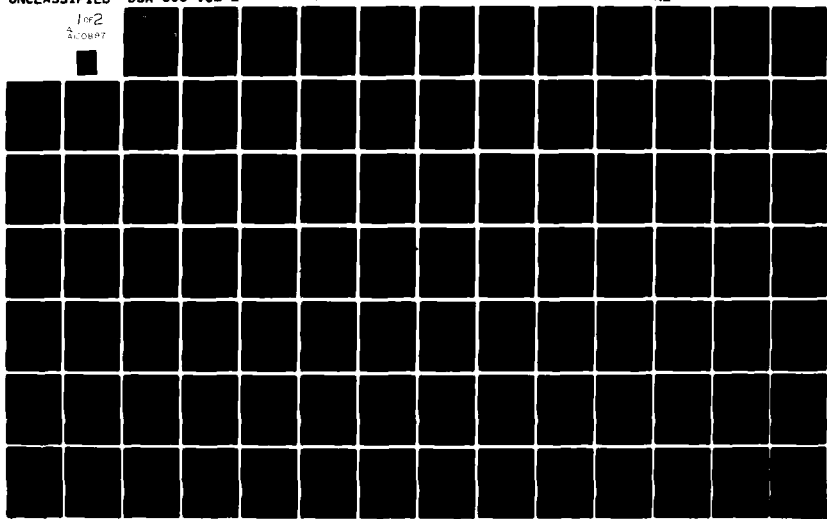


AD-A100 897

DECISION-SCIENCE APPLICATIONS INC ARLINGTON VA F/G 5/3
THE DYNEVAL (DYNAMIC ECONOMIC VALUES) MODEL. VOLUME II. DOCUMENT--ETC(U)
MAY 81 G E PUGH, M T NUNENKAMP, J C KRUPP AC00C104
DSA-306-VOL-2 NL

UNCLASSIFIED

for 2
20087



LEVEL

10

May 1981

DTIC FILE COPY AD A100897

THE DYNEVAL (DYNAMIC ECONOMIC VALUES) MODEL
DOCUMENTATION

AC00C104 - Volume II

DTIC
JUL 2 1981

prepared for:

UNITED STATES ARMS CONTROL AND DISARMAMENT AGENCY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

prepared by:

Decision-Science Applications, Inc.
1500 Wilson Boulevard, Suite 810
Arlington, Virginia 22209

Ym

81 7 01 006

REPORT DOCUMENTATION PAGE		1. REPORT NO. AD-A100897		2. Recipient's Accession No.																
4. Title and Subtitle The DYNEVAL (Dynamic Economic Values) Model Volume II - Documentation				5. Report Date May 1981 (Pub.)																
7. Author(s) George Pugh				6.																
9. Performing Organization Name and Address Decision-Science Applications, Inc. 1500 Wilson Blvd., Suite 810 Arlington, VA 22209				8. Performing Organization Rept. No.																
12. Sponsoring Organization Name and Address U.S. Arms Control and Disarmament Agency 320 21st St., N.W. Washington, D.C. 20451				10. Project/Task/Work Unit No.																
				11. Contract(C) or Grant(G) No. (C) AC00C104 (G)																
				13. Type of Report & Period Covered Final																
15. Supplementary Notes Volume II of three volumes.				14.																
16. Abstract (Limit: 200 words) Volume II provides documentation and user guide for the DYNEVAL (Dynamic Economic Values) Model. DYNEVAL uses Lagrange Dynamic Programming to chart through time the optimal trajectory of an economy which has either been disrupted in some way or in which value structures have been changed.																				
17. Document Analysis <table border="0"> <tr> <td>a. Descriptors</td> <td>Production Models</td> <td>Forecasting</td> <td>Population</td> <td>Military Expenditure</td> </tr> <tr> <td></td> <td>Operations Research</td> <td>Economic Analysis</td> <td>Investment</td> <td>Computer Programs</td> </tr> <tr> <td></td> <td>Econometrics</td> <td>Capital</td> <td>Consumption</td> <td></td> </tr> </table> b. Identifiers/Open-Ended Terms Economic Models 0503 Input/Output Routines 0902 c. COSATI Field/Group 05-03 Economics						a. Descriptors	Production Models	Forecasting	Population	Military Expenditure		Operations Research	Economic Analysis	Investment	Computer Programs		Econometrics	Capital	Consumption	
a. Descriptors	Production Models	Forecasting	Population	Military Expenditure																
	Operations Research	Economic Analysis	Investment	Computer Programs																
	Econometrics	Capital	Consumption																	
18. Availability Statement: Release unlimited.			19. Security Class (This Report) UNCLASSIFIED.		21. No. of Pages 50															
			20. Security Class (This Page) UNCLASSIFIED.		22. Price															

DO NOT PRINT THESE INSTRUCTIONS AS A PAGE IN A REPORT

INSTRUCTIONS

Optional Form 272, Report Documentation Page is based on Guidelines for Format and Production of Scientific and Technical Reports, ANSI Z39.18-1974 available from American National Standards Institute, 1430 Broadway, New York, New York 10018. Each separately bound report—for example, each volume in a multivolume set—shall have its unique Report Documentation Page.

1. Report Number. Each individually bound report shall carry a unique alphanumeric designation assigned by the performing organization or provided by the sponsoring organization in accordance with American National Standard ANSI Z39.23-1974, Technical Report Number (STRN). For registration of report code, contact NTIS Report Number Clearinghouse, Springfield, VA 22161. Use uppercase letters, Arabic numerals, slashes, and hyphens only, as in the following examples: FASEB/NS-75/87 and FAA/RD-75/09.
2. Leave blank.
3. Recipient's Accession Number. Reserved for use by each report recipient.
4. Title and Subtitle. Title should indicate clearly and briefly the subject coverage of the report, subordinate subtitle to the main title. When a report is prepared in more than one volume, repeat the primary title, add volume number and include subtitle for the specific volume.
5. Report Date. Each report shall carry a date indicating at least month and year. Indicate the basis on which it was selected (e.g., date of issue, date of approval, date of preparation, date published).
6. Sponsoring Agency Code. Leave blank.
7. Author(s). Give name(s) in conventional order (e.g., John R. Doe, or J. Robert Doe). List author's affiliation if it differs from the performing organization.
8. Performing Organization Report Number. Insert if performing organization wishes to assign this number.
9. Performing Organization Name and Mailing Address. Give name, street, city, state, and ZIP code. List no more than two levels of an organizational hierarchy. Display the name of the organization exactly as it should appear in Government indexes such as Government Reports Announcements & Index (GRA & I).
10. Project/Task/Work Unit Number. Use the project, task and work unit numbers under which the report was prepared.
11. Contract/Grant Number. Insert contract or grant number under which report was prepared.
12. Sponsoring Agency Name and Mailing Address. Include ZIP code. Cite main sponsors.
13. Type of Report and Period Covered. State interim, final, etc., and, if applicable, inclusive dates.
14. Performing Organization Code. Leave blank.
15. Supplementary Notes. Enter information not included elsewhere but useful, such as: Prepared in cooperation with . . . Translation of . . . Presented at conference of . . . To be published in . . . When a report is revised, include a statement whether the new report supersedes or supplements the older report.
16. Abstract. Include a brief (200 words or less) factual summary of the most significant information contained in the report. If the report contains a significant bibliography or literature survey, mention it here.
17. Document Analysis. (a). Descriptors. Select from the Thesaurus of Engineering and Scientific Terms the proper authorized terms that identify the major concept of the research and are sufficiently specific and precise to be used as index entries for cataloging.
(b). Identifiers and Open-Ended Terms. Use identifiers for project names, code names, equipment designators, etc. Use open-ended terms written in descriptor form for those subjects for which no descriptor exists.
(c). COSATI Field/Group. Field and Group assignments are to be taken from the 1964 COSATI Subject Category List. Since the majority of documents are multidisciplinary in nature, the primary Field/Group assignment(s) will be the specific discipline, area of human endeavor, or type of physical object. The application(s) will be cross-referenced with secondary Field/Group assignments that will follow the primary posting(s).
18. Distribution Statement. Denote public releasability, for example "Release unlimited", or limitation for reasons other than security. Cite any availability to the public, with address, order number and price, if known.
19. & 20. Security Classification. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED).
21. Number of pages. Insert the total number of pages, including introductory pages, but excluding distribution list, if any.
22. Price. Enter price in paper copy (PC) and/or microfiche (MF) if known.

DSA No. 306

11/ May 1981

(6) THE DYNEVAL (DYNAMIC ECONOMIC VALUES) MODEL. Volume II.
DOCUMENTATION.

(15) AC00C104 - Volume II

+ DSA-306-VOL-2

prepared for:

UNITED STATES ARMS CONTROL AND DISARMAMENT AGENCY

(1) Final rept.,

(12) 134

prepared by:

Decision-Science Applications, Inc.
1500 Wilson Boulevard, Suite 810
Arlington, Virginia 22209

(10) George E. Pugh
Max T. Nunenkamp
Joseph C. Krupp

343157

CONTENTS

SECTION		Accession For NTIS GRA&I ☒ DTIC TAB ☐ Unannounced ☐ Justification By Distribution/ Availability Codes Avail and/or Dist Special <i>A</i>	PAGE
1.0	INTRODUCTION		1
2.0	FUNDAMENTAL RELATIONS		4
2.1	Fundamental Variables		4
2.2	Basic Assumptions		6
2.3	Derived Relations		8
3.0	REQUIRED INPUT DATA		11
4.0	CURRENT ECONOMY		13
4.1	Utility Coefficients		13
5.0	EQUILIBRIUM ECONOMY		20
5.1	Capital Inventory Levels and Investment		20
5.2	Capital Rental Values		23
5.3	Finding the Equilibrium State		23
6.0	SINGLE TIME PERIOD OPTIMIZER		27
6.1	Production Activities		27
6.2	Consumption Activities		30
6.3	Investment Activities		31
6.4	Trade		34
6.5	Convergence		34
7.0	PROPAGATION THROUGH TIME OF CAPITAL AND VALUE		36
7.1	Resource Propagation		38
7.2	Value Propagation		40
APPENDICES			
A	USER MANUAL		44
B	DATA AGGREGATOR--AGGRAT		80
C	GLOSSARY OF VARIABLES USED IN DSA ECONOMIC RECOVERY MODEL		91
D	SOURCE CODE LISTING		95

FIGURES

<u>NUMBER</u>		<u>PAGE</u>
1-1	Logic Flow of Economic Model	2
4-1	Logic Flow of Current Economy Parameter Calculation	14
5-1	Logic Flow of Equilibrium Calculation	21
6-1	Logic Flow of Single Timer Period Optimizer	28
7-1	Temporal Structure of the Model	37
A-1	Sample Listing of Inputs and Outputs	55
B-1	Description of DSA Data Format	82

1.0 INTRODUCTION

The Dynamic Economic Values Model (DYNEVAL) uses Lagrange Dynamic Programming to chart through time the optimal trajectory of an economy which has either been disrupted in some way, or in which people have changed their value structure (e.g., by giving more emphasis to leisure or the military). "Optimal trajectory" is that time-dependent mix of investment and production activities which allows consumers (people, government, military, etc.) to maximize over time a sum of value functions. Too much early investment causes people in early time periods to have too low a standard of living based on their value functions; too little investment causes the economy to recover too slowly.

The basic logic flow of the model is shown in Fig. 1-1. Since each problem the model solves is different, a storage allocator is used to parse a large storage array into a series of smaller data arrays of appropriate dimensions for the problem. The basic data for the problem, including current activity levels and a direct requirements matrix, are then input to the program which proceeds to calculate both the current day economy and the final equilibrium economy. In some studies, one does not care about the trajectory to equilibrium, but rather only about the final equilibrium economy itself. In these cases, the model may be stopped here.

