555
      DO 4551 I=1,[illegible]
 4551 IF(INDEX.[illegible]TIMES(I))CALL PRINT(TIME,J)
 4555 CONTINUE
      IF(JDEBUG.LE.INDEX .AND. INDEX.LE.KDEBUG)WRITE(6,1234)
 1234 FORMAT(
     +,/,-    1-,4(5(1X,E12.5),/,4X),5(1X,E10.5),
     +,/,-  26-,4(5(1X,E1[illegible].5),/,4X),5(1X,E10.5),
     +,/,-  [illegible]1-,4(5(1X,E1[illegible].5),/,4X),5(1X,E10.5),
     +,/,-  [illegible]-,4(5(1X,E1[illegible].5),/,4X),5(1X,E1[illegible].5),
     +,/,- 101-,4(5(1X,E1[illegible].5),/,4X),5(1X,E10.5),
     +,/,- 12[illegible]-,4(5(1X,E1[illegible].5),/,4X),5(1X,E1[illegible].5),
     +,/,- 151-,4(5(1X,E1[illegible]),/,4X),5(1X,E1[illegible].5),
     +,/,- 175-,4(5(1X,E1[illegible].5),/,4X),5(1X,E1[illegible].5))
      IF([illegible](TIME,TYS),[illegible]) GO TO 5[illegible]
      KK=KK+1
      DO 56[illegible] II=1,[illegible]
      IF([illegible](II).[illegible]) GO TO 56[illegible]
      LL=[illegible](II)
      KI=[illegible](VLIST(2,II))
      IF(KI.[illegible]) GO TO 55[illegible]
      Z(3,KI)=AMIN1(Z(3,KI),X(LL))
  55[illegible] KI=[illegible](VLIST(4,II))
```

```
      IF(NI.[illegible]) [illegible]
      Z(K,[illegible])=[illegible]
  158 Z(K,[illegible])=[illegible]
    [illegible] CONTINUE
 [illegible] CONTINUE
C***LEVEL [illegible]ATIONS
 5000 CONTINUE
      IF(NTI[illegible].EQ.1) CALL PRINT(-1,[illegible])
      RETURN
      END
```

```
[illegible]
[illegible]                                                    1001
[illegible]                                                    1002
[illegible]                                                    1003
[illegible] (1,7)
[illegible]                                                    1004
[illegible]                                                    1005
[illegible]                                                    1006
      DATA [illegible]/-LIST-/,[illegible]/-[illegible]-/      1007
      DIMENSION IAUX(125),I[illegible]T(2[illegible]),I[illegible](156)
      COMMON /PTVRS/ IAUX,I[illegible]T,I[illegible]           1008
      DIMENSION ITI[illegible](2[illegible])                   1009
      COMMON /TIMES/ ITI[illegible],KTI[illegible]             1010
C --------  -                                                  1011
      [illegible]S=[illegible]
      N LEVS=46
      [illegible]
C---------+ -                                                  1012
      DIMENSION T[illegible](25)
      INTEGER T[illegible]                                     1013
      COMMON /T[illegible]/ T[illegible]                       1014
      DAT[illegible] [illegible]/-T[illegible]-/               1015
      DIMENSION Z(2[illegible]4,5), [illegible]
      DIMENSION [illegible](196),[illegible](26), [illegible](5)
      DIMENSION [illegible]                                    1016
      DIMENSION [illegible]                                    1017
      DIMENSION [illegible]
      DIMENSION [illegible]
     +I[illegible](7),[illegible],
     +I[illegible],
     +I[illegible],
     [illegible]
```

```
     +ICC [illegible],I[illegible],IC[illegible],ICC [illegible],I[illegible],I[illegible],I[illegible],I[illegible],I[illegible],
     +ICF[illegible],IC[illegible],I[illegible],ICU[illegible],F[illegible],I[illegible],I[illegible],[illegible]
      CO[illegible] /[illegible]/ I[illegible]Y[illegible],I[illegible],[illegible],[illegible],YIJ,[illegible]XT     10[illegible]
      C[illegible] /Y/ Y                                                10[illegible]
      CO[illegible] /I/ IVALS                                           10[illegible]
      COMMON /[illegible]/ [illegible]                                  1021
      REAL IVALS                                                        1022
      INTEGER YES,CONST,S[illegible],[illegible],[illegible]LF,[illegible],[illegible]T     1023
      INTEGER P[illegible]S[illegible],STAR,PO[illegible],[illegible]     1024
      DATA VLIST/2.,[illegible].,[illegible].,[illegible].,5[illegible].,
     +5.,0.,0.,[illegible].,1[illegible].,
     +12.,[illegible].,[illegible].,[illegible].,4[illegible].,
     +19.,[illegible].,[illegible].,[illegible].,500.,
     +24.,[illegible].,[illegible].,[illegible].,2[illegible]./
      DATA IS[illegible]/-C-,-[illegible]-,-I-,-L-,-P-/                 1025
      DATA NO/-NO-/,DOT/-.-/,BLANK/- -/,ISPACE/-SPACE-/                 1026
      DATA I[illegible]/-*-/                                            1027
      DATA ISCALE/-SCALE-/,IHALF/-HALF-/,IONE/-[illegible]E-/,[illegible]PACE[illegible]/[illegible]/     10[illegible]
      DATA ISET/-SET-/,IST[illegible]RT/-START-/,IPLTPD/-PLTPD-/,[illegible]/-[illegible]YRS-/     10[illegible]
      DATA IPA/-PA-/,IAP/-AP-/,IAR/-AR-/,IEA/-EA-/                      10[illegible]
      DATA IS[illegible]/-S[illegible]-/,ISTORE/-STORE-/,I[illegible]/-[illegible]-/,IOFF/-[illegible]-/     10[illegible]
      DATA [illegible]/196[illegible]./,PLTP[illegible]/[illegible]./,[illegible]YRS/2[illegible]/     10[illegible]
      DATA STAR/-*-/,P[illegible]/-[illegible]-/,IPLOT/-PLOT-/,I[illegible]/-[illegible]-/     10[illegible]
      DATA HALF/-.-/,[illegible]/-.-/,[illegible]T/-[illegible]T-/,[illegible]T/.2[illegible]/,IPA[illegible]/-[illegible]-/     10[illegible]
      DATA [illegible]/[illegible]/.[illegible]/     10[illegible]
      DATA IVARBLE/-VARIA-/,I[illegible]/-[illegible]T-/,I[illegible]T/-[illegible]T-/     10[illegible]
      DATA Y[illegible]/-[illegible]-/,[illegible]/-[illegible]-/,I[illegible]/-[illegible]-/     [illegible]
      DATA I[illegible]/-[illegible]-/,[illegible]/[illegible]/     [illegible]
      CALL [illegible]     [illegible]
      DO 7501 I=1,5     [illegible]
 7501 [illegible](I)=1     [illegible]
      [illegible] I=1,5     [illegible]
```

```
      I[illegible]X(I)=IS[illegible](I)                                1043
      [illegible] J=1,[illegible]                                       1044
    1 VLIST(I,J)=VLIS[illegible](I,J)                                   1045
      I[illegible]=[illegible]                                          1046
      JST[illegible]=                                                   1047
 1000 CALL CONSTS                                                       1048
      CALL LEVELS                                                       1049
      CALL TABLES                                                       1050
      DO 2 I=1,6                                                        1051
    2 PAUSE(I)=0                                                        1052
      DO 4 I=1,20                                                       1053
    4 ITIMES(I)=0                                                       1054
      DO 5 I=1,30
    5 IRAT(I)=0                                                         1055
      DO 6 I=1,45
    6 ILEV(I)=0                                                         1056
      DO 7 I=1,125
    7 IAUX(I)=0                                                         1057
      KTIME=0                                                           1058
      DO 3587 I=1,5                                                     1059
      IST(I)=ISAX(I)                                                    1060
      DO 3587 J=1,5                                                     1061
 3587 VLIST(I,J)=ULIST(I,J)                                             1062
      DO 3 I=1,20                                                       1063
    3 ITIMES(I)=0                                                       1064
      KTIME=[illegible]                                                 1065
C***INSTRUCTIONS                                                        1066
   39 WRITE(6,40)                                                       1067
   40 FORMAT(' INSTRUCTIONS ')                                          1068
 7000 NEXT=73                                                           1069
      ITYPE=0                                                           1070
      CALL SETUP                                                        1071
```

```
      IF(INSYM.EQ.[illegible]) GO TO [illegible]
      IF(INSYM.EQ.[illegible]) GO TO [illegible]
      IF(INSYM.EQ.INT[illegible]) GO TO [illegible]
      IF(INSYM.EQ.[illegible]) GO TO 7[illegible]
      IF(INSYM.EQ.I[illegible]) GO TO 62[illegible]1
      IF(INSYM.EQ.I[illegible]) GO TO 7[illegible]
      IF(INSYM.EQ.IPRINT) GO TO [illegible]
      IF(INSYM.EQ.I[illegible]) GO TO [illegible]
      IF(INSYM.EQ.IOUT) GO TO 7055
      IF(INSYM.EQ.ISCALE) GO TO 6301
      IF(INSYM.EQ.IRESET) GO TO 7[illegible]
      IF(INSYM.EQ.IDEBUG) GO TO 6767
      IF(INSYM.EQ.JTIME) GO TO 4530
      IF(INSYM.EQ.ILIST) GO TO 4531
 9998 DO 9997 I=1,6[illegible]
 9997 LINE(I)=BLANK
      LINE(NEXT)=IARROW
      LINE(NEXT-2)=IARROW
      J=NEXT+1
      K=NEXT-3
      DO 9995 I=J,K
 9995 LINE(I)=DOT
      WRITE(6,9996) LINE
 9996 FORMAT(6[illegible])
      WRITE(6,[illegible])
 [illegible] FORMAT(- [illegible] IS [illegible] THE [illegible]-,/,
     +- RETYPE THE LINE-)
      GO TO 7[illegible]
 [illegible]
 453[illegible] J=[illegible]
      [illegible]TI[illegible]=1
 4532 CALL [illegible]
```

```
      J=J+1                                                        1104
      [illegible]                                                  1105
      [illegible]                                                  1106
      [illegible]                                                  1107
      [illegible] ITYPE=-1                                         1108
      CALL [illegible]                                             1109
      ITYPE=                                                       1110
      DO 4535 I=1,174
 4535 IF(INSYM.EQ.PNAMES(I)) IAUX(I)=1                             1111
      DO 4533 I=1,25
 4533 IF(INSYM.EQ.PNAMES(I+126)) IRAT(I)=1
      DO 4534 I=1,46
 4534 IF(INSYM.EQ.PNAMES(I+151)) ILEV(I)=1
      IF(IDEL.NE.1) GO TO 4531                                     1112
      GO TO 7099                                                   1113
 6767 CALL GETSYM                                                  1114
      [illegible]                                                  1115
      CALL GETSYM                                                  1116
      [illegible]                                                  1117
      GO TO 7099                                                   1118
C***PRINT MODEL INFORMATION                                        1119
   65 ITYPE=-1                                                     1120
      CALL GETSYM                                                  1121
      ITYPE=                                                       1122
      [illegible]                                                  1123
      [illegible]                                                  1124
      [illegible]                                                  1125
      DO 97 I=1,[illegible]                                        1126
   97 [illegible]                                                  1127
      [illegible]                                                  1128
      [illegible]                                                  1129
      DO 6496 I=1,[illegible]
```