Once the above is accomplished, the model starts its real job of calculating the optimal investment, production, and consumption activity levels which gradually move the economy to its equilibrium state. This is done by optimizing each time period in turn and propagating the capital resources produced forward in time, taking into account population growth, gestation time, depreciation, etc. Each time period is presented with a set of capital which the past has made available to it, and a set of capital values which the future is willing to pay. From these, and internal requirements detailed later, the single time period produces,

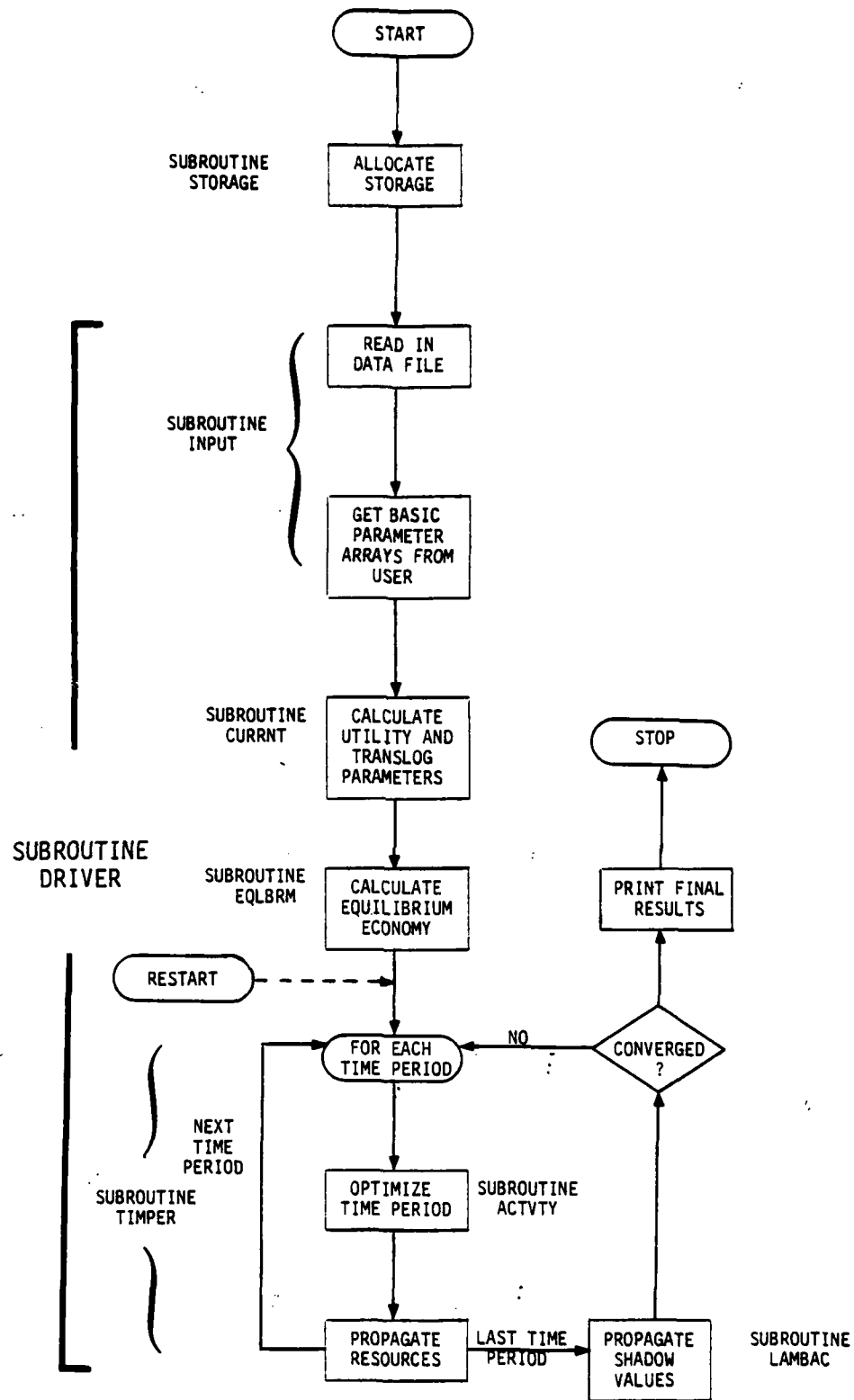


Figure 1-1. Logic Flow of Economic Model

invests, and consumes in an optimal way. As each time period considers its options it automatically assigns a rental (shadow) value to the capital and labor needed to produce the goods and services for this time period. These rental values can then be propagated backward in time (taking into account discount values, population growth, gestation times, etc.) to give each time period a more realistic view of the future. Each time period is then reoptimized taking into account this new information and new capital passed to it by the past and the process is repeated.

The following sections give the mathematical details of each stage of the program. These are technical in nature and intended for use by someone interested in understanding the precise way the model operates or interested in making changes to the model. For the person interested in using the model as a research tool, Appendix A is a user's manual explaining how to run the model and giving examples of output appearing on the terminal screen and the line printer. Appendix B is a description of the companion program, AGGRAT, used to aggregate economic data and put it into model format. Definitions of all pertinent variables and arrays are given in Appendix C. Appendix D is a complete listing of the source code for the model.

2.0 FUNDAMENTAL RELATIONS

Throughout the course of the program certain variables are known and others must be calculated. In the single time period optimizer, e.g., the amount of capital resources available for production is known. In the equilibrium calculation the production and investment activities are known as a simple function of the resources, and the rental value for capital is known as a simple function of its cost. In order to calculate missing quantities certain fundamental relationships among resource levels, activity levels, costs, prices, and values need to be derived. This section details relationships among these variables for the production and consumption activities. In addition, other relations are valid when the economy is in equilibrium. These will be discussed in Sec. 5. Investment and trade activities present special problems and will be discussed in Sec. 6.

2.1 FUNDAMENTAL VARIABLES

The following is a list of the fundamental variables. Other subsidiary variables will be defined as they occur. Superscripts refer to types of activities, i.e., production, investment (production of capital), or consumption, and are not dummy indices.

K_j = Inventory level of capital resource j . These are physical items (machines, etc.) and may be used only in the production of a single consumable

K_i^{eff} = "Effective level" (explained later) of the total capital mix used to produce consumable i . To reduce cumbersome notation the "eff" will sometimes be omitted in this text. The subscript i will implicitly refer to K_i^{eff} , while the subscript j will refer to a particular capital resource K_j

L_j = Amount of labor used in the production of consumable i

L = Total labor pool

x_i^D = Activity level for production of consumable i

x_i^C = Activity level for the i th consumption activity. This activity need not consume any of consumable i , nor is it restricted to consuming only one consumable. Rather it is defined as the consumption of any particular group of consumables by any or all of a group of consumers (people, government, etc.).

x_j^I = Activity level for investment, i.e., the production of capital resource j

Λ_i^I = Value of the effective capital needed to produce consumable i

Λ_{K_j} = Value of capital resource j

Λ_i = Price of consumable i

λ_{K_j} = Rental value of capital resource K_j

λ_L = Rental value of labor. It has the same value regardless of the activity

There are three basic types of activities: production (x_i^D), consumption (x_i^C), and investment (x_j^I). The production activities require capital, labor, and a specified mix of various consumables. Each production activity produces only one type of consumable and each consumable is produced by only one type of production activity. Consumption activities require no capital and no labor (except consumption of leisure which requires only labor), but do require a specified mix of consumables. The same consumable may be used in several different

consumption activities. Investment activities produce capital. They require no labor but do require a specified mix of consumables. The investment activities are, in effect, bookkeeping activities whereby various consumables are turned into a single capital resource specific to a given production activity. The labor required to produce the capital is implicit in the labor required to produce the consumables used to create the capital.

2.2 BASIC ASSUMPTIONS

In order to calculate any relations among the above listed variables, certain assumptions must be made about how the economy responds to changes in prices, supply, or demand. The following assumptions are basic to the model. In some cases these assumptions are given functional form. Where the form and its implications are spread throughout the model, as opposed to a localized function which could easily be changed in the model, it will be noted as such.

1. Consumption activity levels vary inversely with cost. The higher the cost of the consuming activity, the lower the level of the activity. If consumption activity i uses α_{ij} of consumable j per unit activity, then its cost, C_i , is given as:

$$C_i = \sum_j \alpha_{ij} \Lambda_j \quad (2-1)$$

In the model Assumption 1 is given functional form by defining the consumption activity level as:

$$x_i^c = \left(\frac{\Gamma_i}{C_i} \right)^{1/b_i} \quad (2-2)$$

where Γ_i = normalization constant (based on current economy) and b_i = elasticity of demand factor.

The exact form of Eq. 2-2 is not required by the model. However, this form is a particularly simple yet useful way of expressing Assumption 1. It is equivalent to defining a utility for consumption activity i of the form

$$U(X_i^C) = r_i \frac{X_i^{C(1-b_i)} - 1}{1 - b_i}, \quad b_i \neq 1 \quad (2-3a)$$

$$= r_i \ln(X_i^C), \quad b_i = 1 \quad (2-3b)$$

and stating that X_i^C is the level where the marginal utility, dU/dX_i^C , is equal to the cost. Usually b_i is set equal to 1.0. For this value of b_i , doubling the activity level results in the same increment of utility regardless of the original value. It states, for example, that two people whose salaries are doubled experience the same increase in happiness regardless of their original salaries. Values of b_i between 0.5 and 2.0 seem to be most reasonable.

2. The production of consumable i may be considered a function of capital and labor alone: $X_i^P = f(K_i^{\text{eff}}, L_i)$. Furthermore, capital and labor may be traded for one another in the production of any consumable according to a particular relation known as the translog function (also called the "constant elasticity function"):

$$X_i^P = g_i \left[\gamma_{K_i} K_i^{-\beta_i} + \gamma_{L_i} L_i^{\beta_i} \right]^{-1/\beta_i} \quad (2-4)$$

where g_i = normalization constant (based on current economy)

$$\gamma_{K_i} + \gamma_{L_i} = 1.0$$

$$\frac{1}{\beta_i + 1} = \text{elasticity of substitution}$$

The implications of the translog function, e.g., its homogeneity [$X_i^P(aK, aL) = aX_i^P(K, L)$], are spread throughout the model. It would be difficult to change to another production function.

3. Capital is used in place of labor where capital is cheaper, and labor is used in place of capital where labor is cheaper. That is, capital and labor are traded one for the other until the value of their marginal products are equal. This assumption is expressed mathematically as:

$$\lambda_{K_i} = L \frac{\partial X_i^p / \partial K_i^{\text{eff}}}{\partial X_i^p / \partial L_i} = - \lambda_L \frac{dL_i}{dK_i^{\text{eff}}} \quad X_i^p = \text{constant} \quad (2-5)$$

4. The price of any consumable is equal to the average unit cost of producing it, i.e.,

$$\Lambda_i = \frac{K_i \lambda_{K_i} + L_i \lambda_L + \sum_j X_i^p \alpha_{ij} \Lambda_j}{X_i^p} \quad (2-6)$$

where α_{ij} = fixed amount of consumable j required to produce one unit of consumable i .

While capital and labor are traded one for the other in production activities, there exists no mechanism in the model for substituting one consumable for another, i.e., the α_{ij} in Eq. 2-6 are fixed for the given problem.

2.3 DERIVED RELATIONS

The following relationships are derived from the basic assumptions. Their use in the model will be discussed in later sections.

1. Assumption 1 allows an independent calculation of λ_L , the rental value of labor. For the form of Assumption 1 described by Eq. 2-2,

$$\lambda_L = \frac{r_{\text{Leisure}}}{(L - \sum_i L_i)^{\beta_{\text{Leisure}}}} \quad (2-7)$$

since the cost of leisure is equal to the cost of labor.

2. Differentiating the translog function of Assumption 2 (Eq. 2-4) and substituting it in Eq. 2-5 of Assumption 3 yields a relation between the ratio of capital to labor and the ratio of their rental values:

$$\left(\frac{K_i}{L_i}\right)^{\beta_i+1} = \frac{\gamma_{K_i} \lambda_L}{\gamma_{L_i} \lambda_{K_i}} \quad (2-8)$$

3. The fair price assumption, Assumption 4, may be written as:

$$\Lambda_i' X_i^D = K_i \lambda_{K_i} + L_i \lambda_L \quad (2-9)$$

where

$$\Lambda_i' = \Lambda_i - \sum_j \alpha_{ij} \Lambda_j$$

stating that value added is divided between capital and labor. Equations 2-4 and 2-8 may be substituted in Eq. 2-9 to yield:

$$L_i = K_i \left[\frac{\gamma_{L_i}}{\left(\frac{\lambda_{K_i}}{g_i \gamma_{K_i} \Lambda_i'} \right)^{-(\beta_i)/(1+\beta_i)} - \gamma_{K_i}} \right]^{1/\beta_i} \quad (2-10a)$$

$$L_i = K_i \left[\frac{\gamma_{K_i}}{\frac{\lambda_L}{g_i \gamma_{L_i} \Lambda'_i} - \gamma_{L_i}} \right]^{-1/\beta_i} \quad (2-10b)$$

or

$$\Lambda'_i = \frac{\lambda_L}{g_i \gamma_L} \left[\gamma_K \left(\frac{L_i}{K_i} \right)^{\beta_i} + \gamma_L \right]^{\frac{\beta+1}{\beta}} \quad (2-10c)$$

3.0 REQUIRED INPUT DATA

The model expects the following data to be input at the start of the model. The first five items describe the given economy directly and cannot be changed within the model. The last five are particular to the study at hand and may be changed for each run.

1. Current day activity levels. These include the investment, trade, and consumption activities as well as the production activities.
2. Current day capital resource inventories.
3. Gestation times, depreciation rates, discount rates, and growth rates (above population growth) for all capital resources. These values are usually calculated or input at the time the model data is formed by using the data aggregator AGGRAT, which is described in Appendix B.
4. A direct requirements matrix, α_{ij} , giving the amount of consumable j used per unit of activity i for all activities i mentioned in (1) above.
5. A rental array giving the current mix of capital and labor for each of the production activities.
6. The values of b_i used in the utility function for consumption activities (see Assumption 1, Sec. 2).
7. The fraction of each capital resource which "survived" the proposed disruption.
8. Minimum consumption activity levels. While this array is normally set to zero, it allows the user to specify, e.g., that a minimum of agricultural products must be consumed. By setting the minimum activity level negative, the user allows a zero level of consumption.
9. The ratios of two utility coefficients Γ ; those used for the study to the current day Γ values. If no change in priorities is postulated, these ratios are unity.

10. The translog elasticity (see Sec. 4.2) parameters β for both the capital labor trade-offs and, if two types of capital are required, the effective capital calculation.

4.0 CURRENT ECONOMY

While the inventory and activity levels of the current economy are input directly to the model, the utility function and translog coefficients which describe the economy must be calculated. This is done by subroutine CURRNT. The logical flow of CURRNT is diagrammed in Fig. 4-1.

4.1 UTILITY COEFFICIENTS

Equation 2-2 gave the consumption activity level as a function of the cost for that activity.

$$x_i^C = \left(\frac{\Gamma_i}{C_i} \right)^{1/b_i} \quad (4-1)$$

The Γ_i are normalized to the current economy. Since physical units are defined as the amount valued at 1.0 in the current economy, $C_i = 1$, and

$$\Gamma_i = (x_i^C - Z_i)^{b_i} \cdot F_i^{b_i} \quad (4-2)$$

where Z_i = minimum activity level (offset),
 F_i = ratio of study economy to current economy
priorities,
and x_i^C refers here to the current consumption level.

The parameters b_i are input by the user (see Appendix A).

4.2 TRANSLOG PARAMETERS

Capital and labor may be traded one for the other according to the translog function, Eq. 2-4:

$$x_i^P = g_i \left[\gamma_{K_i} K_i^{-\beta_i} + \gamma_{L_i} L_i^{-\beta_i} \right]^{-1/\beta_i} \quad (4-3)$$

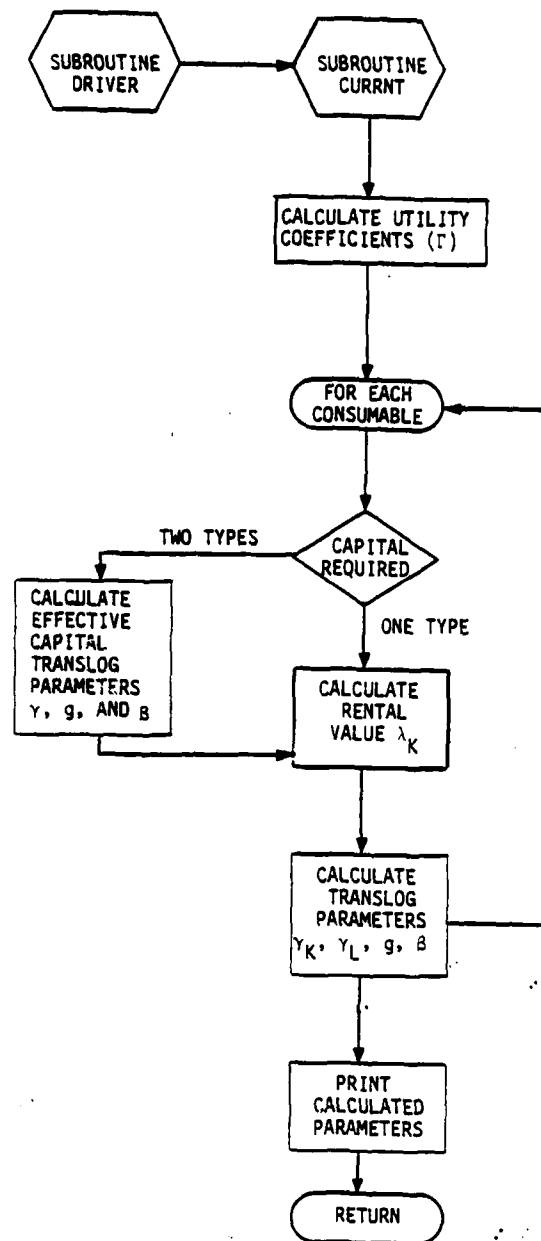


Figure 4-1. Logic Flow of Current Economy Parameter Calculation

The elasticity parameter, β_i , is input directly or indirectly by the user (see 4.2.3 below). The γ 's and e_i must be calculated.

4.2.1 Capital and Labor Coefficients γ

Equation 2-8 gave the relation between the ratio of capital and labor and the ratio of their rental values as:

$$\left(\frac{K_i}{L_i}\right)^{\beta_i+1} = \frac{\gamma_{K_i} \lambda_L}{\gamma_{L_i} \lambda_{K_i}} \quad (4-4)$$

In the current economy, $\lambda_L = 1.0$ by definition of the units of labor. The rental value for capital may be found by dividing the total spent by industry i on capital by the total inventory of the capital used by the industry:

$$\lambda_{K_i} = \frac{r_j}{K_j} \quad (4-5)$$

where r = amount spent on capital j in the current economy.

Since K_i and L_i are known for the current economy, as well as λ_{K_i} and λ_L , the γ_{K_i} and γ_{L_i} may be found from Eq. 4-4 and the fact that their sum must be unity:

$$\gamma_{K_i} = \frac{\lambda_{K_i} (K_i/L_i)^{\beta_i+1}}{\lambda_{K_i} (K_i/L_i)^{\beta_i+1} + \lambda_L} \quad (4-6a)$$

$$\gamma_{L_i} = \frac{1.0}{\lambda_{K_i} \left(\frac{K_i}{L_i}\right)^{\beta_i+1} + \lambda_L} \quad (4-6b)$$

4.2.2 Normalization Factor g_i

The parameter g_i is simply the value required to reproduce the current level of activity if the capital and labor used equal that used in the current economy.

$$g_i = \frac{x_i^p}{\left[\gamma_{K_i} K_i^{-\beta_i} + \gamma_{L_i} L_i^{-\beta_i} \right]^{-1/\beta_i}} \quad (4-7)$$

4.2.3 Elasticity Parameter β_i

Under most circumstances the elasticity parameter is input by the user. Valid values for β_i lie in the range $-1.0 \leq \beta_i \leq \infty$. If β_i lies outside this range the model assumes the user wants the model to calculate the actual β_i and that the input value refers to the productivity at infinite labor levels. Some industries are able to increase their output significantly by increasing the amount of labor used, e.g., by adding a second shift. Other industries are not. If the user thinks that the most a particular industry could increase its output given fixed capital and unlimited labor is by a factor $p > 1.0$, he may set $\beta_i = -p$, and the model will calculate β_i such that:

$$\frac{g_i \left[\gamma_{K_i} K_i^{-\beta_i} + \gamma_{L_i} (L_i = \infty)^{-\beta_i} \right]^{-1/\beta_i}}{g_i \left(\gamma_{K_i} K_i^{-\beta_i} + \gamma_{L_i} L_i^{-\beta_i} \right)^{-1/\beta_i}} = p \quad (4-8)$$

Equation 4-8 may be rewritten as:

$$\frac{\frac{\gamma_{K_i}}{\gamma_{L_i}} \left(\frac{K_i}{L_i} \right)^{-\beta_i}}{\frac{\gamma_{K_i}}{\gamma_{L_i}} \left(\frac{K_i}{L_i} \right)^{-\beta_i} + 1.0} = p^{-\beta_i} \quad (4-9)$$

From Eq. 4-4:

$$\frac{\gamma_{K_i}}{\gamma_{L_i}} = \lambda_{K_i} \left(\frac{K_i}{L_i} \right)^{\beta_i + 1} \quad (4-10)$$

Therefore, combining Eq. 4-9 and 4-10:

$$\frac{\lambda_K \left(\frac{K_i}{L_i} \right)}{\lambda_K \left(\frac{K_i}{L_i} \right) + 1.0} = p^{-\beta_i}$$

or

$$\beta_i = \frac{-\ln \frac{\lambda_K (K_i/L_i)}{\lambda_K (K_i/L_i) + 1.0}}{\ln p} \quad (4-11)$$

4.2.4 Effective Capital

Normally a given production activity uses one type of capital. The model allows two types of capital per industry, however, and treats the combination as an "effective" level of capital. The two types of capital are combined by a separate translog function:

$$K_i^{\text{eff}} = g \left[\gamma_1 K_{j_1}^{-\beta} + \gamma_2 K_{j_2}^{-\beta} \right]^{-1/\beta} \quad (4-12)$$

To avoid cumbersome notation, we will set $K_i^{\text{eff}} = K_{\text{eff}}$ and "j₁" and "j₂" to "1" and "2" respectively in this section.

Eq. 2-5 may be generalized for this case to derive the relationship between $\lambda_{K_{eff}}$ and the individual rental values:

$$\begin{aligned}
 \lambda_{K_j} &= \lambda_L \frac{dL}{dK_{eff}} \frac{\partial K_{eff}}{\partial K_j} \\
 &= \lambda_{K_{eff}} \frac{\partial K_{eff}}{\partial K_j} \\
 &= \lambda_{K_{eff}} \gamma_j g \left(\frac{K_{eff}}{g K_j} \right)^{\beta+1}
 \end{aligned} \tag{4-13}$$

where $j = 1$ or 2 .

While Eq. 4-13 may look imposing, it offers a very simple representation for $\lambda_{K_{eff}}$ given λ_{K_1} and λ_{K_2} :

$$\begin{aligned}
 \lambda_{K_1} K_1 + \lambda_{K_2} K_2 &= e(K_{eff}/e)^{\beta+1} (\gamma_1 K_1^{-\beta} + \gamma_2^{-\beta}) \lambda_{K_{eff}} \\
 &= K_{eff} \lambda_{K_{eff}}
 \end{aligned}$$

or,

$$\lambda_{K_{eff}} = \frac{\lambda_1 K_1 + \lambda_2 K_2}{K_{eff}} \tag{4-14}$$

Equation 4-14 states that the effective rental value of the capital combination is equal to the total spent on both divided by the effective level of capital. This property follows directly from the homogeneity of the translog production function.

The model calculates the γ and g parameters for the effective capital function in a way similar to Secs. 4.2.1 and 4.2.2. Equation 4-5 gives the λ_{K_j} for the current economy. By replacing K_i with K_1 , L_i with K_2 , λ_{K_i} with λ_{K_1} and λ_L with λ_{K_2} , Eqs. 4-6 allow the calculation of γ_{K_1} and γ_{K_2} . Similarly, replacing X_i^D with $K_1 + K_2$, and β_i with β , Eq. 4-7 allows the calculation of g .

5.0 EQUILIBRIUM ECONOMY

The equilibrium economy is defined as that economy where capital is being produced at such a rate that capital per unit population remains constant. Furthermore, real prices must remain constant. For these reasons, the equilibrium economy is not necessarily the current economy, although it could be if no industry is experiencing a growth greater than population growth and there are no consumer preference changes. The equilibrium economy is calculated in subroutine EQLBRM. The logical flow of EQLBRM is diagrammed in Fig. 5-1.

5.1 CAPITAL INVENTORY LEVELS AND INVESTMENT

Capital must increase with the population to provide the same level of production per person. Assuming an exponential growth in population, P , and a depreciation rate, d , the capital inventory at time t , $K(t)$, is related to the inventory at time t_0 by:

$$K(t) = K(t_0) e^{P(t-t_0)} \quad (5-1a)$$

$$\text{and } K(t) = K(t_0) e^{-d(t-t_0)} + \int_{t_0}^t X^I(t' - \tau) e^{-d(t-t')} dt' \quad (5-1b)$$

where

τ = gestation time

$X^I(t' - \tau)$ = amount of new capital becoming available at t'

Note that Eq. 5-1b is a statement of physical fact. Inventory at time t must be equal to the inventory at time t_0 which has not depreciated, plus what has been made between t_0 and t . Eq. 5-1a, on the other hand, is valid only in equilibrium. It states that capital inventory increases at the same rate as population.