```
[illegible]
[illegible]
[illegible]
      IF(I.EQ.2) WRITE(6,64[illegible]) INSYM,TCCJ
      IF(I.EQ.3) WRITE(6,6498) INSYM,TCCF
      IF(I.EQ.4) WRITE(6,6498) INSYM,TCC[illegible]
      IF(I.EQ.5) WRITE(6,6498) INSYM,TCCJ
      IF(I.EQ.6) WRITE(6,649[illegible]) INSYM,TC[illegible]
      IF(I.EQ.7) WRITE(6,6498) INSYM,TCCC
      IF(I.EQ.8) WRITE(6,6497) INSYM,TCCD
      IF(I.EQ.9) WRITE(6,6498) INSYM,TCCF
      IF(I.EQ.10) WRITE(6,6498) INSYM,TCCJ
      IF(I.EQ.11) WRITE(6,6498) INSYM,TCCY
      IF(I.EQ.12) WRITE(6,6498) INSYM,TCCP
      IF(I.EQ.13) WRITE(6,6498) INSYM,TCCR
      IF(I.EQ.14) WRITE(6,6498) INSYM,TCCX
      IF(I.EQ.15) WRITE(6,6498) INSYM,TCD[illegible]
      IF(I.EQ.16) WRITE(6,6498) INSYM,TCE[illegible]
      IF(I.EQ.17) WRITE(6,6498) INSYM,TCF[illegible]
      IF(I.EQ.18) WRITE(6,6498) INSYM,TCFC
      IF(I.EQ.19) WRITE(6,6498) INSYM,TCFJ
      IF(I.EQ.20) WRITE(6,6498) INSYM,TC[illegible]
      IF(I.EQ.21) WRITE(6,6498) INSYM,TCJ[illegible]
      IF(I.EQ.22) WRITE(6,6498) INSYM,[illegible]
      IF(I.EQ.23) WRITE(6,64[illegible]) INSYM,[illegible]
      IF(I.EQ.24) WRITE(6,64[illegible]) INSYM,T[illegible]
      IF(I.EQ.25) WRITE(6,6498) INSYM,T[illegible]
      IF(I.EQ.26) WRITE(6,6498) INSYM,[illegible]
 6497 FORMAT([illegible])
[illegible]
 649[illegible] [illegible]
[illegible]
```

```
      DO 91 I=1,[illegible]
      WRITE(6,92) I,([illegible],I),J=1,6)
   92 FORMAT(I3,5X,[illegible])
   91 CONTINUE
      GO TO 6489
 6491 WRITE(6,142)
  142 FORMAT(- NO-,5X,-CONST.-,5X,-PRESENT VALUE-)
      DO 143 I=1,NICONS
  143 WRITE(6,144) I,CONST(I),Y(I)
  144 FORMAT(I3,5X,A6,5X,E10.5)
      GO TO 6489
 6492 WRITE(6,94)
   94 FORMAT(- NO-,5X,-LEVEL -,5X,-PRESENT VALUE-)
      DO 96 I=1,NCLEVS
   96 WRITE(6,144) I,ILEVLS(I),IVALS(I)
      GO TO 6489
 6493 WRITE(6,99) CONST(I),Y(I)
   99 FORMAT(1X,A6,-= -,E10.5)
      GO TO 6489
 6494 WRITE(6,99) ILEVLS(I),IVALS(I)
 6489 DO 145 I=1,2
  145 WRITE(6,16)
      IF(IDEL.EQ.1) GO TO 39
      GO TO 68
C***COLLECT DATA
 7100 DO 7101 I=1,[illegible]
      DO 7101 J=1,6
 7101 VLIST(I,J)=0.
      J=1
 7102 ITYPE=-1
      JJ=1
      CALL GETSYM
```

```
      IF(I[illegible].[illegible].1) [illegible] TO [illegible]
      [illegible] I=1,[illegible]
 5436 IF([illegible] .[illegible].NAME(I.1)) GO TO 5437
      GO TO [illegible]
 5437 JJ=1
      VLIST(1.J)=I
      CALL GETSYM
      I[illegible](J)=INSYM
      ITYPE=[illegible]
      IF(IDEL.NE.5) GO TO 7103
      J=J+1
      GO TO 7102
 7103 CALL GETSYM
      IF(ITYPE.EQ.1) GO TO 7104
      VLIST(3.J )=XNUM
      GO TO 7105
 7104 IF(INSYM.NE.STAR .AND. INSYM.NE.POUND) GO TO 9998
      IF(INSYM.EQ.STAR) VLIST(2.JJ)=JJ
      IF(INSYM.EQ.POUND) VLIST(2.JJ)=-JJ
      VLIST(3.JJ)=99999.
 7105 CALL GETSYM
      IF(ITYPE.EQ.1) GO TO 7106
      VLIST(5.JJ)=XNUM
      GO TO 7107
 7106 IF(INSYM .NE.STAR .AND. INSYM.NE.POUND) GO TO 9998
      IF(INSYM.EQ.STAR) VLIST(4.JJ)=JJ
      IF(INSYM.EQ.POUND) VLIST(4.JJ)=-JJ
      VLIST(5.J )=-99999.
 7107 DO 7108 K=J .J
      DO 7108 L=2.5
 7108 VLIST(L.K)=VLIST(L.J )
      IF(J.[illegible].5) GO TO 7[illegible]
```

```
      J=J+1
      JJ=J
      NEXT=NEXT+1
      GO TO 7602
C***SUMMARY PLOT
 [illegible] IF(JSTORE.[illegible]) GO TO 9975
      CALL GETSYM
      IF(INSYM.NE.IPLOT) GO TO 9998
      DO 7611 J=1,5
      DO 7611 I=1,254
 7611 Z(I,J)=0.
      DO 7601 I=1,5
      SAC(I)=0
      Z(3,I)=999999.
      Z(4,I)=-999999.
      DO 7601 J=1,5
 7601 VLIST(I,J)=0
      J=1
      JJ=1
 7602 ITYPE=-1
      CALL GETSYM
      IF(IDEL.EQ.1) GO TO 32
      DO 5567 I=1,NPLTS
 5567 IF(INSYM.EQ.[illegible]E(1,I)) GO TO 5568
      GO TO 9998
 5568 JJ=I
      SAC(J)=JJ
      ITYPE=0
      CALL GETSY
      IF(ITYPE.NE.2) GO TO 9998
      IRL=X[illegible]
      NEXT=NEXT+1
```

```
      ITY[illegible]=-1
      [illegible]
      I[illegible](J)=I[illegible]
      ITY[illegible]=[illegible]
      V[illegible]INT(1,J)=JJ
      [illegible](J)=IRN
      DO 7613 K=1,[illegible]
 7613 IF(IVAR(IRN,K).EQ.JJ) GO TO 7614
      GO TO 9998
 7614 KK=SZ(IRN,1,K)+4
      DO 7615 I=1,KK
      IF(I.EQ.3 .OR. I.EQ.4) GO TO 7615
      Z(I,J)=SZ(IRN,I,K)
      IF(I.LE.4) GO TO 7615
      Z(3,JJ)=AMIN1(Z(3,JJ),Z(I,J))
      Z(4,JJ)=AMAX1(Z(4,JJ),Z(I,J))
 7615 CONTINUE
      IF(IDEL.EQ.2) GO TO 7603
      J=J+1
      GO TO 76[illegible]2
 7603 CALL GETSYM
      IF(ITYPE.EQ.1) GO TO 76[illegible]4
      Z(3,J[illegible])=[illegible]
      GO TO 76[illegible]5
 76[illegible]4 IF(INSY[illegible] [illegible]
      IF(INSY[illegible]) V[illegible]INT(2,J[illegible])=J[illegible]
      IF(INSY[illegible]) V[illegible]INT(2,J[illegible])=-J[illegible]
 76[illegible]5 CALL GETSY[illegible]
      IF(ITYP[illegible].1) [illegible]
      Z(4,J[illegible])=[illegible]
      [illegible] TO 76[illegible]7
 76[illegible]6 IF(INSY[illegible] [illegible]
```

```
      IF(INSY[illegible]) VLIST(4,J)=J
      IF(INSY[illegible]) VLIST(4,J)=-J
 7[illegible] DO 7608 K=JO,J
      Z(3,K)=Z(3,JO)
      Z(4,K)=Z(4,JO)
      DO 7608 L=2,3
 7608 VLIST(L,K)=VLIST(L,JO)
      IF(J.EQ.5) GO TO 32
      J=J+1
      JO=J
      NEXT=NEXT+1
      GO TO 7602
C***RESET VALUES
 7550 CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9998
      IF(INSYM.EQ.ICONS) GO TO 7551
      IF(INSYM.EQ.ILEVS) GO TO 7552
      GO TO 9998
 7551 CALL CONSTS
      GO TO 7553
 7552 CALL LEVELS
 7553 IF(IDEL.EQ.1) GO TO 7[illegible]
      GO TO 755[illegible]
C***STORE SETUP
 6[illegible]51 CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9[illegible]
      IF(INSY[illegible].EQ.I[illegible]) GO TO 6352
      IF(INSY[illegible].EQ.I[illegible]) GO TO [illegible]
      GO TO 9[illegible]
 6352 JS[illegible]=1
      I[illegible]=[illegible]
      GO TO 7[illegible]
```

```
      6353 JUST[illegible]L=
           [illegible]
           DO 6354 J=1,[illegible]
      6354 [illegible](J)=1
           GO TO 7[illegible]99
C***SCALE DATA
      6301 CALL GETSYM
           IF(ITYPE.NE.1) GO TO 9998
           INAME=INSYM
           CALL GETSYM
           IF(ITYPE.EQ.2) GO TO 6302
           IF(INAME.EQ.IHALF .AND. INSYM.EQ.NO) HALF=BLANK
           IF(INAME.EQ.IHALF .AND. INSYM.EQ.YES) HALF=DOT
           IF(INAME.EQ.IONE .AND. INSYM.EQ.NO) ONE=BLANK
           IF(INAME.EQ.IONE .AND. INSYM.EQ.YES) ONE=DOT
           GO TO 6303
      6302 IF(INAME.NE.ISPACE) GO TO 9998
           SPACES=XNUM
      6303 IF(IDEL.EQ.1) GO TO 7099
           GO TO 6301
C***PAUSE DATA
      6310 IPASS=0
           DO 6311 I=1,6
           IF(IPASS.EQ.1) GO TO 6312
           CALL GETSYM
           IF(ITYPE.NE.2) GO TO 99[illegible]
           PAUSE(I)=XNUM
           IF(IDEL.EQ.1 .OR. I.EQ.5) IPASS=1
           GO TO 6311
      6312 PAUSE(I)=[illegible]
      6311 CONTINUE
           GO TO 7099
```

```
C***CHANGE TABLES                                                       1328
 7430 CALL GETSYM                                                       1329
      J=0                                                               1330
      DO 7431 I=1,24
 7431 IF(INSYM.EQ.TNAME(I)) GO TO 7432                                  1331
      GO TO 9998                                                        1332
 7432 J=J+1                                                             1333
      CALL GETSYM                                                       1334
      IF(I.EQ.1) TCBB(J)=XNUM
      IF(1.EQ.2)TCAD(J)=XNUM
      IF(I.EQ.3)TCBF(J)=XNUM
      IF(I.EQ.4)TCBH(J)=XNUM
      IF(I.EQ.5)TCBJ(J)=XNUM
      IF(I.EQ.6)TCBS(J)=XNUM
      IF(I.EQ.7)TCCC(J)=XNUM
      IF(I.EQ.8)TCCD(J)=XNUM
      IF(I.EQ.9)TCCF(J)=XNUM
      IF(I.EQ.10)TCCJ(J)=XNUM
      IF(I.EQ.11)TCCM(J)=XNUM
      IF(I.EQ.12)TCCP(J)=XNUM
      IF(I.EQ.13)TCCR(J)=XNUM
      IF(I.EQ.14)TCCX(J)=XNUM
      IF(I.EQ.15)TCDB(J)=XNUM
      IF(I.EQ.16)TCFA(J)=XNUM
      IF(I.EQ.17)TCFB(J)=XNUM
      IF(I.EQ.18)TCFC(J)=XNUM
      IF(I.EQ.19)TCFJ(J)=XNUM
      IF(I.EQ.20)TCHC(J)=XNUM
      IF(I.EQ.21)TCJM(J)=XNUM
      IF(I.EQ.22)FACT(J)=XNUM
      IF(I.EQ.23)FEOT(J)=XNUM
      IF(I.EQ.24)TCFH(J)=XNUM
```