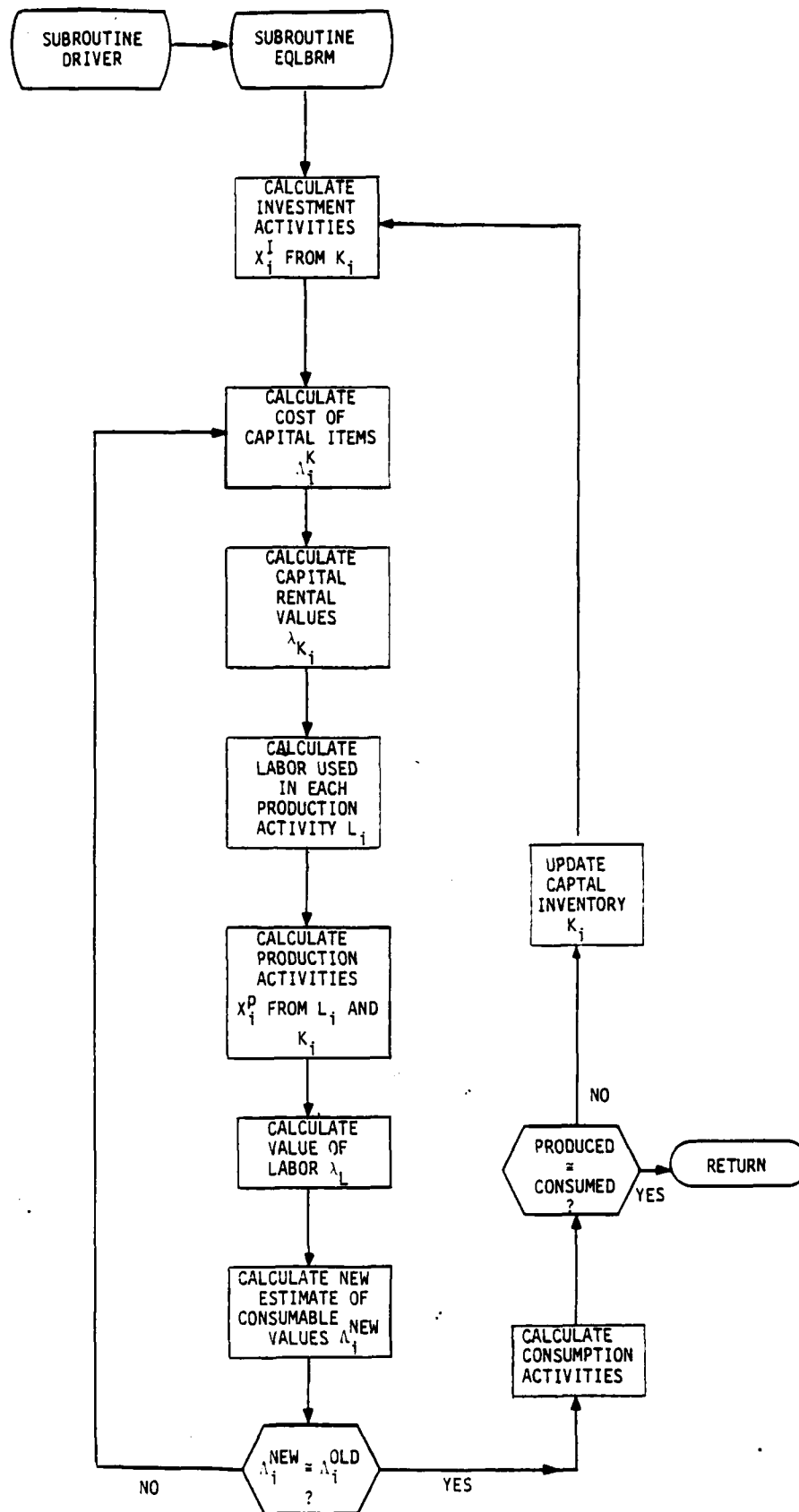


Figure 5-1. Logic Flow of Equilibrium Calculation

Setting the right hand sides of Eqs. 5-1 equal to each other and differentiating with respect to t yields the expression:

$$K(t_0) \left[P e^{P(t-t_0)} + d e^{-d(t-t_0)} \right] =$$

$$X^I(t - \tau) - d \int_{t_0}^t X^I(t' - \tau) e^{-d(t-t')} dt' \quad (5-2)$$

The integral in Eq. 5-2 may be related to $K(t)$ and $K(t_0)$ from Eq. 5-1 to finally yield:

$$K(t_0)(P + d) e^{P(t-t_0)} = X^I(t - \tau) \quad (5-3)$$

or

$$K(t_0)(P + d) e^{P(t-t_0)} e^{P\tau} = X^I(t)$$

which, through Eq. 5-1a becomes

$$K(t)(P + d) e^{P\tau} = X^I(t) \quad (5-4)$$

Equation 5-4 relates the activity level in equilibrium to the inventory level. The inventory level will grow in time at the same rate as population. It is convenient in the model and clearer to the user to normalize inventory and activity level to the current day population so that the time dependence in Eq. 5-4 may be removed. After all, there is no way to know before the model is run how long it will take to reach equilibrium, and a curve (trajectory) which flattens at equilibrium is much easier to interpret than one which grows exponentially at the population growth rate. With this normalization, Eqs. 5-1 become:

$$K(t) = K(t_0) \quad (5-5a)$$

$$K(t) = K(t_0) e^{-(P+d)(t-t_0)} + e^{-P\tau} \int_{t_0}^t X^I(t' - \tau) e^{-(P+D)(t-t')} dt' \quad (5-5b)$$

where K is normalized to population and is not the same as the K of Eqs. 5-1. The end result of solving Eq. 5-5 for X^I as a function of K is identical to Eq. 5-4 with the time dependence removed:

$$X^I_{\text{Equilibrium}} = K_{\text{Equilibrium}} (P + d) e^{P\tau} \quad (5-6)$$

5.2 CAPITAL RENTAL VALUES

In equilibrium, the rental value for capital is related to its cost of production in a very simple way. Since the cost of production must be equal to the sum over time of all the discounted rents, and since in equilibrium the rental value must remain constant, Eq. 7-10 (See Sec. 7) reduces to:

$$\Lambda_{\text{Equilibrium}} = \lambda_{\text{Equilibrium}} \frac{e^{-(P+\rho)\tau}}{P + d + \rho} \quad (5-7)$$

where Λ = cost of production

λ = rental value per unit time

ρ = discount rate

5.3 FINDING THE EQUILIBRIUM STATE

The following is the algorithm used by the model to calculate the equilibrium state (see Fig. 5-1). It is an iterative technique which converges rapidly. If the current economy contains no growth relative to population growth, i.e., is itself an equilibrium, and if the utility parameters have not changed, the process clearly takes one iteration. If the above is not true, the process may take ten to twenty iterations.

The reader should note that by this time the parameters γ , β , Γ , b , and α have been calculated from the current economy and are fixed hereon and available to the model.

1. Assume some set Λ_i of consumable values and some set K_i of capital resources, e.g., the current economy values.
2. Calculate the investment activities X_i^I from Eq. 5-6.
3. Find the cost of capital, Λ_i^I , from the consumable values Λ_i .
4. Calculate the rental value λ_{K_i} , from Λ_i^I and Eq. 5-7.
5. Calculate the labor required for each production activity from Eq. 2-10a:

$$L_i = K_i \frac{\gamma_{L_i}}{\left(\frac{\lambda_{K_i}}{e_i \gamma_{K_i} \Lambda_i^I} \right)^{-\beta_i / (1+\beta_i)} - \gamma_{K_i}} \quad (5-8)$$

$$\text{where } \Lambda_i^I = \Lambda_i - \sum_j \alpha_{ij} \Lambda_j$$

and α_{ij} = amount of consumable j required per unit activity i .

6. From L_i and K_i calculate the production activities X_i^P from the translog function, Eq. 2-4:

$$X_i^P = e_i \left[\gamma_{K_i} K_i^{-\beta_i} + \gamma_{L_i} L_i^{-\beta_i} \right]^{-1/\beta_i} \quad (5-9)$$

7. Find λ_L from the utility function for leisure:

$$\lambda_L = \frac{\Gamma_{\text{Leisure}}}{(L - \sum_i L_i)^{b_{\text{Leisure}}}} \quad (5-10)$$

8. Find a new estimate for Λ_i from Eq. 2-6:

$$\Lambda_i^{\text{Calculated}} = \frac{\lambda_{K_i} K_i + \lambda_L L_i + \sum_{j \neq i} x_j^P \alpha_{ij} \Lambda_j}{(1 - \alpha_{ii}) x_i^P} \quad (5-11)$$

$$\Lambda_i^{\text{New}} = \frac{1}{2} (\Lambda_i^{\text{Calculated}} + \Lambda_i) \quad (5-11b)$$

9. If $\Lambda_i^{\text{New}} \approx \Lambda_i$, proceed. Otherwise go back to Step 3.
10. Next find the cost per unit consumption activity and the level of that activity:

$$\Lambda_i^C = \sum_j \alpha_{ij} \Lambda_j \quad (5-12a)$$

$$x_i^C = \left(\Gamma_i / \Lambda_i^C \right)^{1/b_i} \quad (5-12b)$$

11. Compare how much of each commodity was produced, x_i^P , with how much was used, $x_i^{\text{Used}} = \sum_{\text{all activities } j} x_j \alpha_{ji}$, and form a new estimate of the $\{K_i\}$:

$$K_i^{\text{New}} = K_i^{\text{Old}} \frac{2x_i^{\text{Used}}}{x_i^{\text{Used}} + x_i^{\text{P}}} \quad (5-13)$$

If $K_i^{\text{New}} \approx K_i^{\text{Old}}$, the equilibrium is known. If not, go back to Step 2.

6.0 SINGLE TIME PERIOD OPTIMIZER

The single time period optimizer (subroutine ACTVTY in the model) optimizes the level of production and consumption activities, calculates the rental values for capital, and calculates the level of investment (production of capital). Inputs include capital inventory available for the production activities, capital value from the future (Phase 1) or investment activities (Phase 2), depreciation rates for capital, translog parameters (see Assumption 2 of Sec. 2), minimum consumption activity levels, and the direct requirements matrix. Outputs include the rental values for capital and labor, the activity levels, and the consumable values. The logic flow of subroutine ACTVTY is diagrammed in Fig. 6-6.

6.1 PRODUCTION ACTIVITIES

Each consumable i is produced by a separate activity using capital K_i , labor L_i , and a specified fraction α_{ij} of each of the N consumables per unit activity. If more than one type of capital is used, K_i refers to the "effective" capital produced by the combination (see Sec. 4.2.4).

The capital used in the production of consumable i is the entire inventory of capital available. Obviously one cannot use more capital than available, and using less would cause its value to drop to zero relative to labor. Therefore, from the translog function,

$$x_i^p = g_i \left[\gamma_{K_i} K_i^{-\beta_i} + \gamma_{L_i} L_i^{-\beta_i} \right]^{-1/\beta_i} \quad (6-1)$$

the level of production activity is dependent on labor alone. But how much labor should be applied? Section 2 answered that question through Eq. 2-10:

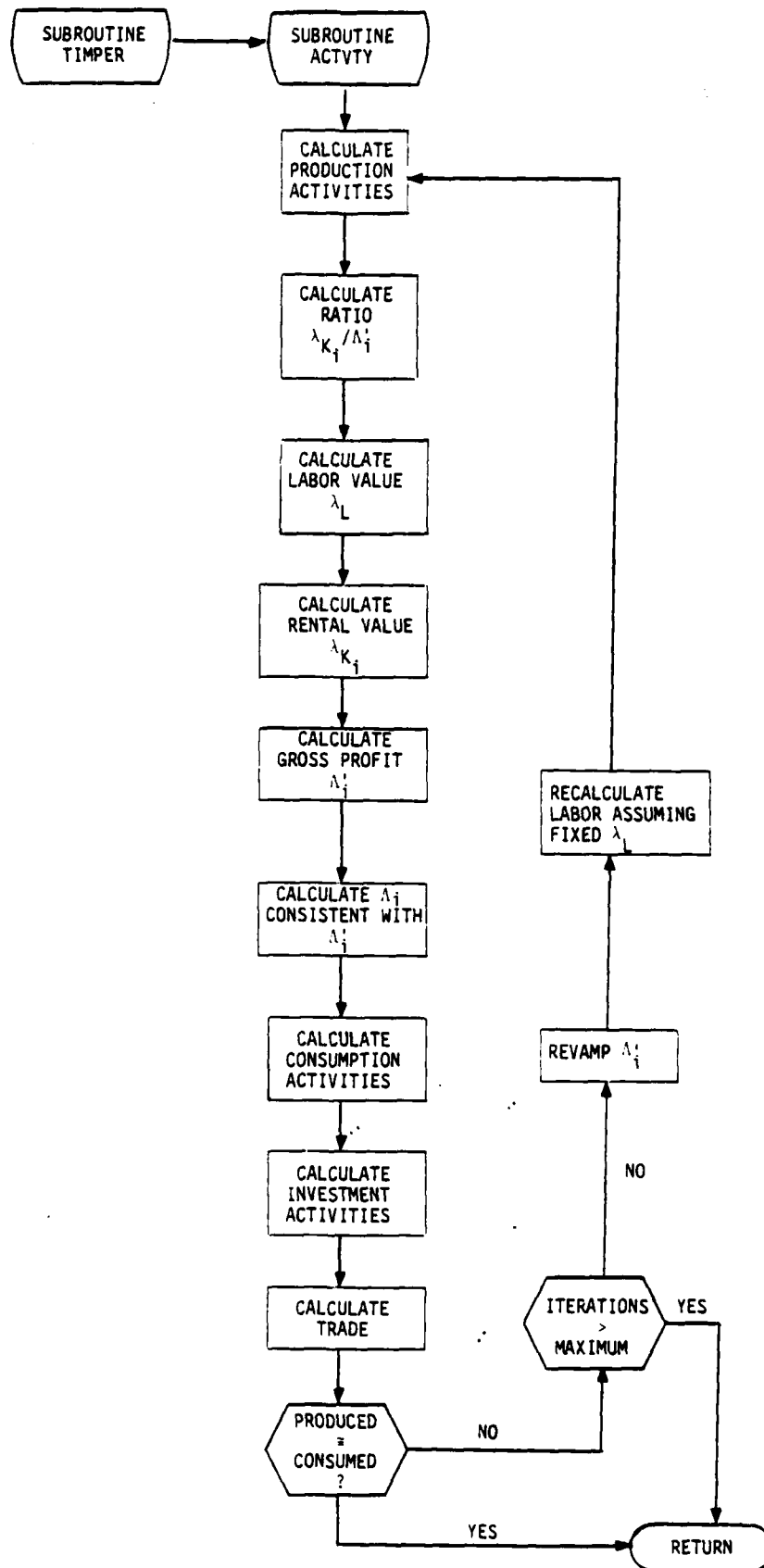


Figure 6-1. Logic Flow of Single Time Period Optimizer

$$L_i = K_i \left[\frac{\gamma_{L_i}}{\left(\frac{\lambda_{K_i}}{g_i \gamma_{K_i} \Lambda'_i} \right)^{-\beta_i/(1+\beta_i)} - \gamma_{K_i}} \right]^{1/\beta_i} \quad (6-2)$$

where the parameters were described in Sec. 2. The important variables in Eq. 6-2 are the Λ'_i , the gross profit from producing consumable i which is available for renting the capital and labor used to produce it:

$$\Lambda'_i \equiv \Lambda_i - \sum_j \alpha_{ij} \Lambda_j$$

By combining Eqs. 6-1 and 6-2, Λ'_i may be related directly to the activity level x_i^p :

$$\left(\frac{x_i^p}{g_i} \right) = \left(\frac{\lambda_{K_i}}{g_i \gamma_{K_i} \Lambda'_i} \right)^{1/\beta_i + 1} K_i \quad (6-3)$$

Equations 6-1 through 6-3 are the crux of the single time period optimization. The process starts as follows:

1. Assume some level of labor, L_i , for each production activity. This labor is usually the labor required to produce the same level of activity as achieved for this consumable and time period last pass.
2. From L_i calculate x_i^p from Eq. 6-1.

3. From X_i^P calculate the ratio $\lambda_{K_i}/(e_i \gamma_{K_i} \Lambda_i')$ for each consumable from Eq. 6-3.
4. Sum each L_i to find the total labor used and calculate λ_L from the resulting consumption of leisure:

$$\lambda_L = \frac{\Gamma_{\text{Leisure}}}{\left(L - \sum_i L_i\right)^{b_{\text{Leisure}}}} \quad (6-4)$$

5. Find λ_{K_i} from Eq. 2-8:

$$\left(\frac{K_i}{L_i}\right)^{\beta_i+1} = \frac{\gamma_{K_i} \lambda_L}{\gamma_{L_i} \lambda_{K_i}} \quad (6-5)$$

6. From the ratio found in step 3 and λ_{K_i} , calculate the Λ_i' .

At this point the set $\{\Lambda_i'\}$ is known, but the individual Λ_i are not. However, through the definition of the Λ_i' in matrix notation,

$$\Lambda' = (I - \alpha)\Lambda$$

the $\{\Lambda_i\}$ may be found by inverting the matrix $(I - \alpha)$. From the Λ_i the consumption activities may be calculated.

6.2 CONSUMPTION ACTIVITIES

Once the Λ_i are known, the cost per unit of consumption activity i may be calculated. If the consumption activity i uses α_{ij} of consumable j , then the cost per unit of activity is:

$$C_i = \sum_j \alpha_{ij} \Lambda_j \quad (6-7)$$

The activity level is determined by Eq. 2-2:

$$x_i^c = \left(\frac{r_i}{c_i} \right)^{1/b_i} \quad (6-8)$$

6.3 INVESTMENT ACTIVITIES

Investment activities are calculated in one of two ways depending on whether the program has just started or is trying to refine its trajectory.

6.3.1 Phase One

In phase one of the program, the level of investment activities is found by comparing the cost of producing capital with its expected worth to the future. Assuming the program has completed at least one pass, there exists a time stream of rental values for the various capital. These may be propagated backwards in time, accounting for discount rates, depreciation, etc., as described in Section 7.2, to yield a value for capital to each time period. The cost of producing the capital, which uses only consumables in its production, may be found in a way identical to Eq. 6-7.

The problem arises, however, that if the particular time period has a cost of production greater or less than the worth of the capital, it will want to produce zero or infinite amounts of the capital. This problem is essentially solved by assuming the value of the capital is correct only if the given time period makes capital in reasonable quantities. Less production increases the future value of the capital, and more production reduces the future value of the capital. The functional nature of this process is described in terms of two parameters, R_{TGT} and R_{FUT} . R_{FUT} is the capital resource level one gestation time in the future if nothing is produced. R_{TGT} is the target resource level, i.e., it is the level of inventory which existed one gestation time in the

future during the last pass. The difference, $R_{TGT} - R_{FUT}$, is what should be produced this time period assuming cost equals value.¹

Once R_{FUT} and R_{TGT} are known, X_i^I is found through the equation:

$$X_i^I = \frac{R_{TGT} - R_{FUT} + s_i \ln (\Lambda_i / C_i^I)}{\Delta t} \quad (6-9)$$

where Λ_i is the value of the capital propagated from the future, and s_i is a scale factor essentially equal to the equilibrium activity level X_{iEQ}^I . The effect of Eq. 6-9 is to produce negative feedback. If the investment activity is too small, rental values will be high in the future causing Λ_i to be large next pass, causing the time period to produce more than it did last pass. This in turn causes the inventory in the future to rise and rental values to fall, lowering Λ_i^I for the next pass.

While this process sounds as if it should cause swift convergence, it in fact causes too much feedback, resulting in an oscillatory state wherein every other pass results in too little investment followed by too much investment. This effect is cured by averaging inventories from pass to pass until a pseudo-stable trajectory arises, and then proceeding into phase two.

For the first pass of the program R_{TGT} is defined as the equilibrium resource level while Λ_i^I is the equilibrium capital value. These values cause the model to invest to such a degree that, while not producing an optimal trajectory, it at least produces one which brings the economy to equilibrium.

¹In reality, more needs to be produced because of depreciation. Therefore the model uses $R_{FUT}^I \equiv R_{GUT}(1 - \Delta t/F) + R_{FUT}(\Delta t/F)$,

where $F = e^{-P\tau} (1 - e^{-(P+d)\Delta t}) / (P+d)$ (See Sec. 7.1).

6.3.2 Phase Two

In phase two the investment trajectory is approximately known. At this stage the single time period optimizer accepts the investment activities as given, regardless of the cost of production or the future value of the capital. These activities are adjusted up or down external to the single time period optimizer through the use of an Everett type algorithm.

This algorithm operates in the following way. Consider a single activity X_i^I . On the previous pass the activity had a cost of production equal to C_i^I . After the rental values have been propagated back in time a value Λ_i results. For the next pass X_i^I is reset by:

$$X_i^I = X_i^I(1 + \epsilon), \quad C_i^I < \Lambda_i \quad (6-10)$$

or
$$X_i^I = X_i^I(1 - \epsilon), \quad C_i^I > \Lambda_i.$$

The parameter ϵ is reset each time by

$$\epsilon = \epsilon(1 + \delta F), \quad \text{for } S = +1$$

or
$$\epsilon = \epsilon(1 - \delta), \quad \text{for } S = -1 \quad (6-11)$$

where S is defined as the direction of change, i.e., $S = +1$ if $C_i^I > \Lambda_i$ both this pass and last pass or $C_i^I < \Lambda_i$ both this pass and last pass. $S = -1$ if $C_i^I > \Lambda_i$ for one of the two passes and $C_i^I < \Lambda_i$ for the other. The parameters δ and F are constants. In the model they are set to 0.6 and 0.5 respectively. This algorithm very quickly and efficiently narrows the investment trajectory to the optimum. Inventories are not averaged during phase two.

6.4 TRADE

The current ACDA version of DYNEVAL does not model trade explicitly. Some mechanism is needed, however, to account for the "undifferentiated imports" required in some production and investment activities. This is handled by summing all activities to determine the amount of undifferentiated imports required, and setting the trade activity at the proper level. All other commodities are imported or exported at this level according to the proportions in which they are imported or exported in the current economy.

6.5 CONVERGENCE

Once the production, consumption, trade and investment activities are known, a comparison may be made between the amount of a given consumable which is produced and the amount which is used or consumed. These two values should be equal. In general, however, they will not be, and a scheme must be developed to bring them together. This is accomplished by using once again an Everett type of algorithm on the Λ_i^1 of each consumable. When production exceeds consumption Λ_i^1 is reduced and when consumption exceed production Λ_i^1 is increased.

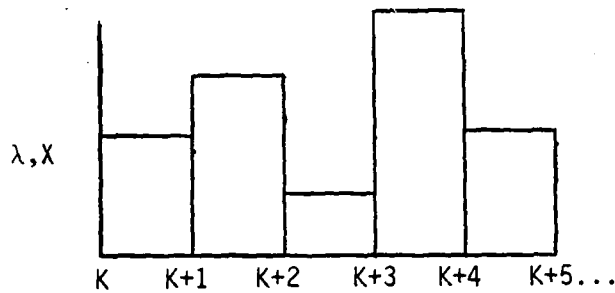
When Λ_i^1 is changed, all other variables will also change. One parameter, however, is less sensitive than most. This is the rental value of labor. Using Eq. 2-10b with the new value of Λ_i^1 and the old value of λ_L provides a new estimate for L_i . One may then proceed as discussed in 6.1 above.

¹It is important that the Everett algorithm be applied to the Λ_i^1 and not the individual Λ_i . Doing the latter results in wild swings in the production activities since Λ_i^1 can easily become negative resulting in zero production. A higher Λ_i^1 must result in increased production. A higher Λ_i need not do so if the other Λ_i increase as well.

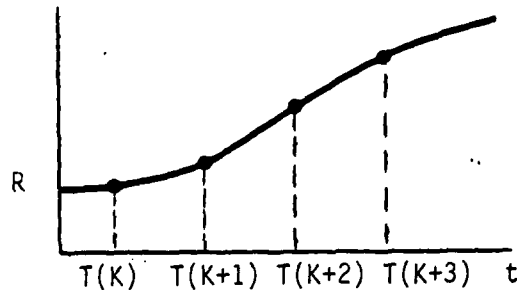
This process usually converges quickly and efficiently. When there has been a large destruction of capital, however, the early time periods may have trouble converging. This occurs because one or more industries is working near capacity and yet its Δ_i^j keeps getting signals to increase. The industry cannot increase its output so costs, including labor costs, increase until consumers cannot purchase anything and the system reverses itself. To prevent the model from spending too much of its time on a problem which will be changed anyway after the future sends its information back through time, a limit of 20 iterations is imposed on the single time period optimizer.

7.0 PROPAGATION THROUGH TIME OF CAPITAL AND VALUE

Figure 7-1 illustrates the temporal structure of the model. Each time period K is of equal length, Δt . The array, $T(K)$, stores the time at the middle of time period K , [i.e., $T(K) = (K - 1)\Delta t + \Delta t/2$]. $T_0(K)$ is the time at the start of time period K . The activities, X , and the shadow (rental) values, λ , are represented as constant rates within time period K :



Resource inventories are considered continuous in time, with the values at $T(K)$ stored in array $R(K)$:¹



¹In this section capital resources will be designated as R to conform to the model representation and to avoid confusion with the time index K .

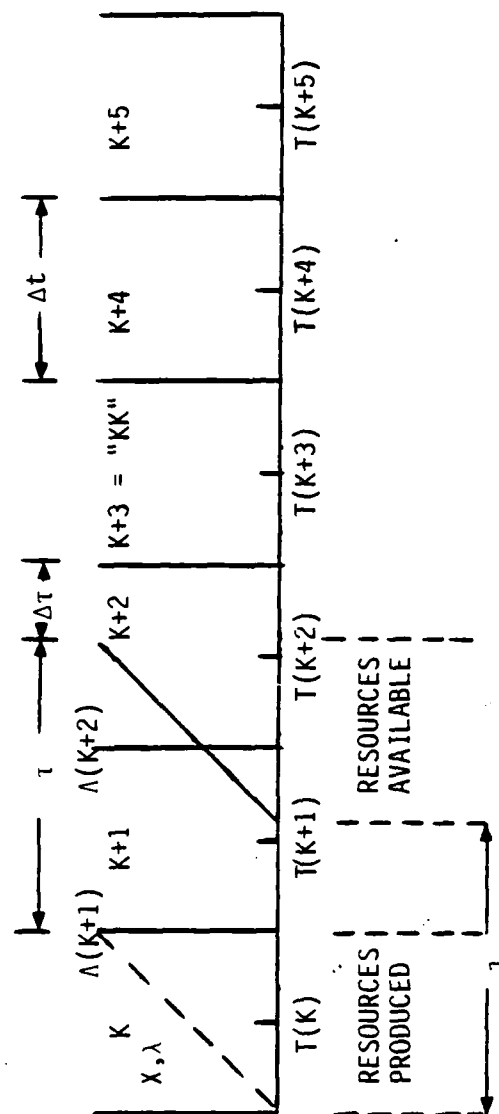
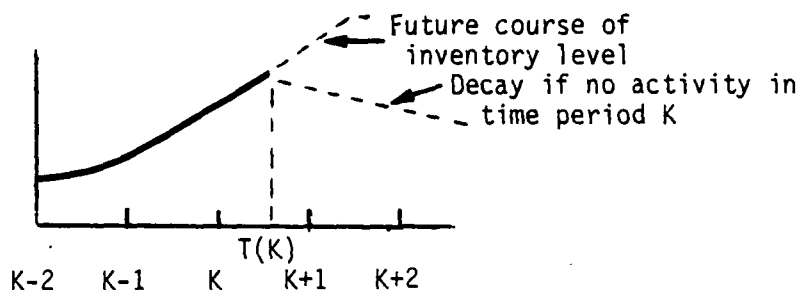


Figure 7-1. Temporal Structure of the Model

The shadow value $\Lambda(K + 1)$ represents the worth of the resource which time period K passes to the future measured in present day value. It does not represent the worth to time period $K + 1$. Rather, Λ is used by time period K in determining the amount it should produce, and so reflects the discounted future stream of λ 's encountered after the gestation time τ .

The resource level at the center of time period K , $R_i(K)$, is the inventory available to time period K . Strictly speaking, an integrated average of R_i would be a better approximation. However, for gestation times τ less than Δt the averaging would be incorrect anyway. For example, if $\tau = \frac{1}{2}\Delta t$, the following picture arises:



At the beginning of time period K , the activity level $X(K)$ has not been determined, and the resource level at the end of time period K reflects the natural depreciation of the resource. In this example, using $R(K)$ is obviously a better estimate of the available resources than an actual integration of the curve would yield.

7.1 RESOURCE PROPAGATION

After the X_i^{I-1} are determined, the resources produced are propagated through time. This occurs in subroutine TIMPER. As resources propagate,

$X_i^{I-1}(K)$ here refers to the rate of production of resource R_i during the Kth time period in units of resource/unit time.

they depreciate with a natural depreciation rate d . Furthermore, resources are normalized to population so as the population grows at a rate P , resource inventories are reduced. As the resource propagates, each value of $R(K' \geq K)$ is increased by an amount $\Delta R(K')$, where $\Delta R(K')$ is determined in the following way.

Resources begin to come on line at time $T_{\text{start}} = T_0(K) + \tau$
 $= T(K) - \frac{\Delta t}{2} + \tau$. For time $t \leq T_{\text{start}}$, $\Delta R = 0$. No resource is yet available due to the gestation time τ . Therefore, for K' such that $T(K') \leq T_{\text{start}}$,

$$\Delta R(K') = 0 \quad (7-1)$$

Resources finish coming on line at time $T_{\text{stop}} = T_0(K+1) + \tau$
 $= T(K+1) - \frac{\Delta t}{2} + \tau$. For time t between T_{start} and T_{stop} , ΔR reflects the incoming resources which have come on line by time t . Therefore, for K' such that

$$T_{\text{start}} \leq T(K') \leq T_{\text{stop}},$$

$$\begin{aligned} \Delta R(K') &= X^I(K') e^{-P\tau} \int_0^{T(K') - T_{\text{start}}} e^{-(P+d)t} dt \\ &= X^I(K') e^{-P\tau} \frac{1 - e^{-(P+d)[T(K') - T_{\text{start}}]}}{P + d} \end{aligned} \quad (7-2)$$

where d = depreciation rate
 P = population growth rate

$X^I_i(K)$ here refers to the rate of production of resource R_i during the K th time period in units of resource/unit time.

For time $t \geq T_{\text{stop}}$, ΔR reflects the depreciated value of all the resources produced in time period K :

$$\Delta R(K) = X(K) e^{-P} \frac{1 - e^{-(P+d)\Delta t}}{P+d} e^{-(P+d)[T(K') - T_{\text{stop}}]} \quad (7-3)$$

population growth during gestation
depreciation between time coming available and time when all units come on line
depreciation after all units are on line

Equations 7-1 through 7-3 may be combined as follows:

For all $K' \geq K$

$$\Delta R(K') = X(K) e^{-P\tau} \left[\frac{1.0 - \exp\left(-(P+d) \cdot \text{AMIN1}\{\Delta t, \text{AMAX1}[0.0, t(K') - T_{\text{start}}]\}\right)}{P+d} \right] \exp\left\{-(P+d) \cdot \text{AMAX1}[0.0, T(K') - T_{\text{stop}}]\right\} \quad (7-4)$$

Note that for $\tau < \frac{\Delta t}{2}$, $\Delta R(K) \neq 0$. This is allowed for all time periods except $K = 1$, since by definition $R(1)$ is the starting resource and if it is allowed to increase, the next pass of the model will in effect be solving a different problem with a higher starting resource. Therefore, the length of each time period must be such that $\Delta t \leq 2\tau$. The smaller Δt is, the longer each pass of the model takes since more time periods are required, but, in general fewer passes are required due to increased accuracy.

7.2 VALUE PROPAGATION

This section describes how the $\Lambda(K+1)$ are derived for the next pass. After a complete pass, subroutine LAMBAC propagates the rental values λ back in time to define a future worth, $\Lambda(K+1)$, to time period K for its capital. Since it is assumed that the activity level $X^1(K)$ is constant throughout the time period and that the decision

maker decides this level at the beginning of the time period, the basic equation for $\Lambda(K+1)$ is:

$$\Lambda(K+1)X^I(K)\Delta t' = \sum_{\text{future}} \text{discounted rents on amount produced} \quad (7-5)$$

The value $\Delta t'$ represents the fact that the value $\Lambda(K+1)$ is also discounted through the time period, i.e.,

$$\Delta t' = \frac{1.0 - e^{-\rho\Delta t}}{\rho}$$

where ρ is the discount rate. Throughout this discussion it will be assumed that Fig. 7-1 applies. The index KK in Fig. 7-1 refers to the first index after K such that gestation is completed:

$$KK = \text{INT} \left[\frac{T(K+1) + \tau}{\Delta t} + 1.5 \right]$$

In the $(KK-2)$ th time period, resources start to come on line and continue (in this time period) for a time $\Delta\tau$ (see Fig. 7-1). Therefore, the first component of $\Lambda(K+1)$ is:

$$\begin{aligned} \Lambda^1(K+1)X^I(K)\Delta t' &= \lambda(KK-2)e^{-(P+\rho)\tau} \int_0^{\Delta\tau} X^I(K) \frac{1 - e^{-(P+d)t}}{P+d} e^{-\rho t} dt \\ &\quad \begin{array}{l} \uparrow \\ \text{rental value} \end{array} \quad \begin{array}{l} \uparrow \\ \text{discount and} \\ \text{population growth} \\ \text{during gestation} \end{array} \quad \begin{array}{l} \uparrow \\ \text{amount available} \\ \text{between } \tau \leq t \leq \tau + \Delta\tau \end{array} \quad \begin{array}{l} \uparrow \\ \text{discount} \\ \text{beyond} \\ \text{gestation time} \end{array} \\ &= \lambda(KK-2) e^{-(P+\rho)\tau} X^I(K) \left[\frac{e^{-(P+d+\rho)\Delta\tau} - 1}{P+d+\rho} - \frac{e^{-\rho\Delta\tau} - 1}{\rho} \right] \quad (7-6) \end{aligned}$$

In the $(KK-1)$ th time period, resources continue to come on line for a time at $\Delta t - \Delta \tau$. They all then depreciate for the rest of the time period $\Delta \tau$. Thus:

$$\Lambda^2(K+1)X^I(K)\Delta t' = \lambda(KK-1) e^{-(P+\rho)\tau} X^I(K)$$

$$\begin{aligned} & \cdot \int_{\Delta \tau}^{\Delta t} \left[\frac{1 - e^{-(P+d)t}}{P+d} \right] e^{-\rho t} dt \\ & + \left[\frac{1 - e^{-(P+d)\Delta t}}{P+d} \right] e^{-\rho \Delta t} \cdot \left[\int_0^{\Delta \tau} e^{-(P+d+\rho)t} dt \right] \\ & = \frac{\lambda(KK-1) e^{-(P+\rho)\tau} X^I(K)}{P+d} \left[\frac{e^{-(P+d+\rho)\Delta t} - e^{-(P+d+\rho)\Delta \tau}}{P+d+\rho} \right. \\ & \left. + \frac{e^{-\rho \Delta \tau} - e^{-\rho \Delta t}}{\rho} + \frac{[e^{-\rho \Delta t} - e^{-(P+d+\rho)\Delta t}][1 - e^{-(P+d+\rho)\Delta \tau}]}{P+d+\rho} \right] \quad (7-7) \end{aligned}$$

For the rest of time, the resources continue to depreciate:

$$\begin{aligned} \Lambda^3(K+1)X^I(K)\Delta t' &= e^{-(P+\rho)\tau} X^I(K) \sum_{k=0}^{\infty} \lambda(KK+k) e^{-\rho \Delta t} \left[\frac{1 - e^{-(P+d)\Delta t}}{P+d} \right] \\ & \cdot e^{-(P+d+\rho)(\Delta \tau + k\Delta)} \left[\frac{1 - e^{-(P+d+\rho)\Delta t}}{P+d+\rho} \right] \quad (7-8) \end{aligned}$$

Similar expressions may be derived for $\Lambda(K + 2)$. If this is done it is seen that $\Lambda(K + 1)$ may be related to $\Lambda(K + 2)$ and the three rental values, $\lambda(KK - 2)$, $\lambda(KK - 1)$, and $\lambda(KK)$:

$$\begin{aligned}
 \Lambda(K + 1) = & \lambda(K + 2) e^{-(P+d+\rho)\Delta t} \\
 & + \frac{e^{-(P+\rho)\tau}}{(P+d)\Delta t} \left(\lambda(KK - 2) \left[\frac{1 - e^{-\rho\Delta\tau}}{\rho} - \frac{1 - e^{-(P+d+\rho)\Delta\tau}}{P+d+\rho} \right] \right. \\
 & + \lambda(KK - 1) \left\{ \frac{e^{-\rho\Delta\tau} [1 + e^{-(P+d+\rho)\Delta t}] - e^{-(P+d+\rho)\Delta t} - e^{-\rho\Delta t}}{\rho} \right. \\
 & + \left. \frac{e^{-\rho\Delta t} [1 - e^{-(P+d+\rho)\Delta\tau}]}{P+d+\rho} + \frac{[e^{-(P+d+\rho)\Delta t} - e^{-(P+d+\rho)\Delta\tau}]}{P+d+\rho} \right\} \\
 & + \lambda(KK) \left\{ \frac{e^{-\rho\Delta t} [e^{-(P+d+\rho)\Delta\tau} - e^{-(P+d+\rho)\Delta t}]}{P+d+\rho} \right. \\
 & + \left. e^{-(P+d+\rho)\Delta t} \frac{(e^{-\rho\Delta t} - e^{-\rho\Delta\tau})}{\rho} \right\} \quad (7-9)
 \end{aligned}$$

Equation 7-9 is used by the model in subroutine LAMBAC to derive the shadow values $\Lambda(K + 1)$ used in the next pass. Note that at equilibrium $\Lambda(K + 1) = \Lambda(K + 2)$ and $\lambda(KK - 2) = \lambda(KK - 1) = \lambda(KK)$. In this case, Eq. 7-9 reduces to:

$$\Lambda = \frac{\lambda e^{-(P+d+\rho)\tau}}{P+d+\rho} \quad (7-10)$$

which is the equilibrium value used in Sec. 5.1.