```
      IF(I.EQ.25)ITPL(J)=XNUM
      IF(I.EQ.26)SAAT(J)=XNUM
      IF(ITYPE.EQ.9)GO TO 7499
      GO TO 7432
C***MISC INPUT DATA
 6201 CALL GETSYM
      IF(ITYPE.NE.1) GO TO 9998
      INAME=INSYM
      CALL GETSYM
      IF(ITYPE.NE.2) GO TO 9998
      INSYM=INAME
      DO 6205 I=1,NOCONS
 6205 IF(INSYM.EQ.CONST(I)) GO TO 6206
      DO 6207 I=1,NOLEVS
 6207 IF(INSYM.EQ.ILEVLS(I)) GO TO 6208
      IF(JSTORE.EQ.1) GO TO 6221
      IF(INSYM.EQ.IDT) GO TO 6209
      IF(INSYM.EQ.ISTART) GO TO 6202
      IF(INSYM.EQ.IPLTPD) GO TO 6203
      IF(INSYM.EQ.NYRS) GO TO 6204
      GO TO 9998
 6221 WRITE(6,6222) INSYM
 6222 FORMAT(1X,A6,- CANNOT BE CHANGED WHILE STORE IS ON.-)
      GO TO 6210
 6202 TIME =XNUM
      GO TO 6210
 6203 PLTPD=XNUM
      GO TO 6210
 6204 NOYRS=XNUM
      GO TO 6210
 6206 Y(I)=XNUM
      GO TO 6210
```

```
  6200 IVALS(I)=X[illegible]
       GO TO 6210
  6209 DT=X[illegible]
  6210 IF(I[illegible]L.EQ.1) GO TO 7[illegible]9
       GO TO 6201
  7055 DO 7056 I=1,5
       Z(3,I)=VLIST(3,I)
       Z(4,I)=VLIST(5,I)
  7056 SVC(I)=VLIST(1,I)
C***INITIAL VALUES
    16 FORMAT( )
       INP=1
       IF(PLTPD.LE.DT) GO TO 6241
       INP=PLTPD/DT
       IF(MOD(PLTPD,DT).GT.0) INP=INP+1
  6241 MAX=(NOYRS/DT)+1
       IF(MOD(NOYRS,DT).GT.0) MAX=MAX+1
       TMS=INP*DT
       IFLG=0
       DO 562 I=1,5
       NPTS=((MAX-1)/INP)+1
       Z(1,I)=NPTS
   562 Z(2,I)=TMS
       DO 61 I=1,2
    61 WRITE(6,16)
       TIME=TIME[illegible]
       CALL MODEL(TIME,DT,TMS,[illegible],VLIST,Z,[illegible])
    32 DO 72 I=1,5
       JJ=ABS(VLIST(2,I))
       IF(JJ.NE.I) GO TO 73
       DO 75 J=J[illegible],5
       IF(IFIX([illegible](VLIST(2,J))).[illegible].J[illegible]) GO TO 75
```

```
      Z(3,J)=Z(3,J)
   7[illegible] CONTINUE
   73 J[illegible]=[illegible](4,I))
      IF(J[illegible].NE.I) GO TO 72
      DO 51 J=J[illegible],[illegible]
      IF(IFIX([illegible]LIST(4,J))).[illegible].J[illegible]) GO TO [illegible]
      Z(4,J)=Z(4,J[illegible])
   51 CONTINUE
   72 CONTINUE
      WRITE(6,33)
   33 FORMAT(1X,-RUN-,4X,-LOWER-,4X,-UPPER-,3X,-SYM-,3X,
     +-NAME-,2X,-DESCRIPTION-)
      DO 34 K=1,5
      IF(SMC(K).EQ.0) GO TO 34
      KM=SMC(K)
      WRITE(6,35) KRUN(K),Z(3,K),Z(4,K),ISY(K),(NAME(J,KM),J=1,6)
   35 FORMAT(1X,I3,2F9.3,4X,A2,3X,6A6)
   34 CONTINUE
      DO 62 L=1,2
   62 WRITE(6,16)
      IF(IFLG.EQ.1) GO TO 6
      IFLG=1
      ICOUNT=[illegible]
      DO 6101 I=2,4,2
      IF(I.EQ.2)KK=3
      IF(I.EQ.4)KK=4
      DO 6102 J=1,5
      IF(VLIST(I,J).GT.-.5) GO TO 6102
      IF(-IFIX(VLIST(I,J)).[illegible].J) GO TO 6102
      K=VLIST(I,J)
      IF(I.EQ.2) WRITE(6,6103) K
 6103 FORMAT(- [illegible] L[illegible]-,I[illegible],- -)
```

```
      IF(I.EQ.4) WRITE(6,6104) K
 6104 FORMAT(- NEW UPPER-,I3,-/-)
      NEXT=73
      CALL GETSYM
      IF(ITYPE.EQ.0) GO TO 6102
      ICOUNT=1
      Z(KK,J)=XNUM
 6102 CONTINUE
 6101 CONTINUE
      IF(ICOUNT.EQ.0) GO TO 6000
      DO 83 II=1,2
   83 WRITE(6,16)
      GO TO 32
 6000 CONTINUE
      IFLG=0
      TIME=TIMEP
      CALL PLOT(TIME,SWC,Z,ISM,HALF,ONE,SPACES,PAUSE)
      DO 63 I=1,2
   63 WRITE(6,16)
      IF(JSTORE.EQ.0) GO TO 1000
C***STORE RUN DATA
 7109 IRUN=IRUN+1
      IF(IRUN.GT.10) GO TO 9998
      DO 7110 J=1,5
      KRUN(J)=IRUN+1
      IVAR(IRUN,J)=VLIST(1,J)
      DO 7110 I=1,254
 7110 SZ(IRUN,I,J)=Z(I,J)
      GO TO 1000
      END
```

```
      SUBROUTINE PLOT(TIME,SWC,Z,ISUM,HALF,ONE,SPACES,PAUSE)           1459
      DIMENSION SWC(5),Z(254,5),ISUM(5)                                1460
      DIMENSION LINE(60)                                               1461
      DIMENSION PAUSE(6),ISUM(5),IPLT(5)                               1462
      INTEGER BLANK,DOT,SWC,NO,HALF,ONE,SPACES,PAUSE                   1463
      DATA EPS/.000001/                                                1464
      DATA BLANK/- -/                                                  1465
      DATA DOT/-.-/                                                    1466
      DATA NO/-NO-/                                                    1467
      DO 610 I=1,5                                                     1468
  610 ISUM(I)=0                                                        1469
      NPTS=Z(1,1)+4                                                    1470
      TMS=Z(2,1)                                                       1471
      STIME=PAUSE(1)                                                   1472
      IJ=1                                                             1473
      TIME=TIME-TMS                                                    1474
      DO 10 II=5,NPTS                                                  1475
      DO 612 I=1,5                                                     1476
  612 IPLT(I)=0                                                        1477
      TIME=TIME+TMS                                                    1478
      JJ=II-5                                                          1479
      IF(MOD(JJ,SPACES).NE.0) GO TO 13                                 1480
      DO 12 I=1,60                                                     1481
   12 LINE(I)=DOT                                                      1482
      GO TO 14                                                         1483
   13 DO 1 I=1,60                                                      1484
    1 LINE(I)=BLANK                                                    1485
      LINE(30)=HALF                                                    1486
      LINE(60)=ONE                                                     1487
   14 DO 1000 J=1,5                                                    1488
      MAX=-1                                                           1489
      DO 611 K=1,5                                                     1490
```

```
      IF(IPLT(K).EQ.1) GO TO 611
      IF(MAX.GT.ISUM(K)) GO TO 611
      MAX=ISUM(K)
      I=K
  611 CONTINUE
      IPLT(I)=1
      IF(SAC(I).EQ.0) GO TO 1000
      IF(Z(4,I)-Z(3,I).LE.EPS) GO TO 1000
      IF(Z(II,I).LT.Z(3,I)) GO TO 1000
      IF(Z(II,I).GT.Z(4,I)) GO TO 1000
      K=((Z(II,I)-Z(3,I))/(Z(4,I)-Z(3,I)))*60.0+1
      IF(K.GT.60) K=60
      IF(K.LT.1) K=1
  400 IF(LINE(K).NE.BLANK) GO TO 402
  401 LINE(K)=ISM(I)
      ISUM(I)=0
      GO TO 1000
  402 IF(LINE(K).EQ.DOT) GO TO 401
      ISUM(I)=ISUM(I)+1
 1000 CONTINUE
      PTIME=IFIX(TIME)
      WRITE(6,9001) PTIME,LINE
      IF(TIME.LT.STIME) GO TO 10
      IF((TIME-TMS).LT.PAUSE(IJ) .AND. PAUSE(IJ).LE.TIME) GO TO 55
      GO TO 10
   55 READ(5,111)IB[illegible]
  111 FORMAT(A1)
      IJ=IJ+1
   10 CONTINUE
      RETURN
 9001 FORMAT(1X,F5.0,60A1)
      END
```

```
      SUBROUTINE PRINT(TIME,ISWC)                                       1523
      DIMENSION A(126,20),R(25,20),F(46,20)
      DIMENSION IA(126),IR(25),IF(46)
      DIMENSION PRNT(20)                                                1524
      COMMON /PTVRS/ IA,IR,IF                                           1525
      DIMENSION NAMES(197)
      COMMON /PNAMS/ NAMES                                              1526
      COMMON /PRNT/                                                     1527
     +AAA,AAB,AAC,AAD,AAG,BAA,BAB,BAC,BAD,BAE,BAH,BAJ,BAK,BAL,BAM,
     +CBA,CBB,CBD,CBF,CBH,CBJ,CBM,CBN,CBP,CBQ,CBR,CBS,CCA,CCB,CCC,
     +CCD,CCG,CCH,CCJ,CCL,CCM,CCP,CCR,CCV,CCW,CCX,CCY,CDA,CDB,CDC,
     +CDD,CEA,CFA,CFB,CFC,CFE,CFH,CFJ,CFK,CFN,CFP,CHA,CHC,CHE,CJA,
     +CJB,CJD,CKA,CLA,CLC,CLD,CLE,CLF,CLG,CLH,CNA,CQA,CXC,CXD,CXG,
     +CXJ,CXK,CXL,CXR,CXS,CXT,RCD,F1,FAA,FAB,FAC,FAD,FAE,FAF,FAG,
     +FAK,FBA,FBB,FBC,FBD,FBE,FBF,FBG,FBK,FEA,FEB,FPA,FPB,FPC,FPD,
     +FPE,FPH,FPJ,FPK,FPL,FPP,FPQ,PAA,PAB,PAC,PAD,PAG,PAH,SAA,SAC,
     +X1,X2,X4,X5,X6,X8,
     +AA,AB,AC,BA,BB,BC,CA,CB,CC,CD,CE,CF,CG,
     +CH,CJ,CK,CL,CM,CN,CP,CQ,PA,PB,PC,SA,
     +A1,A2,AAF,B1,B2,BAF,BAG,
     +C1,C2,C3,C4,C5,C6,CAA,CBC,CBE,CBG,
     +CBL,CCE,CCF,CCK,CCN,CCQ,CCS,CCZ,
     +CEB,CFG,CFL,CFM,CHB,CHD,CJC,
     +FAH,FAJ,FBH,FBJ,FEC,FPM,FPN,
     +P1,P2,PAE,PAF,PAJ,S1,SAB
      DATA NAMES/-AAA-,-AAB-,-AAC-,-AAD-,-AAG-,-BAA-,-BAB-,-BAC-,-BAD-, 1528
     +-BAE-,-BAH-,-BAJ-,-BAK-,-BAL-,-BAM-,-CBA-,-CBB-,-CBD-,-CBF-,
     +-CBH-,-CBJ-,-CBM-,-CBN-,-CBP-,-CBQ-,-CBR-,-CBS-,-CCA-,-CCB-,
     +-CCC-,-CCD-,-CCG-,-CCH-,-CCJ-,-CCL-,-CCM-,-CCP-,-CCR-,-CCV-,
     +-CCW-,-CCX-,-CCY-,-CDA-,-CDB-,-CDC-,-CDD-,-CEA-,-CFA-,-CFB-,
     +-CFC-,-CFE-,-CFH-,-CFJ-,-CFK-,-CFN-,-CFP-,-CHA-,-CHC-,-CHE-,
     +-CJA-,-CJB-,-CJD-,-CKA-,-CLA-,-CLC-,-CLD-,-CLE-,-CLF-,-CLG-,
```