APPENDIX A
USER'S MANUAL

A.1 INTRODUCTION

This appendix explains how to use the Dynamic Economic Values Model (DYNEVAL). It is assumed that a data file exists containing the primary data describing the current economy. This file is most easily created through the use of the data aggregator AGGRAT which aggregates DSA format economic data into economic sectors specified by the user and puts the aggregated data onto a file in a form accepted by the model. AGGRAT is described in Appendix B.

Besides the data describing the current economy, which includes depreciation rates, gestation times for capital formation, etc., specific variables describing the disruption or change to the economy are needed. These variables are unique to each study and need to be entered when the model is run. The following is a step by step description of how these variables are entered, what options are available to the user, and what the final output of the model (terminal and line printer) looks like. A sample listing of line printer output is given in Sec. A.5.

A.2 START-UP

Upon initiating execution of DYNEVAL, the following question is asked:

IS THIS A RESTART?

The model has a restart capability which allows the user to stop the model and restart it where it left off with changes in print and plot switches (see Sec. A.3). The restart capability is especially useful when peculiarities in the summary outputs are noticed and the user wants to get detailed prints or plots of the next couple of passes. This question, like all yes/no questions, may be answered with YES(NO) or simply Y(N).

DYNEVAL next asks:

FULL PRINT OF EACH PASS (Y) OR
SUMMARY ONLY (N)

This question refers to line printer output. The model calculates the optimal economic trajectory through an iterative technique. A summary of each pass is printed on both the terminal and the line printer (see Sec. D). If the user answers this question with YES, however, a time period by time period listing of important parameters is also printed on the line printer. A description of this output is given in Sec. A.4.

Besides numerical output, the model is also capable of graphical summaries. The next question asked by DYNEVAL is:

IS OUTPUT FOR PLOTS DESIRED? (YES, NO, OR END).

A YES or END answer opens (creates) a plotting file and sets the plot switch. If the user wants to see how selected variables change from pass to pass or wants to make graphs at various stages of convergence, he should answer with YES. If the user does not care about the convergence process but would like time period plots of the final economic trajectories, he should answer END. It should be noted that if the plot option is selected the model merely outputs relevant data onto a file. A separate program is required to convert the data into plots. Such a program (PLOTTER) has been written using the Tektronix software, PLOT-10. Should a different software package (e.g., DEC's DSSPLA) be desired, a new program will have to be designed.

Once the output of the model has been determined, DYNEVAL queries the user concerning the parameters for the study. The first data asked for is:

INPUT PASS LIMIT AND STOPPING CRITERION

The model will continue its operation until either convergence is achieved or it has completed a maximum number of passes. The pass limit is the maximum number of passes the user wants the model to complete. A value of 50-100 is normally sufficient. The stopping criterion is the value of FOM2, the second figure-of-merit printed in the summary of each pass, which indicates that convergence is satisfactory. The various figures-of-merit are discussed in Sec. A.3. A value of 0.01-0.05 for the stopping criterion is usually sufficient.

If this is a restart, the program proceeds where it left off. Otherwise, DYNEVAL next asks for the extent of the run:

INPUT NUMBER OF TIME PERIODS DESIRED

The number of time periods times the length of each time period should be large enough so that the economy has time to reach its equilibrium state. Forty to fifty years is sufficient for most studies. After the number of time periods has been input, the model is able to calculate the total storage requirement for the study. It informs the user how much storage he is using and how much is available:

STORAGE USED = 8732, STORAGE AVAILABLE = 20,000

If too much storage is required, the following message will be printed on the terminal:

INCREASE DIMENSION OF ARRAY W IN MAIN PROGRAM TO AT
LEAST (amount required) -- PROGRAM ABORTED

The user will have to increase the dimension of array W and change the value of parameter NW in the main program and recompile. Ample storage is allowed for most studies.

Assuming the storage requirement is smaller than the available storage, DYNEVAL next asks for the length of each time period:

INPUT LENGTH OF TIME PERIOD (YEARS)

For a given length of time over which the model optimizes, greater accuracy with fewer passes is achieved with many time periods of short duration. More data storage and longer time per pass is also required, however, so the user needs to trade off the costs and benefits of the division. In no case may the time period length be greater than twice the smallest gestation time. If this occurs, DYNEVAL will warn the user:

TIME PERIOD LENGTH MAY NOT BE GREATER THAN (NO. OF YEARS),
I.E., 2.* MINIMUM GESTATION TIME OF (SMALLEST GESTATION
TIME) INPUT LENGTH OF TIME PERIOD (YEARS)

The model next asks the user about the basic parameters of the study. It remembers the parameters used for the last study, but the user may change any he wishes. If the user would like to see the current values of the basic parameters he may either answer YES to the following:

WOULD YOU LIKE BASIC PARAMETER ARRAYS
PRINTED ON LP? (Y or N),

which will output the basic arrays to the line printer, or, if this question is answered with NO, he may have specific arrays displayed on the terminal by answering YES to:

WOULD YOU LIKE ANY OF THE BASIC PARAMETER ARRAYS
DISPLAYED? (Y or N)

Assuming the user answers YES, DYNEVAL answers with:

INPUT ARRAY NAME
TYPE 'M' TO DISPLAY MENU

Entering an M on the terminal shows the user what the "basic parameter arrays" are and how to access them.

MENU: Z = MINIMUM CONSUMPTION ACTIVITY
 B = UTILITY EXPONENT FOR CONSUMPTION ACTIVITY
 G = RATIO OF POST-ATTACK UTILITY COEFFICIENTS TO TODAY'S
 UTILITY COEFFICIENTS

 BETA = TRANSLOG EXPONENT FOR CAPITAL/LABOR TRADE-OFFS
 (IF .LT. -1.0, BETA IS CALCULATED TO GIVE THE
 LIMITING PRODUCTIVITY A VALUE OF ABS (INPUT
 VALUE) (See Sec. 4.2.3)

 BET2 = TRANSLOG EXPONENT FOR CAPITAL/CAPITAL TRADE-OFFS
 (ONLY FOR THOSE PRODUCTION ACTIVITIES USING
 2 TYPES OF CAPITAL)

 F = FRACTION OF CAPITAL SURVIVING

 M = REDISPLAY MENU

 E = END OF REQUESTS

INPUT ARRAY NAME

TYPE 'M' TO DISPLAY MENU

After perusing the data, the user may wish to change some of it:

 WOULD YOU LIKE TO CHANGE ANY OF THE DATA (Y or N)

If the user answers with YES, he may change any or all of the data provided in the above menu:

 INPUT ARRAY NAME, DATA CHANGE, AND INDUSTRY NUMBER
 (OR RANGE OF INDUSTRY NUMBERS)
 TYPE 'M' TO DISPLAY MENU

The data arrays are dimensioned by the number of consumables, number of capital items, number of consumption activities, etc. If the user tries to change a piece of data outside its normal range, DYNEVAL will inform the user of his error:

EXCUSE ME?

RANGE MUST BE BETWEEN 1 AND 5

(values for example only)

When all changes are complete, the user has the option of once again seeing the data either printed on the line printer or displayed on the terminal. If all is in order, the run begins with the calculation of the final equilibrium state. When this is completed, the model tells the user how long the calculation took by printing:

CPU TIME TO CALCULATE EQUILIBRIUM = (NUMBER) CPU SECONDS
FOR (NUMBER) ITERATIONS

on the terminal. Following this calculation, the optimal trajectory to this equilibrium state is calculated through the iterative technique of Dynamic Lagrange Programming.

A.3 RESTART

If, for any reason, the user wishes to stop the run, he need only type any character on the terminal. At the beginning of each pass, the program checks to see if the terminal input buffer is empty. If it is not, the model responds with:

EXCUSE ME?

YOU HAVE INTERRUPTED ME

TYPE "C" TO CONTINUE, "S" TO STOP

The user may then stop the program by typing on "S". All files will be closed and the program will end.

The user may then check the output by printing file PRINT.DAT. If he wishes to continue the run later, he may simply reexecute EREC and answer YES to the question

IS THIS A RESTART?

(see Sec. B). All printer output in a restart is appended to the current version of PRINT.DAT so the initial output is not lost.

In normal operation, a restart is also possible if the program fails for some reason, e.g., because of a system crash, as long as the failure did not occur while the insurance file was being written. The insurance file is rewritten after each pass and the output file (PRINT.DAT) is closed to save the output and then reopened in the append mode. To be safe, however, it is advisable to stop the program with a FORTRAN interrupt as described above.

A.4 TERMINAL OUTPUT

Throughout the course of the run, a summary of each pass is displayed on the terminal, e.g.:

PASS 5; 24.320 CPU SEC; 508 LOOPS; FOM = 0.967E-02 0.179 1.000

The above summary is for the fifth pass which took 24.32 cpu sec to complete. The number of loops refers to the cumulative total of single time period optimization iterations. There is a maximum number of iterations for any given time period (nominally 20). The three figure-of-merit's are defined for each pass as:

$$FOM(1) = \frac{\sum_{k=1}^N \sum_{i=1}^M \bar{R}_{ik} (\Delta R_{ik})^2}{\sum_{k=1}^N \sum_{i=1}^M \bar{R}_{ik}} \quad (1a)$$

$$FOM(2) = \frac{\sum_{k=1}^N \sum_{i=1}^M (\Delta \Lambda_{ik})^2 x_{ik}}{\sum_{k=1}^N \sum_{i=1}^M x_{ik}} \quad (1b)$$

$$FOM(3) = \frac{\sum_{k=1}^N \sum_{i=1}^M \Lambda_{ik} x_{ik} \epsilon_{ik}^2}{\sum_{k=1}^N \sum_{i=1}^M \Lambda_{ik} x_{ik}} \quad (1c)$$

where

$N \equiv$ Number of time periods

$M \equiv$ Number of capital resources

$\bar{R}_{ik} \equiv$ Average inventory of capital i ,
time period k ($\bar{R}_{ik} = \frac{1}{2}(R_{ik} + R_{ik}(\text{last pass}))$)

$\Delta R_{ik} \equiv$ Percentage difference in R_{ik}
($\Delta R_{ik} = (R_{ik} - R_{ik}(\text{last pass}))/\bar{R}_{ik}$)

$\Delta \Lambda_{ik} \equiv$ Percentage difference between value of capital
and effective cost of producing capital

$\Lambda_{ik} \equiv$ Effective cost of producing capital

$x_{ik} \equiv$ Amount of capital i produced in time period k

$\epsilon_{ik} \equiv$ Everett parameter (phase two only).

The first figure-of-merit is a measure of how closely the capital inventory trajectory is being reproduced each pass. It is the square root of the average square percentage change in the inventory, weighted by the average size of that inventory. The second figure-of-merit shows how closely the value of the capital, found by propagating the rental values back in time, matches the cost of producing the capital. It is simply the RMS value of the percentage difference of the two weighted by the size of the investment activity. The third figure-of-merit is a measure of how well the Everett algorithm of phase two (see Sec. 6.3.2) is performing. The change in investment activity level, ϵ_{jk} , is weighted by the total cost of capital produced.

Phase two of the model is started whenever one of the following is achieved:

1. FOM (1) < .005
2. $K > 20$
3. $K > N/2$

where K is the current pass number. When phase two begins, the following message is printed on the terminal:

BEGINNING PHASE TWO

The user will notice that after each summary print on the terminal the cursor will remain at the end of the line for a short period and then return to the beginning of the next line. The cursor return is a signal that the insurance file (INS.DAT) has been written. This insurance file is used by the model whenever the user specifies that the run is a restart. It is rewritten at the end of each pass.

A.5 LINE PRINTER OUTPUT

A sample of the line printer output is given in Sec. A.5.1 below. The output from the line printer begins by describing the basic parameter

arrays for this run. These consist of the length of the run, number of consumables, etc., minimum activity levels, utility function elasticity parameters (β), the ratio of post-attack to preattack utility function coefficients (Γ), the input elasticity parameters for the translog substitution function (β) (see Sec. A.2), and the fraction of capital surviving.

Following the above is a listing of the basic data arrays for the economy under study: current activity levels, resource inventories, gestation times, depreciation, growth factors, discounts, direct requirements matrix for consumables, the current capital and labor usage for each production activity, and a summary of the level of the current economy (e.g., GNP).