```
     +-CLH-,-CNA-,-CQA-,-CXC-,-CXD-,-CXG-,-CXJ-,-CXK-,-CXL-,-CXR-,
     +-CXS-,-CXT-,-RCD-,-F1-,-FAA-,-FAB-,-FAC-,-FAD-,-FAE-,-FAF-,
     +-FAG-,-FAK-,-FBA-,-FBB-,-FBC-,-FBD-,-FBE-,-FBF-,-FBG-,-FBK-,
     +-FEA-,-FEB-,-FPA-,-FPB-,-FPC-,-FPD-,-FPE-,-FPH-,-FPJ-,-FPK-,
     +-FPL-,-FPP-,-FPQ-,-PAA-,-PAB-,-PAC-,-PAD-,-PAG-,-PAH-,-SAA-,
     +-SAC-,-X1-,-X2-,-X4-,-X5-,-X6-,-X8-,-AA-,-AB-,-AC-,-BA-,-BB-,
     +-BC-,-CA-,-CB-,-CC-,-CD-,-CE-,-CF-,-CG-,-CH-,-CJ-,-CK-,-CL-,
     +-CM-,-CN-,-CP-,-CQ-,-PA-,-PB-,-PC-,-SA-,-A1-,-A2-,-AAF-,-B1-,
     +-B2-,-BAF-,-BAG-,-C1-,-C2-,-C3-,-C4-,-C5-,-C6-,-CAA-,-CBC-,-CBE-,
     +-CBG-,-CBL-,-CCE-,-CCF-,-CCK-,-CCN-,-CCQ-,-CCS-,-CCZ-,
     +-CEB-,-CFG-,-CFL-,-CFM-,-CHB-,-CHD-,-CJC-,
     +-FAH-,-FAJ-,-FBH-,-FBJ-,-FEC-,-FPM-,-FPN-,
     +-P1-,-P2-,-PAE-,-PAF-,-PAJ-,-S1-,-SAB-/
      IF(ISWC.EQ.1) GO TO 1                                             1529
      IF(J+1.GT.20)RETURN                                               1530
      J=J+1                                                             1531
      PRNT(J)=TIME                                                      1532
      A(1,J)=AAA
      A(2,J)=AAB
      A(3,J)=AAC
      A(4,J)=AAD
      A(5,J)=AAG
      A(6,J)=BAA
      A(7,J)=BAB
      A(8,J)=BAC
      A(9,J)=BAD
      A(10,J)=BAE
      A(11,J)=BAH
      A(12,J)=BAJ
      A(13,J)=BAK
      A(14,J)=BAL
      A(15,J)=BAM
```

```
      A(16,J)=CBA
      A(17,J)=CBB
      A(18,J)=CBD
      A(19,J)=CBF
      A(20,J)=CBH
      A(21,J)=CBJ
      A(22,J)=CBM
      A(23,J)=CBN
      A(24,J)=CBP
      A(25,J)=CBQ
      A(26,J)=CBR
      A(27,J)=CBS
      A(28,J)=CCA
      A(29,J)=CCB
      A(30,J)=CCC
      A(31,J)=CCD
      A(32,J)=CCQ
      A(33,J)=CCH
      A(34,J)=CCJ
      A(35,J)=CCL
      A(36,J)=CCM
      A(37,J)=CCP
      A(38,J)=CCR
      A(39,J)=CCV
      A(40,J)=CCW
      A(41,J)=CCX
      A(42,J)=CCY
      A(43,J)=CDA
      A(44,J)=CDB
      A(45,J)=CDC
      A(46,J)=CDD
      A(47,J)=CEA
```

```
      A(48,J)=CFA
      A(49,J)=CFB
      A(50,J)=CFC
      A(51,J)=CFL
      A(52,J)=CFH
      A(53,J)=CFJ
      A(54,J)=CFK
      A(55,J)=CFN
      A(56,J)=CFP
      A(57,J)=CHA
      A(58,J)=CHC
      A(59,J)=CHE
      A(60,J)=CJA
      A(61,J)=CJB
      A(62,J)=CJD
      A(63,J)=CKA
      A(64,J)=CLA
      A(65,J)=CLC
      A(66,J)=CLD
      A(67,J)=CLE
      A(68,J)=CLF
      A(69,J)=CLG
      A(70,J)=CLH
      A(71,J)=CNA
      A(72,J)=COA
      A(73,J)=CXC
      A(74,J)=CXD
      A(75,J)=CXG
      A(76,J)=CXJ
      A(77,J)=CXK
      A(78,J)=CXL
      A(79,J)=CXR
```

```
      A(80,J)=CX[illegible]
      A(81,J)=CXT
      A(82,J)=RCD
      A(83,J)=FI
      A(84,J)=FAA
      A(85,J)=FAB
      A(86,J)=FAC
      A(87,J)=FAD
      A(88,J)=FAE
      A(89,J)=FAF
      A(90,J)=FAG
      A(91,J)=FAK
      A(92,J)=FBA
      A(93,J)=FBB
      A(94,J)=FBC
      A(95,J)=FBD
      A(96,J)=FBE
      A(97,J)=FBF
      A(98,J)=FBG
      A(99,J)=FBK
      A(100,J)=FEA
      A(101,J)=FEB
      A(102,J)=FPA
      A(103,J)=FPB
      A(104,J)=FPC
      A(105,J)=FPD
      A(106,J)=FPE
      A(107,J)=FPH
      A(108,J)=FPJ
      A(109,J)=FPK
      A(110,J)=FPL
      A(111,J)=FPP
```

```
      A(112,J)=[illegible]
      A(113,J)=PAA
      A(114,J)=PAB
      A(115,J)=PAC
      A(116,J)=PAD
      A(117,J)=PAG
      A(118,J)=PAH
      A(119,J)=SAA
      A(120,J)=SAC
      A(121,J)=X1
      A(122,J)=X2
      A(123,J)=X4
      A(124,J)=X5
      A(125,J)=X6
      A(126,J)=X8
      R(1,J)=AA
      R(2,J)=AB
      R(3,J)=AC
      R(4,J)=BA
      R(5,J)=BB
      R(6,J)=BC
      R(7,J)=CA
      R(8,J)=CB
      R(9,J)=CC
      R(10,J)=CD
      R(11,J)=CE
      R(12,J)=CF
      R(13,J)=CG
      R(14,J)=CH
      R(15,J)=CJ
      R(16,J)=CK
      R(17,J)=CL
```

```
      R(18,J)=CY
      R(19,J)=CN
      R(20,J)=CP
      R(21,J)=C[illegible]
      R(22,J)=[illegible]
      R([illegible],J)=[illegible]
      R(24,J)=PC
      R(25,J)=SA
      F(1,J)=A1
      F(2,J)=A2
      F(3,J)=AAF
      F(4,J)=B1
      F(5,J)=B2
      F(6,J)=BAF
      F(7,J)=BAG
      F(8,J)=C1
      F(9,J)=C2
      F(10,J)=C3
      F(11,J)=C4
      F(12,J)=C5
      F(13,J)=C6
      F(14,J)=CAA
      F(15,J)=CBC
      F(16,J)=CBE
      F(17,J)=CBG
      F(18,J)=CBL
      F(19,J)=CCE
      F(20,J)=CCF
      F(21,J)=CCK
      F(22,J)=CCN
      F(23,J)=CCQ
      F(24,J)=CCS
```

```
      F(25,J)=CCZ
      F(26,J)=CEB
      F(27,J)=CFG
      F(28,J)=CFL
      F(29,J)=CFM
      F(30,J)=CHR
      F(31,J)=CHD
      F(32,J)=CJC
      F(33,J)=FAH
      F(34,J)=FAJ
      F(35,J)=FBH
      F(36,J)=FBJ
      F(37,J)=FEC
      F(38,J)=FPM
      F(39,J)=FPN
      F(40,J)=P1
      F(41,J)=P2
      F(42,J)=PAF
      F(43,J)=PAF
      F(44,J)=PAJ
      F(45,J)=S1
      F(46,J)=SAB
      RETURN                                                          1533
    1 WRITE(6,76)(PRNT(I),I=1,J)                                      1534
      DO 2 I=1,126
      IF(IA(I).EQ.1) WRITE(6,77)NAMES(I),(A(I,K),K=1,J)               1535
    2 CONTINUE                                                        1536
      DO 3 I=1,25
      IF(IR(I).EQ.1) WRITE(6,77) NAMES(I+76),(R(I,K),K=1,J)           1537
    3 CONTINUE                                                        1538
      DO 4 I=1,46
      IF(IF(I).EQ.1) WRITE(6,77) NAMES(I+106),(F(I,K),K=1,J)          1539
    4 CONTINUE                                                        1540
      RETURN                                                          1541
   77 FORMAT(1X,A6,5E10.5,3(/,7X,5E10.5))                             1542
   76 FORMAT(/,4(7X,5F10.2,/))                                        1543
      END                                                             1544
```

```
      SUBROUTINE GETSYM                                        1545
      DIMENSION IA(72)                                         1546
      COMMON /GETCNT/ ISYM,ITYPE,XNUM,IDEL,NEXT ,NEXT          1547
      DATA ISTOP/-STOP-/                                       1548
      LENGTH=0                                                 1549
      XNUM=0.0                                                 1550
      NEXT0=NEXT                                               1551
      IF(ITYPE.NE.-1) ITYPE=0                                  1552
      IDEC=-50                                                 1553
      ISIGN=0                                                  1554
      IF (NEXT .LE. 72) GO TO 10                               1555
  620 NEXT=1                                                   1556
      NEXT0=NEXT                                               1557
      READ(5,1) IA                                             1558
    1 FORMAT(72A1)                                             1559
   10 DO 100 I=NEXT,72                                         1560
      IF(ITYPE.EQ.-1) GO TO 47                                 1561
      IF (IA(I) .NE. -+-) GO TO  48                            1562
      IF (ITYPE .NE. 0) GO TO 891                              1563
      ISIGN=1                                                  1564
      ITYPE=2                                                  1565
      GO TO 100                                                1566
   43 IF (IA(I) .NE. ---) GO TO 49                             1567
      IF (ITYPE .NE. 0) GO TO 892                              1568
      ISIGN=-1                                                 1569
      ITYPE=2                                                  1570
      GO TO 100                                                1571
   49 IF (IA(I) .EQ. -0-) GO TO 200                            1572
      IF (IA(I) .EQ. -1-) GO TO 200                            1573
      IF (IA(I) .EQ. -2-) GO TO 200                            1574
      IF (IA(I) .EQ. -3-) GO TO 200                            1575
      IF (IA(I) .EQ. -4-) GO TO 200                            1576
```

```
      IF (IA(I) .EQ. -5-) GO TO 200                                     1577
      IF (IA(I) .EQ. -6-) GO TO 200                                     1578
      IF (IA(I) .EQ. -7-) GO TO 200                                     1579
      IF (IA(I) .EQ. -8-) GO TO 200                                     1580
      IF (IA(I) .EQ. -9-) GO TO 200                                     1581
   47 IF (IA(I) .EQ. - -) GO TO 300                                     1582
      IF(IA(I) .EQ. -)-) GO TO 1200                                     1583
      IF(IA(I) .EQ. -*-) GO TO 1300                                     1584
      IF (IA(I) .EQ. -(-) GO TO 400                                     1585
      IF(IA(I) .EQ. -.-) GO TO 501                                      1586
      IF(IA(I) .EQ.'-)-) GO TO 500                                      1587
      IF (IA(I) .EQ. -,-) GO TO 600                                     1588
      IF(IA(I) .EQ. -/-) GO TO 700                                      1589
      IF(IA(I) .EQ. -=-) GO TO 1100                                     1590
      IF (ITYPE .EQ. 2) GO TO 800                                       1591
   50 IF (LENGTH .EQ. 5) GO TO 100                                      1592
      FLD(LENGTH*6,6,INSYM) = FLD(0,6,IA(I))                            1593
      LENGTH=LENGTH+1                                                   1594
      IF(ITYPE.NE.-1) ITYPE=1                                           1595
      GO TO 100                                                         1596
  200 XNUM=FLD(0,6,IA(I))-48+XNUM*10.0                                  1597
      IDEC=IDEC+1                                                       1598
      ITYPE=2                                                           1599
      GO TO 100                                                         1600
  300 IDEL=1                                                            1601
      GO TO 900                                                         1602
  400 IDEL=2                                                            1603
      GO TO 900                                                         1604
  500 IDEL=3                                                            1605
      GO TO 900                                                         1606
  501 IDEC=0                                                            1607
      I=I+1                                                             1608
```

```
          GO TO 100
    [illegible] IDEL=5
          GO TO 900
      700 IDEL=8
          GO TO 900
     1100 IDEL=10
          GO TO 900
     1200 IDEL=12
          GO TO 900
     1300 IDEL=13
          GO TO 900
  800     WRITE(6,2) IA(I),I
  2       FORMAT(- IMPROPER CHARACTER -,A1,- IN COLUMN -,I2)
          WRITE(6,3)
  3       FORMAT(- RETYPE THE LINE-)
          IDEL=9
          GO TO 900
  891     IDEL=6
          GO TO 901
  892     IDEL=7
          GO TO 901
      100 CONTINUE
          GO TO 620
  900     I=I+1
  901     IF (ITYPE .EQ. 1) GO TO 950
          IF(ITYPE .EQ. -1) GO TO 950
          IF (IDEL .LT. 0) GO TO 920
          DO 910 J=1,IDEL
  910     XNUM=XNUM/10.0
  920     IF (ISIGN .LT. 0) XNUM=-XNUM
          GO TO 999
  950     CONTINUE
```

```
      DO 960 J=LENGTH,5
      FLD(6*J,6,INSYM)=- -
  960 CONTINUE
  999 NEXT=1
      IF(INSYM .EQ. ISTOP) STOP
      RETURN
      END
```

```
SUBROUTINE NAMES(NAME,CONST,LEVELS)                          1648
DIMENSION NAME(6,26),CONST(96)
DIMENSION NAME1(6,18),NAME2(6,8)
DIMENSION ICONS(96),ILEVELS(46)
DIMENSION LEVELS(46)
INTEGER CONST                                                1649
DATA NAME1/                                                  1650
+-S1 -,-COMMUNITY CONCERN                  -,
+-F1 -,-TOTAL ANNUAL CJS BUDGET            -,
+-CB -,-NET FLOW TO CRIME                  -,
+-X6 -,-CRIM AT LRG/PERSON IMPRISONED -,
+-X2 -,-CRIME RATE (PER 100,000)           -,
+-CCB-,-CRIMINALS AT LARGE                 -,
+-CCA-,-REPORTED CRIMES                    -,
+-P2 -,-POLICE OFFICERS                    -,
+-X4 -,-FRACT REP CRIMES PUNISHED          -,
+-CC -,-ARREST RATE                        -,
+-FPA-,-TOTAL ANNUAL POLICE BUDGET         -,
+-CCL-,-REP CRIMES/POLICE OFFICER          -,
+-CCY-,-POLICE BUDGET/OFFICER              -,
+-X8 -,-ARRESTS PER OFFICER PER YEAR       -,
+-CFC-,-FRACTION TRIALS                    -,
+-U2 -,-PROSECUTORS                        -,
+-C3 -,-PERSONS IN CJS                     -,
+-C6 -,-PERSONS ON PROBATION               -/
DATA NAME2/                                                  1651
+-CFK-,-OPEN CASES/PROSECUTOR              -,
+-FBA-,-TOTAL ANNUAL COURTS BUDGET         -,
+-X5 -,-FRACTION REP CRIMES PUNISHED       -,
+-C4 -,-PERSONS IN PRISON                  -,
+-A2 -,-PRISON CAPACITY                    -,
+-CJU-,-PERCENT PRISON CAP IN USE          -,
```

```
     +-FAA-,-TOTAL ANNUAL CONNECTIONS BUDGT-,
     +-C5 -,-PERSONS ON PARCLE                -/
      DATA ICONS/-AAAA-,-AACA-,-AADA-,-AAFD-,-ABXA-,-ACXA-,        1652
     +-BAAA-,-BACA-,-BAFB-,-BAFD-,-BAGD-,-BAMA-,-BBXA-,
     +-BCXA-,-CAAA-,-CABA-,-CBAA-,-CBCA-,-CBEA-,-CBEB-,
     +-CBGA-,-CBLA-,-CBQA-,-CCAB-,-CCLA-,-CCCA-,-CCDA-,
     +-CCEA-,-CCFA-,-CCKA-,-CCNA-,-CCQA-,-CCSA-,-CCZD-,
     +-CDBA-,-CEAA-,-CEBA-,-CFBA-,-CFCA-,-CFEA-,-CFFA-,
     +-CFGA-,-CFJA-,-CFLB-,-CFLD-,-CFMD-,-CGA-,-CHBA-,
     +-CHCA-,-CHDA-,-CJAA-,-CJCA-,-CKAA-,-CLAA-,-CLAD-,
     +-CNAA-,-CQAA-,-CQAD-,-FABA-,-FABC-,-FABE-,-FACA-,
     +-FALA-,-FAJD-,-FAKA-,-FAMD-,-FBBA-,-FBBC-,-FBCA-,
     +-FBEA-,-FBHD-,-FBJD-,-FBKA-,-FECD-,-FPDA-,-FPHA-,
     +-FPMD-,-FPND-,-FPPA-,-PABA-,-PACA-,-PAEA-,-PAFA-,
     +-PAGA-,-PBAA-,-PCAA-,-PEB-,-PEC-,-SAXD-,-SABD-,
     +-C3XJ-,-B2XJ-,-P2IA-,-C1IA-,-CCA1-,-S1XJ-/
      DATA ILEVLS/-A1-,-A2-,-AAF-,-B1-,-B2-,-BAF-,-BAG-,-C1-,-C2-,        1653
     +-C3-,-C4-,-C5-,-C6-,-CAA-,-CBC-,-CBE-,-CBG-,-CBL-,-CCE-,-CCF-,
     +-CCK-,-CCN-,-CCQ-,-CCS-,-CCZ-,-CEB-,-CFG-,-CFL-,-CFM-,-CHB-,
     +-CHD-,-CJC-,-FAH-,-FAJ-,-FBH-,-FBJ-,-FEC-,-FPM-,-FPN-,-P1-,
     +-P2-,-PAE-,-PAF-,-PAJ-,-S1-,-SAB-/
       DO 1 I=1,6                                                      1654
       DO 3 J=1,18
     3 NAME(I,J)=NAME1(I,J)                                            1655
       DO 4 J=1,8
     4 NAME(I,J+18)=NAME2(I,J)                                         1656
     1 CONTINUE                                                        1657
       DO 5 I=1,96
     5 CONST(I)=ICONS(I)                                               1658
       DO 6 I=1,46
     6 LEVELS(I)=ILEVLS(I)                                             1659
       RETURN                                                          1660
       END                                                             1661
```

```
      FUNCTION TABHL(TABLE,POINT,P0,P1,P2)                               1662
      DIMENSION TABLE(20)                                                1663
      DATA EPS/.00001/                                                   1664
      IF(POINT.LE.P0) GO TO 10                                           1665
      IF(POINT.GE.P1) GO TO 20                                           1666
      PT=1+(POINT-P0)/P2                                                 1667
      I=PT                                                               1668
      IF(PT-EPS.GT.FLOAT(I)) GO TO 30                                    1669
      TABHL=TABLE(I)                                                     1670
      GO TO 999                                                          1671
   30 TABHL=TABLE(I)+(TABLE(I+1)-TABLE(I))*(PT-I)                        1672
      GO TO 999                                                          1673
   10 TABHL=TABLE(1)                                                     1674
      GO TO 999                                                          1675
   20 MAX=1+(P1-P0)/P2                                                   1676
      TABHL=TABLE(MAX)                                                   1677
  999 RETURN                                                             1678
      END                                                                1679
```

```
      FUNCTION CLIP(A,B,C,D)                                             1680
      IF(C.GE.D)CLIP=A                                                   1681
      IF(C.LT.D)CLIP=B                                                   1682
      RETURN                                                             1683
      END                                                                1684
```

```
      SUBROUTINE TABLES                                              1685
      DIMENSION TCBB(6),TCBD(6),TCBF(6),TCBH(6),TCBJ(6),TCBS(8),
     +TCCC(7),TCCD(6),TCCF(5),TCCJ(6),TCCM(6),TCCP(5),TCCR(6),
     +TCCX(6),TCDB(8),TCEA(6),TCFB(5),TCFC(6),TCFJ(6),TCHC(5),
     +TCJB(5),FAGT(6),FBGT(6),TFEB(6),TFPL(6),SAAT(6)
      DIMENSION DCBB(6),DCBD(6),DCBF(6),DCBH(6),DCBJ(6),DCBS(8),
     +DCCC(7),DCCD(6),DCCF(5),DCCJ(6),DCCM(6),DCCP(5),DCCR(6),
     +DCCX(6),DCDB(8),DCEA(6),DCFB(5),DCFC(6),DCFJ(6),DCHC(5),
     +DCJB(5),DFAG(6),DFBG(6),DFEB(6),DFPL(6),DSAA(6)
      DIMENSION NAMES(26)
      COMMON /TNAME/ NAMES                                           1686
      COMMON /TABLE/ TCBB,TCBD,TCBF,TCBH,TCBJ,TCBS,TCCC,
     +TCCD,TCCF,TCCJ,TCCM,TCCP,TCCR,TCCX,TCDB,TCEA,TCFB,
     +TCFC,TCFJ,TCHC,TCJB,FAGT,FBGT,TFEB,TFPL,SAAT
      DATA NAMES/-TCBB-,-TCBD-,-TCBF-,-TCBH-,-TCBJ-,-TCBS-,-TCCC-,
     +-TCCD-,-TCCF-,-TCCJ-,-TCCM-,-TCCP-,-TCCR-,-TCCX-,-TCDB-,-TCEA-,
     +-TCFB-,-TCFC-,-TCFJ-,-TCHC-,-TCJB-,-FAGT-,-FBGT-,
     +-TFEB-,-TFPL-,-SAAT-/
      DATA DCBB/.85,1.,1.,1.,1.1,1.3/,
     +DCBD/.9,1.,1.05,1.2,1.4,1.7/,
     +DCBF/1.15,1.05,1.,.95,.9,.85/,
     +DCBH/1.3,1.2,1.1,1.,1.,.95/,
     +DCBJ/1.,1.,.95,.9,.82,.7/,
     +DCBS/1.,1.05,1.15,1.3,1.35,1.25,1.1,1./,
     +DCCC/.75,.9,1.,1.,1.3,1.6,2./,
     +DCCD/1.5,1.3,1.1,1.,1.,.8/,
     +DCCJ/1.,1.2,1.4,1.5,1.7,1.8/,
     +DCCF/1.,1.05,1.2,1.4,1.6/,
     +DCCM/1.2,1.05,1.,1.,.9,.7/,
     +DCCX/1.,1.05,1.15,1.2,1.25,1.3/,
     +DCCR/.7,.95,1.,1.05,1.2,1.4/,
     +DCCP/1.,.95,.9,.8,.7/,
```

```
     +DC[illegible]/2.5,2.,1.7,1.45,1.2,1.,.9,.7/,
     +DCFA/1.,1.,1.,1.,1.,1./,
     +DCHC/1.2,1.05,1.,.95,.8/,
      DATA DCFJ/1.,1.,.95,.9,.85,.7/,
     +DCFC/1.,1.,.95,.85,.75,.6/
     +DCFB/1.2,1.1,1.,1.,.9/,
     +DCJB/1.1,1.,1.,.95,.9/,
     +DFAG/.015,.03,.06,.12,.25,.5/,
     +DFBG/.04,.08,.12,.25,.4,.6/,
     +DFEB/0.0,.015,.03,.05,.075,.125/,
     +DFPL/.02,.05,.1,.2,.4,.7/,
     +DSAA/4.,6.,7.5,8.7,9.5,10./
      DO 5 I=1,5
      TCCF(I)=DCCF(I)
      TCCP(I)=DCCP(I)
      TCFB(I)=DCFB(I)
      TCHC(I)=DCHC(I)
      TCJB(I)=DCJB(I)
    5 CONTINUE
      DO 6 I=1,6
      TCBB(I)=DCBB(I)
      TCBD(I)=DCBD(I)
      TCBF(I)=DCBF(I)
      TCBH(I)=DCBH(I)
      TCBJ(I)=DCBJ(I)
      TCCD(I)=DCCD(I)
      TCCJ(I)=DCCJ(I)
      TCCM(I)=DCCM(I)
      TCCR(I)=DCCR(I)
      TCCX(I)=DCCX(I)
      TCEA(I)=DCEA(I)
      TCFC(I)=DCFC(I)
```

```
      TC[illegible]J(I)=DC[illegible]J(I)
      FAGT(I)=[illegible](I)
      F[illegible]T(I)=[illegible](I)
      TFF[illegible](I)=DFF[illegible](I)
      TFPL(I)=DFPL(I)
      SAAT(I)=DSAA(I)
    6 CONTINUE
      DO 7 I=1,7
      TCCC(I)=DCCC(I)
    7 CONTINUE
      DO 9 I=1,8
      TCBS(I)=DCBS(I)
      TCDB(I)=DCDB(I)
    9 CONTINUE
      RETURN                                                            1687
      END                                                               1688
```

```
      SUBROUTINE CONSTS                                          16[illegible]
      COMMON /Y/ AAAA,A[illegible],[illegible],AAF[illegible],A[illegible]XA,[illegible]C[illegible],[illegible]
     +BACA,[illegible],BAF[illegible],BA[illegible],[illegible],[illegible]XA,[illegible]CXA,CAAA,CABA,C[illegible]A,
     +C[illegible],C[illegible],[illegible],C[illegible]A,C[illegible]L[illegible],C[illegible]A,CCA[illegible],CCBA,CCCA,CCD[illegible],
     +CCEA,CCFA,CCKA,CCNA,CCQA,CCSA,CCZD,CDJA,CEAA,CF[illegible]A,
     +CFPA,CFCA,CFEA,CFFA,CF[illegible]A,CFJA,CFL[illegible],CFLD,CF[illegible]D,CGA,
     +CHBA,C[illegible]CA,CHDA,CJ[illegible]A,CJC[illegible],CKAA,CLAA,CLA[illegible],CNAA,COAA,
     +CQA[illegible],FA[illegible]A,FAPC,FA[illegible],FACA,FAE[illegible],FAJD,FAKA,FAMD,FB[illegible]A,
     +F[illegible]BC,F[illegible]CA,FBFA,FB[illegible],F[illegible]JD,F[illegible]KA,FLC[illegible],FPDA,FPHA,FP[illegible]D,
     +FPND,FPPA,PABA,PACA,PAEA,PAFA,PAGA,PBAA,PCAA,PEB,
     +PEC,SAXD,SABD,C3XJ,B2XJ,P2IA,C1IA,CCAI,S1XJ
      AAAA=15.
      AACA=6.
      AADA=1.1
      AAFD=2.
      ABXA=2.
      ACXA=15.
      BAAA=15.
      BACA=2.
      BAFB=1.
      BAFD=1.
      BAGD=3.
      BA[illegible]A=1.5
      BBXA=2.
      BCXA=10.
      CAAA=1.
      CABA=.01
      C[illegible]A=.0[illegible]3
      C[illegible]=1.
      C[illegible]A=1.
      C[illegible]=4[illegible].
      C[illegible]A=2.
```

```
C[illegible]L[illegible]=3.
C[illegible]A=5.
CC[illegible]=3.3
CC[illegible]A=.8
CCCA=2.
CCDA=5.
CC[illegible]A=1.
CC[illegible]=1.
CCKA=1.
CCNA=1.
CCUA=1.
CCSA=5.
CCZD=2.
CDSA=275.
CLAA=.35
CEBA=2.
CF[illegible]A=.[illegible]
CFCA=.2
CFEA=.1
CFFA=.25
CFGA=1.
CFJA=.1
CFLB=1.5
CFLD=2.
CF[illegible]=2.
C[illegible]A=4[illegible].
CHLA=5.
CHCA=4.
C[illegible]A=1.
CJBA=.05
CJCA=3.
CKAA=.9
```

```
[illegible]
[illegible]
[illegible]
[illegible]
CGA[illegible]=[illegible].
F[illegible]A=[illegible].
FA[illegible]C=5[illegible].
FA[illegible]=5[illegible].
FACA=20[illegible].
FAEA=.2
FAJD=.5
FAKA=0.
FAMD=2.
F[illegible]A=2[illegible].
F[illegible]C=15[illegible].
F[illegible]A=2[illegible].
[illegible]=.2
FB[illegible]D=2.
FBJD=.5
F[illegible]KA=[illegible].
FECD=3.
FP[illegible]A=0.
FP[illegible]=.2
FP[illegible]D=2.
[illegible]=.[illegible]
FPPA=.[illegible]
P[illegible]A=.2[illegible]
P[illegible]C[illegible]=[illegible].
P[illegible]=2.
PAFA=2.
PAGA=15.
P[illegible]A=1.
```

```
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      C3XJ=[illegible].
      B2XJ=8.
      F2IA=7[illegible].
      C1IA=50[illegible].
      CCAI=960[illegible].
      S1XJ=4.
      RETURN                                              1690
      END                                                 1691
```

```
      SUBROUTINE LEVELS                                          [illegible]
      DIMENSION TC[illegible](6),T[illegible](6),TC[illegible](6),TC[illegible](6),TC[illegible]J(6),TC[illegible](6),
     +TCCC(7),TCC[illegible](6),TCC[illegible](5),TCCJ(6),TC[illegible](6),TCCP(5),TCC[illegible](5),
     +TCCX(6),T[illegible](3),TC[illegible](6),TC[illegible](5),TCFC(6),TCFJ(6),TCHC(5),
     +TCJ[illegible],[illegible](6),[illegible](6),T[illegible](6),TPPL(6),SA[illegible](6)
      COMMON /[illegible]/ [illegible],TC[illegible],TCHF,TC[illegible],TC[illegible]J,TC[illegible],TCCC,TC[illegible],
     +TCCF,TCCJ,TCC[illegible],TCCP,TCCR,TCCX,TC[illegible],TCF[illegible],TCF[illegible],TCFC,
     +TCFJ,TCHC,TCJ[illegible],FAST,F[illegible]T,F[illegible],TPPL,S[illegible]
      COMMON /I/ A1,A2,AA[illegible],[illegible],[illegible],[illegible]AF,[illegible]AG,C[illegible],C2,C3,C4,C5,C6,
     +CAA,CBC,CBF,CBG,CBL,CCF,CC[illegible],CCK,CC[illegible],CCG,CCS,CCZ,CEB,
     +CFG,CFL,CFM,CHP,CH[illegible],CJC,FA[illegible],FAJ,FBH,FBJ,FEC,FPK,FPN,
     +P1,P2,PAE,PAF,PAJ,S1,SAB
      COMMON /Y/ AAAA,AACA,AAB[illegible],AAFB,ABXA,ACXA,BAAA,
     +BACA,BAFB,BAFD,[illegible]AG[illegible],[illegible],[illegible]X[illegible],[illegible]CXA,CAAA,CABA,CBAA,
     +CBCA,CBEA,CBFB,CBGA,CBLA,[illegible]B[illegible]A,CCAB,CCB[illegible],CCCA,CCDA,
     +CCEA,CCF[illegible],CCKA,CCNA,CC[illegible]A,CC[illegible],CCZB,CBB[illegible],CEAA,CEBA,
     +CFBA,CFCA,C[illegible]A,CFFA,CF[illegible]A,C[illegible]A,CFL[illegible],CF[illegible],CF[illegible],CGA,
     +CHBA,CHCA,CHDA,CJAA,CJCA,[illegible]KAA,CLAA,CLAD,C[illegible]A,CGAA,
     +CGAB,FABA,FABC,FA[illegible]F,FACA,FAFA,F[illegible]J[illegible],FAKA,FA[illegible],FBBA,
     +FBBC,F[illegible]C[illegible],FBEA,F[illegible],F[illegible]J[illegible],F[illegible]K[illegible],FEC[illegible],FPDA,FPHA,FPMD,
     +FPNB,FPPA,PABA,PACA,PAFA,PAFA,PAG[illegible],P[illegible]A,PCAA,PEB,
     +PEC,SAX[illegible],SA[illegible],C3XJ,B2XJ,P21A,C1IA,[illegible]A[illegible],S1XJ
      B2=B2XJ
      C3=C3XJ
      P2=P21A
      S1=S1XJ
      BAF=P2*CCA[illegible]
      BAG=[illegible]2/[illegible]CXA
      CXF=[illegible]AF/C[illegible]A
      [illegible]=[illegible]XA[illegible]
      [illegible]=[illegible]Y * [illegible]
      C6I[illegible]=.[illegible]2*C[illegible]
```

```
C6=C[illegible]
C4[illegible]=.[illegible]2*C[illegible]
C4I[illegible]=.[illegible]*C[illegible]*C[illegible]A
C5=C4IA*CL[illegible]/C[illegible]
C4=C4IA*C4[illegible]/(C[illegible]A+C[illegible]C[illegible])
C2AI=1./C[illegible]
C2=CCAI*C2[illegible]I+CC[illegible]*(-C3)+C[illegible]*(-C6)+CL[illegible]*(-C5)
C1J[illegible]=C1IA-C2-C3-C4-C5-C6
C1=C1IA-C2-C3-C4-C5-C6
A2=.95*C4
AAF=A2/ACXA
A1=ABXA*AAF
PNA=C2+CCBA*C3+CQAA*C6+CLAA*C5
CAA=C1+PNA
C[illegible]C=100.*PNA/C1
C[illegible]E=C[illegible]E[illegible]*CCCA
C[illegible]G=1.
CBL=300000./(C1+PNA)
CC=P2*CCAE
CCE=CFJA+CF[illegible]A*CFCA
CCF=1.
PAE=CCCA*P[illegible]
CCK=PAE/P2
CCN=CC[illegible]
CC[illegible]=1.
CCS=[illegible]1
YCZ=1./PCA[illegible]
CCZ=1.1*YCZ
CE[illegible]=1.
C[illegible]A=C3/(C[illegible]A*[illegible]2)
CF[illegible]=[illegible]AX1(C[illegible]A,.1)
CFL=.2*[illegible]2*C[illegible]
```

```
      CFM=C3/ 2
      [illegible]
      C[illegible]=1[illegible]•*C4/A[illegible]
      CJC=[illegible]
      [illegible]
      FA[illegible]=S1
      FAJ=0•
      FBH=S1
      FBJ=0•
      FEC=S1
      FPM=S1
      FPN=0•
      PAF=P2/PCAA
      P1=P3AA*PAF
      PAJ=CCAB
      SAB=1•
      RETURN                                                            1693
      END                                                               1694
```

```
      SUBROUTINE [illegible]L(F[illegible], T,[illegible], [illegible]C,VLIST,Z, [illegible]X)            1695
      DIMENSION ITI[illegible](12)                                             1696
      COMMON /TI[illegible]/ ITI[illegible],KTI[illegible]                                       1697
      COMMON /P[illegible]T/                                                   1698
     +AAA,AA[illegible],A[illegible]C,A[illegible]D,A[illegible]G,[illegible],[illegible],[illegible]C,[illegible],BAF,BAH,BAJ,B[illegible],BAL,BAM,
     +CBA,C[illegible],C[illegible],CBF,CBH,CBJ,C[illegible],C[illegible],CBP,CB[illegible],CBR,CBS,DCCA,DCCB,CCC,
     +CCD,CCG,CCH,CCJ,DCCL,CCM,CCP,CC[illegible],CCV,CC[illegible],CCX,DCCY,CD[illegible],CDB,CDC,
     +CDD,CEA,CFA,CFM,DCFC,CF[illegible],CF[illegible],CFJ,[illegible]CFK,CFN,CFP,CHA,CHC,CHF,CJA,
     +CJB,DCJD,CKA,CLA,CLC,CLD,CLE,CL[illegible],CLG,CLH,CMA,CQA,CXC,CXD,CXG,
     +CXJ,CXK,CXL,CXR,CXS,CXT,RCD,DF1,DFAA,FAB,FAC,FAD,F[illegible]E,FAF,FAG,
     +FAK,DF[illegible],FBB,FBC,F[illegible],F[illegible]E,F[illegible]F,F[illegible],F[illegible]K,FLA,FE[illegible],DFPA,FP[illegible],FPC,FPD,
     +FPE,FPH,FPJ,FPK,FPL,FPP,FPU,PAA,PAB,PAC,PAD,PAG,PAH,SAA,SAC,
     +DX1,DX2,DX4,DX5,DX6,DX8,
     +AA,AB,AC,BA,BB,BC,CA,DCB,[illegible]CC,CD,CE,CF,CG,
     +CH,CJ,CK,CL,CM,CN,CP,CQ,PA,PB,PC,SA,
     +A1,DA2,AAF,B1,DB2,BAF,BAG,C1,
     +C2,DC3,[illegible]C4,[illegible]C5,DC6,CAA,C[illegible],C[illegible],
     +CBG,CBL,CCE,CCF,CCK,CCN,CCQ,CCS,
     +CCZ,CEB,CFG,CFL,CFM,CHB,CHD,CJC,
     +FAH,FAJ,F[illegible]H,F[illegible]J,FEC,FP[illegible],FP[illegible],P1,
     +DP2,PAE,PAF,PAJ,DS1,SAB
      DIMENSION S[illegible]C(5),VLIST(5,[illegible]),Z(256,5),X(26)
      DIMENSION TC[illegible](6),TC[illegible](6),TCBF(6),TC[illegible](6),TCBJ(6),TC[illegible]S(8),
     +TCCC(7),TCCD(6),TCCF(5),TCCJ(6),TCC[illegible](6),TCCP(5),TCCR(6),
     +TCCX(6),TC[illegible](8),TCEA(5),TCF[illegible](5),TCFC(6),TCFJ(5),TC[illegible]C(5),
     +TCJB(5),FAGT(6),F[illegible]GT(6),TF[illegible](6),TFPL(6),SAAT(6)
      COMMON /TABLE/ TC[illegible],TCB[illegible],TC[illegible]F,TC[illegible],TC[illegible]J,TCBS,TCCC,
     +TCCD,TCCF,TCCJ,TCC[illegible],TCCP,TCC[illegible],TCCX,TCD[illegible],TC[illegible]A,TCF[illegible],
     +TCFC,TCFJ,TCHC,TCJ[illegible],FAGT,F[illegible]GT,TF[illegible],TFPL,SAAT
      INTEGER S[illegible]C                                                    1699
      COMMON /[illegible]/ J[illegible],K[illegible]                                              1700
      EQUIVALENCE (X(1),S1),(X(2),FT),(X(3),C[illegible]),(X(4),X[illegible]),
```

```
      +(X(5),A2),(X([illegible]),[illegible]),(X(7),[illegible]),(X([illegible]),[illegible]),(X([illegible]),X4),
      +(X([illegible]),[illegible]),(X(11),[illegible]),(X([illegible]),[illegible]),(X(13),CTY),
      +(X(14),X3),(X(15),[illegible]),(X([illegible]),[illegible]),(X(17),C[illegible]),
      +(X(18),C5),(X([illegible]),CFK),(X(20),[illegible]),(X(21),Y5),
      +(X(22),C[illegible]),(X(23),A2),(X(24),C[illegible]),(X(25),[illegible]),
      +(X(26),C5)
       COMMON /Y/ AAAA,AACA,AADA,[illegible],[illegible],ACXA,[illegible],
      + [illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],CP[illegible],
      +CBCA,C[illegible],[illegible],CBGA,C[illegible]LA,C[illegible],CCA[illegible],CCBA,CCCA,CCDA,
      +CCEA,CCFA,CCKA,CCNA,CCQA,CCSA,CCZJ,CDGA,CEAA,CEBA,
      +CFBA,CFCA,CFEA,CFFA,CFGA,CFJA,CFLB,CFLD,CF[illegible]D,CGA,
      +CHBA,CHCA,CHDA,CJAA,CJCA,CKZA,CLAA,CLAD,CNAA,COAA,
      +COAD,FABA,FABC,FABE,FACA,FAEA,FAJD,FAKA,FA[illegible]D,FBJA,
      +FBBC,FBCA,FBEA,FB[illegible],FBJD,FB[illegible]A,FBCJ,FPDA,FPHA,FP[illegible],
      +FPNJ,FPPA,PABA,PACA,PAEA,PAFA,PAGA,PBAA,PCAA,PEB,
      +PEC,SAK[illegible],SAED,C3XJ,C2XJ,P21A,C11A,CCA1,S1XJ
       COMMON /I/ IA1,IA2,IAA[illegible],I[illegible]1,I[illegible]2,IPAF,I[illegible],IC1,
      +IC2,IC3,IC4,IC5,IC6,ICAA,IC[illegible],I[illegible],ICJG,ICPL,
      +ICCE,ICCF,ICCK,ICCN,ICCQ,ICCS,ICCZ,ICFB,ICFG,ICFL,ICFM,
      +ICHB,ICHD,ICJC,IFAH,IFAJ,IF[illegible]H,IF[illegible]J,IFEC,IFP[illegible],IFPN,
      +IP1,IP2,IPAE,IPAF,IPAJ,IS1,ISA[illegible]
C***INITIAL VALUES                                                           1701
C***BEGIN COMPUTATION                                                        1702
       REAL IA1,IA2,IAAF,I[illegible]1,I[illegible]2,I[illegible]AF,I[illegible],IC1,
      +IC2,IC3,IC4,IC[illegible],IC6,I[illegible],I[illegible],I[illegible],IC[illegible],IC[illegible]L,
      +ICCE,ICCF,I[illegible],ICCN,ICC[illegible],I[illegible]CS,ICCZ,ICE[illegible],ICF[illegible],ICFL,IC[illegible],
      +ICH[illegible],ICH[illegible],ICJC,IF[illegible],IF[illegible]J,I[illegible],IF[illegible]J,IFEC,IFP[illegible],IFPN,
      +IP1,IP2,IPA[illegible],IPAF,IPAJ,I[illegible]1,I[illegible]
       FAHJ=2.
       A1=IA1
       A2=IA2
       AAF=IAAF
```

```
[illegible]1=I[illegible]1
[illegible]=I[illegible]2
[illegible]=I[illegible]
[illegible]=I[illegible]
C1=IC1
C2=IC2
C3=IC3
C4=IC4
C5=IC5
C6=IC6
CAA=ICAA
CBC=ICBC
CBE=ICBE
CBG=ICBG
CBL=ICBL
CCE=ICCE
CCF=ICCF
CCK=ICCK
CCN=ICCN
CCO=ICCO
CCS=ICCS
CCZ=ICCZ
CEG=ICEG
CFG=ICFG
CFL=ICFL
CFM=ICFM
CHR=ICHR
CHO=ICHO
CJC=ICJC
FAH=IFAH
FAJ=IFAJ
FBM=IFBM
```

```
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      P1=IP1
      P2=IP2
      PAE=IPAE
      PAF=IPAF
      PAJ=IPAJ
      S1=IS1
      SAB=ISAB
      PNA=C2+CCBA*C3+CDAA*C6+CLAA*C5
      C1JB=C1IA-C2-C3-C4-C5-C6
      TIME=TIME-DT                                                  1703
      KK=4                                                          1704
      DO 5000 INDEX=1,MAX                                           1705
      TIME=TIME+DT                                                  1706
C***AUXILIARY EQUATIONS                                             1707
      AAA=(-A2)/AAAA
      AAD=AADA*C4+ABX1*AAF
      AAC=AAF+(AAD-A2-A1)/AACA
      BAA=(-B2)/BAAA
      BAJ=B2*C3XJ/B2XJ
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
```

```
[illegible]
CCJ=TABHL(TCCJ,CC[illegible],1[illegible],4[illegible])
CC[illegible]=TABHL(TCC[illegible],CC[illegible],1.,2.)
CCP=TABHL(TCCP,CC[illegible],1.,5.,1.)
CC[illegible]=TABHL(TCC[illegible],CC[illegible],[illegible])
CCV=TABHL(TCCV,CC[illegible],1.,5.,1.)
CCX=TABHL(TCCX,CCZ,7[illegible],17[illegible],2[illegible])
CFK=C3/[illegible]2
CFP=CF[illegible]2XJ/C3XJ
CHA=C4/CHJ
CKA=CKAA
CHE=1.-CKA
CJB=TABHL(TCJB,CJC,6[illegible],14[illegible],20.)
CJD=1[illegible]*C4/A2
CLC=CCP*CC[illegible]
CLD=CCR*CCJ
CLE=CLD*CL[illegible]*C[illegible]X
CLG=CFC/CFCA
CLH=1.-CLAA
CNA=CNAA
CQA=1.-CQAA
CFJ=TABHL(TCFJ,CFP,[illegible],5.,1.)
CF[illegible]=TABHL(TCFC,CFP,[illegible],5.,1.)
CFC=CFCA*CF[illegible]
CXC=CFC/CFCA
CXD=TABHL(TC[illegible],CXC,[illegible],1.4,.2)
CXG=TABHL(TCC[illegible],CCE,[illegible],25,[illegible])
CXK=CF[illegible]*CF[illegible]*(-C[illegible])
CXL=CX[illegible]*CF[illegible]
CXR=TABHL(TC[illegible],CL[illegible],[illegible],2.5,[illegible])
CXS=TABHL(TC[illegible],C[illegible],6[illegible],14[illegible],20.)
CXT=TABHL(TC[illegible],C[illegible],6[illegible],14[illegible],2[illegible])
```

```
      FA[illegible]=F[illegible]*A2+FA[illegible]C*C4+F[illegible]*C5
      F[illegible]C=FA[illegible]*A[illegible]
      FAE=FAEA*FAC[illegible]*A2
      FAG=TA[illegible]L(FA[illegible],FAH,[illegible].,1[illegible].,2.)
      FAK=FAKA
      FBB=F[illegible]*[illegible]2+F[illegible]C*C6
      FBC=FBCA*[illegible]G
      F[illegible]=F[illegible]A*F[illegible]A*52
      FAF=FAG*(FAB+FAC)
      FAD=AMIN1(FAE,FAF)
      FAA=FAB+FAC+FAD+FAK
      FBG=TABHL(FBGT,FBH,0.,10.,2.)
      FBK=FBKA
      FEB=TABHL(TFEB,FEC,0.,10.,2.)
      FPC=PEB*PAF
      FPD=FPDA
      FPH=FPHA*PEB*P2
      FPL=TABHL(TFPL,FPM,0.,10.,2.)
      FPG=PEC*CCK*P2IA/(PNA*CCCA)
      PAB=PAE*PABA/PAJ
      PAD=PAB+PBAA*PAF
      PAG=(-P2)/PAGA
      PAH=(FPC+FPN)/PEB
      SAA=TABHL(SAAT,SAB,0.,20.,4.)
      CDB=CDBA*CXD
C***KILROY WAS HERE
      AAG=(FAC+FAJ)/FACA
      BAL=BAF/CDB
      BAK=(F[illegible]C+FBJ)/F[illegible]CA
      BAY=BA[illegible]A*CD[illegible]
      BAH=(C3-BAJ)/[illegible]
      BAL=AMAX1(BAH,[illegible].)
```

```
BAD=[illegible]X[illegible]*[illegible]+[illegible]+[illegible]AL
BAC=[illegible]+([illegible]-[illegible]-[illegible]1)/[illegible]C[illegible]
BAB=AMIN1(BAC,[illegible]AK)
AAB=AMIN1(AAC,AAG)
CBA=C[illegible]-C[illegible]*C[illegible]
CBP=100.*CCB/CI
CCD=CCDA*CXG*CCV
CXJ=TABFL(TCCC,CCD,0.,10.,2.)
CCC=CCCA*CXJ
CCA=CCB*CCC
CBM=CBEB*CCD*CCD/(CCDA*CCDA)
CBQ=CBM*CCC*CBQA
CCH=CCM*CCP*CCR
CCL=CCA/P2
CCW=CCH*CCX*CCJ
CDD=B2*CDB
CFH=C3/CXL
CDA=AMIN1(CDD,CFH)
CFB=CFBA*CXS
CFA=CFJA*CFJ+CFB*CFC
RCD=1.-CFA
CDC=CDA*RCD+CDA*RCD*(-CM[illegible])
CEA=CEA[illegible]*CXR
CFE=(CFB-CAK)/CFC
CHC=CHCA*CXT
CJA=CJA[illegible]*CJ[illegible]
CLA=C3/CDA
F[illegible]F=F[illegible]*(F[illegible]+F[illegible]C)
F[illegible]D=AMIN1(F[illegible],F[illegible]F)
F[illegible]A=F[illegible]+F[illegible]C+F[illegible]+F[illegible]K
FPP=PEC*FPPA+FP[illegible]-FPP[illegible]*FP[illegible]
PAC=PAF+(PAD-P2-P1)/PACA
```

```
      PNA=AMIN1(PAC,P  )
      FPU=FPP*P2
      FPJ=PL *PAC
      FPK=FPL*(FPB+FPC)
      FPE=AMIN1(FPH,FPK)
      FPA=FP +FPC+F  +FP
      FEA=FEE*(FPA+FAA+FBA)
      CLF=FEA/(C1+CCB)
      CCY=FPA/P2
      F1=FPA+FAA+FBA+FEA
      SAC=CCA*C1JB/(C1*PNA*CCCA)
C***SUPPLEMENTARY EQUATIONS                                        1708
      X1=C1+C2+C3+C5+C6
      X2=10 0 0.*CCA/X1
      X4=BAFE*BAF/PAE
      X5=CFLB*CFL/PAE
      X6=CC3/C4
C***RATE EQUATIONS                                                 1709
      AA=AMAX1(AAA,AAB)
      AB=A1/ABXA
      AC=A2/ACXA
      BA=AMAX1(BAA,BAB)
      BB=B1/BBXA
      BC=B2/BCXA
      CA=CAA*C/BA
      CB=C1*CCA*CBN
      CC=P2*CCAB*CC
      X8=CC/P2
      CD=CDC+CDC*(-C  )
      CL=CDC*CLA
      CE=C  *CEA
      CG=C4/C A
```

```
      C[illegible]=C[illegible]*[illegible]+C[illegible]*C[illegible](-[illegible])
      CJ=C[illegible]*C[illegible]*C[illegible]
      C[illegible]=C[illegible]*[illegible]
      CL=C5*CL[illegible]/CL[illegible]
      C[illegible]=(C5+C[illegible]*(-CL[illegible]))/CL[illegible]
      C[illegible]=C[illegible]*[illegible]C[illegible]*C[illegible]
      CP=(C6-C[illegible]*C6)/C[illegible]
      C[illegible]=C6*C[illegible]A/C[illegible]
      PA=AMAX1(PAA,PAG)
      PB=P1/P[illegible]AA
      PC=P2/PCAA
      SA=(SAA-S1)/SAXD
C***STORE PLOT DATA                                                   1710
      DX8=X8
      DCC=CC
      DCB=CB
      DCCA=CCA
      DCCB=CCB
      DCCL=CCL
      DCCY=CCY
      DCFC=CFC
      DCFK=CFK
      DCJ[illegible]=CJ[illegible]
      DF1=F1
      DFAA=FA[illegible]
      DF[illegible]A=F[illegible]
      DFPA=FPA
      DX2=X2
      DX4=X4
      DX5=X5
      DX6=X6
      DA2=A2
```

```
      DB2=B2
      DC3=C3
      DC4=C4
      DC5=C5
      DC6=C6
      DP2=P2
      DS1=S1
      IF(KTIME.EQ.0) GO TO 4555                                        1711
      DO 4551 I=1,20                                                   1712
 4551 IF(INDEX.EQ.ITIMES(I))CALL PRINT(TIME,0)                         1713
 4555 CONTINUE                                                         1714
      IF(JDEBUG.LE.INDEX .AND. INDEX.LE.KDEBUG)WRITE(6,1234)           1715
     +AAA,AAB,AAC,AAD,AAG,BAA,BAB,BAC,BAD,BAE,BAH,BAJ,BAK,BAL,BAM,
     +CBA,CBB,CBD,CBF,CBH,CBJ,CBM,CBN,CBP,CBQ,CBR,CBS,CCA,CCB,CCC,
     +CCD,CCG,CCH,CCJ,CCL,CCM,CCP,CCR,CCV,CCW,CCY,CCY,CDA,CDB,CDC,
     +CDD,CEA,CFA,CFB,CFC,CFE,CFH,CFJ,CFK,CFN,CFP,CHA,CHC,CHE,CJA,
     +CJB,CJD,CKA,CLA,CLC,CLD,CLE,CLF,CLG,CLH,CNA,CQA,CXC,CXD,CXG,
     +CXJ,CXK,CXL,CXR,CXS,CXT,RCD,FI,FAA,FAB,FAC,FAD,FAE,FAF,FAG,
     +FAK,FBA,FBB,FBC,FBD,FBE,FBF,FBG,FBK,FEA,FEB,FPA,FPB,FPC,FPD,
     +FPE,FPH,FPJ,FPK,FPL,FPP,FPQ,PAA,PAB,PAC,PAD,PAG,PAH,SAA,SAC,
     +X1,X2,X4,X5,X6,Xc,
     +AA,AB,AC,BA,BB,BC,CA,CB,CC,CD,CE,CF,CG,
     +CH,CJ,CK,CL,CM,CN,CP,CQ,PA,PB,PC,SA,
     +A1,A2,AAF,B1,B2,BAF,BAG,
     +C1,C2,C3,C4,C5,C6,CAA,CBC,CBH,CBG,
     +CBL,CCE,CCF,CCK,CCN,CCW,CCS,CCZ.
     +CEB,CFG,CFL,CFM,CHB,CHF,CJC,
     +FAH,FAJ,FEH,FBJ,FEC,FPF,FPN,
     +P1,P2,PAE,PAF,PAJ,S1,S2,
 1234 FORMAT(                                                          1716
     +,/,-   1-,4(5(1X,E12.5),/,4X),5(1X,E12.5),                       1717
     +,/,-  26-,4(5(1X,E12.5),/,4X),5(1X,E12.5),                       1718
```

```
     +,/,-  51-,4(5(1X,E10.5),/,4X),5(1X,E10.5),
     +,/,-  76-,4(5(1X,E10.5),/,4X),5(1X,E10.5),
     +,/,- 101-,4(5(1X,E10.5),/,4X),5(1X,E10.5),
     +,/,- 126-,4(5(1X,E10.5),/,4X),5(1X,E10.5),
     +,/,- 151-,4(5(1X,E10.5),/,4X),5(1X,E10.5),
     +,/,- 176-,4(5(1X,E10.5),/,4X),5(1X,E10.5))
      IF(MOD(TIME,PMS).GE.DT) GO TO 5001
      KK=KK+1
      DO 560 II=1,5
      IF(SWC(II).EQ.0) GO TO 560
      LL=SWC(II)
      KI=ABS(VLIST(2,II))
      IF(KI.EQ.0) GO TO 559
      Z(3,KI)=AMIN1(Z(3,KI),X(LL))
  559 KI=ABS(VLIST(4,II))
      IF(KI.EQ.0) GO TO 558
      Z(4,KI)=AMAX1(Z(4,KI),X(LL))
  558 Z(KK,II)=X(LL)
  560 CONTINUE
 5001 CONTINUE
C***LEVEL EQUATIONS
      A1=A1+DT*(AA-AB)
      A2=A2+DT*(AB-AC)
      AAF=AAF+(DT/AAFD)*(AC-AAF)
      B1=B1+DT*(BA-BB)
      B2=B2+DT*(BB-BC)
      BAF=BAF+(DT/BAFD)*(BC-BAF)
      BAG=BAG+(DT/BAGD)*(BC-BAG)
      C1=C1+DT*(CA+CE+CJ-CB+CL+CN)
      C2=C2+DT*(CB+CD+CH-CC+CM+CP)
      C3=C3+DT*(CC-CD-CE-CF-CN)
      C4=C4+DT*(CF-CG-CH-CJ-CK)
```

```
      C5=C5+DT*(CK-CL-C4)
      C6=C6+DT*(CN-CP-CQ)
      CAA=CAA+(DT/CAAA)*(C1+C2+C3+C5+C6-CAA)
      CBC=CBC+(DT/CBCA)*(CBP-CBC)
      CBE=CBE+(DT/CBEA)*(CBG-CBE)
      CBG=CBG+(DT/CBGA)*(CLE-CBG)
      CBL=CBL+(DT/CBLA)*(CLF-CBL)
      CCE=CCE+(DT/CCEA)*(CFA-CCE)
      CCF=CCF+(DT/CCFA)*(CCG-CCF)
      CCK=CCK+(DT/CCKA)*(CCL-CCK)
      CCN=CCN+(DT/CCNA)*(CCD-CCN)
      CCQ=CCQ+(DT/CCQA)*(CFP-CCQ)
      CCS=CCS+(DT/CCSA)*(S1-CCS)
      CCZ=CCZ+(DT/CCZD)*(CCY-CCZ)
      CEB=CEB+(DT/CEBA)*(CLG-CEB)
      CFG=CFG+(DT/CFGA)*(CLA-CFG)
      CFL=CFL+(DT/CFLD)*(CF-CFL)
      CFM=CFM+(DT/CFMD)*(CFK-CFM)
      CHB=CHB+(DT/CHBA)*(CHC-CHB)
      CHD=CHD+(DT/CHDA)*(CJD-CHD)
      CJC=CJC+(DT/CJCA)*(CJD-CJC)
      FAH=FAH+(DT/FAHD)*(S1-FAH)
      FAJ=FAJ+(DT/FAJD)*(FAL-FAJ)
      FBH=FBH+(DT/FBHD)*(S1-FBH)
      FBJ=FBJ+(DT/FBJD)*(FBD-FBJ)
      FEC=FEC+(DT/FECD)*(S1-FEC)
      FPM=FPM+(DT/FPMD)*(S1-FPM)
      FPN=FPN+(DT/FPND)*(FPE-FPN)
      P1=P1+DT*(PA-PB)
      P2=P2+DT*(PB-PC)
      PAE=PAE+(DT/PAEA)*(CCZ-PAE)
      PAF=PAF+(DT/PAFA)*(PC-PAF)
      PAJ=PAJ+DT*(CC/PL-PAJ)
      S1=S1+DT*SA
      SAB=SAB+(DT/SABD)*(SAC-SAB)
 5000 CONTINUE
      IF(KTIME.EQ.1) CALL PRINT(-1.,1)
      RETURN
      END
```