After these basic arrays, derived current day values for functional parameters are displayed. These include the actual elasticity parameters (β), the utility function coefficients (Γ), the rental values for capital and labor (λ), the translog parameters for the mix of capital and labor (γ) and, if two types of capital are used in a production activity, the translog parameters for the effective capital, and the translog normalization coefficients (e).

The equilibrium values of selected arrays are displayed next. These include the new utility function coefficients (Γ), resource inventories, activity levels, values, and capital rental values. A listing of the productivity of each production industry as a function of the labor/capital ratio is also given as an aid to determining those industries which are capital intensive and those which are labor intensive. Following these arrays is again a summary of equilibrium economic levels.

The output next shows the level of capital inventories at time zero and proceeds to duplicate the terminal output (Sec. A.4). If the user specified during the start-up phase that he wanted detailed prints, then the output will also contain a listing of the inputs and outputs as shown in Fig. A-1.

		INPUT FOR PERIOD 9									
R	0.6529E+05	0.4390E+05	0.3007E+05	0.1495E+06	0.5433E+05	0.2902E+05	0.4021E+05	0.2671E+06	0.1030E+07	1.000	1.000
FLAM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
FLAM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
RTRGT	0.1024E+06	0.6870E+05	0.6334E+05	0.3630E+06	0.1265E+06	0.9000E+05	0.1358E+06	0.8437E+05			
RTRGT	0.5748E+05	0.3857E+05	0.3091E+05	0.1375E+06	0.5426E+05	0.2649E+05	0.3723E+05	0.2409E+06			
XTRGT	0.4491E+05	0.3013E+05	0.3242E+05	0.2275E+06	0.7220E+05	0.5353E+05	0.9849E+05	0.6228E+06			
----- OUTPUT FROM PERIOD 9 -----											
DLAMB	0.4255E-02	0.8154E-02	10.21	0.3189	0.3489E-01	1.454	14.74	0.2011	0.6728	0.1064E+05	5957.
X	0.0000	0.0000	5103.	0.2491E+05	0.1412E+05	8415.	0.1532E+05	0.6827E+05	0.2675E+05	0.1064E+05	5957.
X	0.8331E+05	0.2767E+06	0.2095E+05	0.866.	0.1246E+05	0.1954E+06	0.7656E+06	0.1275E+06	0.2897E+05	0.1064E+05	5957.
X	0.2548E+05	1881.									
FLAM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.6728	1.474	
FLAM	2.219	5.438	2.377	1.669	6.253	59.95	1.995	4.342			
EFFCST	2.958	4.355	2.484	3.214	2.746	4.102	4.379	4.342			
LASOR	1231.	1389.	0.5825E+05	0.8697E+05	8890.	6157.	6224.	0.9220E+05			
LAMPW	0.4134E-01	0.1214	4.155	0.3837	0.3756	5.228	47.94	0.6012			
TCUNS	0.2677E+05	0.1068E+05	0.8293E+05	0.2755E+06	0.2097E+05	8409.	0.1243E+05	0.1959E+06			

Figure A-1. Sample Listing of the Inputs and Outputs

R refers to the capital resource inventory (including labor). In this case there are eight capital items, each used in one of eight production industries. FLAM refers to the value of capital (propagated from the future or, in this case for the first past, set equal to the equilibrium value). The last eight values of FLAM are the values of the eight consumables. These values have meaning only for the output FLAM's. RTRGT, RFUT, and XTRGT refer to the target resource inventory, i.e., the inventory which will occur one gestation time in the future if no investment is made, and the corresponding target activity level. These parameters have meaning and are printed for phase one only. DLAMB is the rental value for capital (and labor). X refers to the activity level of all activities. EFFCST is the cost of production for the capital. LABOR is the amount of labor used in producing each commodity. LAMPRM (Δ) refers to the gross profit for each commodity which may be divided between capital and labor. Finally, TCONS is the total amount of each commodity used or consumed. It should be compared with the total amount X produced (elements 9 - 16 of the X array).

Once the model converges (or reaches the pass limit), the final results are printed. These include the inventory, activity level, shadow value and capital rental value trajectories. Cost of capital production, percentage difference with capital value, and Everett parameters (phase two) are also printed so the extent of convergence may be determined. A summary of the economy (gross consumption, investment, GNP, etc.) ends the output.

A.5.1 Sample Listing of Line Printer Output

THIS RUN CONTINUES FOR 40 TIME PERIODS OF LENGTH 1.00 YEARS

CAPITAL	CONSUMABLES	PROD ACT	CONSTR ACT	TOTAL ACT
14	13	25	16	43

BASIC PARAMETER ARRAYS:

LISTING OF ARRAY MINIMUM ACTIVITY:

C OF LEISURE	CBP-METALS	CBP-ENERGY	CBP-MACH BLDG	CBP-CHEMICALS	CBP-WOOD PROD	CBP-CNSTR MAT	CBP-POF SUPP/POF
0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
CBP-AGRICULTURE	CBP-TRANS/COMM	CBP-TRADE/SEK	CBP-MIL PROD	CBP-R. ESTATE	C BY MILITARY	C BY FED GOV'T	C BY LOC GOV'T
0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00

LISTING OF ARRAY B/C UTILITY-ELAS:

C OF LEISURE	CBP-METALS	CBP-ENERGY	CBP-MACH BLDG	CBP-CHEMICALS	CBP-WOOD PROD	CBP-CNSTR MAT	CBP-POF SUPP/POF
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
CBP-AGRICULTURE	CBP-TRANS/COMM	CBP-TRADE/SEK	CBP-MIL PROD	CBP-R. ESTATE	C BY MILITARY	C BY FED GOV'T	C BY LOC GOV'T
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

LISTING OF ARRAY WEIGHT FACTOR:

C OF LEISURE	CBP-METALS	CBP-ENERGY	CBP-MACH BLDG	CBP-CHEMICALS	CBP-WOOD PROD	CBP-CNSTR MAT	CBP-POF SUPP/POF
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
CBP-AGRICULTURE	CBP-TRANS/COMM	CBP-TRADE/SEK	CBP-MIL PROD	CBP-R. ESTATE	C BY MILITARY	C BY FED GOV'T	C BY LOC GOV'T
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

LISTING OF ARRAY BETA P-CONSUMBL:

P-METALS	P-ENERGY	P-MACH BLDG	P-CHEMICALS	P-WOOD PROD	P-CNSTR MAT	P-POF SUPP/POF	P-CONSTRUCTION
-1.750000	-1.270000	-2.250000	-1.500000	-2.500000	-2.500000	-2.000000	-3.000000
P-AGRICULTURE	P-TRANS/COMM	P-TRADE/SEK	P-MIL PROD	P-R. ESTATE			
-1.500000	-1.750000	-3.000000	-2.250000	-1.500000			

LISTING OF ARRAY BETA CAPL-TRADE:

NI-METALS	NI-ENERGY	NI-MACH BLDG	NI-CHEMICALS	NI-WOOD PROD	NI-CNSTR MAT	NI-POF SUPP/POF	NI-CONSTRUCTION
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
NI-AGRICULTURE	NI-TRANS/COMM	NI-TRADE/SEK	NI-MIL PROD	NI-R. ESTATE			
1.000000	1.000000	1.000000	1.000000	1.000000			

LISTING OF ARRAY FRACTIONAL RES:

NI-METALS	NI-ENERGY	NI-MACH BLDG	NI-CHEMICALS	NI-WOOD PROD	NI-CNSTR MAT	NI-POF SUPP/POF	NI-CONSTRUCTION
0.120000	0.442000	0.263000	0.192000	0.870000E-01	0.520000	0.352000	0.304000
NI-AGRICULTURE	NI-TRANS/COMM	NI-TRADE/SEK	NI-MIL PROD	NI-R. ESTATE	LABOR		
0.996000	0.315000	0.548000	0.236000	0.406000	0.700000		

BASIC DATA ARRAYS:

LISTING OF ARRAY TODAY'S ACTIVITY:

PK1-METALS	PK1-ENERGY	PK1-MACH BLDG	PK1-CHEMICALS	PK1-WOOD PROD	PK1-CNSTK MAT	PK1-FOP SUPPCKR FK1 CONSTRUCTION
3368.174	6692.506	5581.414	2144.235	1578.449	1870.597	3142.364
PK1-AGRICULTURE	PK1-TRANS/CUMH	PK1-TRADE/SER	PK1-MIL PROD	PK1-R. ESTATE	P-METALS	P-ENERGY
9996.506	6286.294	17794.85	842.3266	23830.52	40404.91	30762.36
P-CHEMICALS	P-WOOD PROD	P-CNSTK MAT	P-FOP SUPPCKR	P-CONSTRUCTION	P-AGRICULTURE	P-TRANS/CUMH
27729.20	21352.23	18232.98	172049.4	77308.13	109817.0	29530.00
P-MIL PROD	P-R. ESTATE	C OF LEISURE	CBP-METALS	CBP-ENERGY	CBP-MACH BLDG	CBP-CHEMICALS
36352.91	91873.59	348958.7	9.186252	3125.291	5522.560	2089.992
CBP-CNSTK MAT	CBP-FOP SUPPCKR	CBP-AGRICULTURE	CBP-TRANS/CUMH	CBP-TRADE/SER	CBP-MIL PROD	CBP-WOOD PROD
421.3950	95089.21	17947.57	2624.437	87239.61	206.0435	55275.20
C BY FED GOV'T	C BY LOC GOV'T	EXPORTS				C BY MILITARY
78411.25	75074.19	-18558.91				102188.8

LISTING OF ARRAY RESOURCES:

PK1-METALS	PK1-ENERGY	PK1-MACH BLDG	PK1-CHEMICALS	PK1-WOOD PROD	PK1-CNSTK MAT	PK1-FOP SUPPCKR FK1 CONSTRUCTION
34852.80	71004.32	53223.59	21584.71	12763.53	17371.99	32223.90
PK1-AGRICULTURE	PK1-TRANS/CUMH	PK1-TRADE/SER	PK1-MIL PROD	PK1-R. ESTATE	LABOR	
113391.0	64118.00	215288.4	7008.068	348501.0	697917.4	

58

LISTING OF ARRAY GESTATION TIME:

PK1-METALS	PK1-ENERGY	PK1-MACH BLDG	PK1-CHEMICALS	PK1-WOOD PROD	PK1-CNSTK MAT	PK1-FOP SUPPCKR FK1 CONSTRUCTION
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
PK1-AGRICULTURE	PK1-TRANS/CUMH	PK1-TRADE/SER	PK1-MIL PROD	PK1-R. ESTATE		
1.000000	1.000000	1.000000	1.000000	1.000000		

LISTING OF ARRAY DEPRECIATION:

PK1-METALS	PK1-ENERGY	PK1-MACH BLDG	PK1-CHEMICALS	PK1-WOOD PROD	PK1-CNSTK MAT	PK1-FOP SUPPCKR FK1 CONSTRUCTION
0.600245E-01	0.5497129E-01	0.6521018E-01	0.5987201E-01	0.8336248E-01	0.6792545E-01	0.5811834E-01
PK1-AGRICULTURE	PK1-TRANS/CUMH	PK1-TRADE/SER	PK1-MIL PROD	PK1-R. ESTATE	LABOR	
0.4909739E-01	0.5861691E-01	0.4376624E-01	0.7999998E-01	0.3000000E-01	-0.1600000E-01	

LISTING OF ARRAY GROWTH FACTOR:

PK1-METALS	PK1-ENERGY	PK1-MACH BLDG	PK1-CHEMICALS	PK1-WOOD PROD	PK1-CNSTK MAT	PK1-FOP SUPPCKR FK1 CONSTRUCTION
0.1997685E-01	0.1992554E-01	0.1995440E-01	0.1996031E-01	0.1994116E-01	0.1995176E-01	0.1992506E-01
PK1-AGRICULTURE	PK1-TRANS/CUMH	PK1-TRADE/SER	PK1-MIL PROD	PK1-R. ESTATE		
0.1994988E-01	0.1996408E-01	0.1996990E-01	0.1997046E-01	0.1996498E-01		

LISTING OF ARRAY DISCOUNTS:

PK1-METALS	PK1-ENERGY	PK1-MACH BLDG	PK1-CHEMICALS	PK1-WOOD PROD	PK1-CNSTK MAT	PK1-FOP SUPPCKR FK1 CONSTRUCTION
0.1475871	0.1011097	0.1734534	0.1761574	0.1109193	0.1130710	0.3346658
						0.2127662

P-AGRICULTURE	P-TRANS/COMM	P-TRADE/SER	P-MIL PROD	P-R. ESTATE
3.848476	0.9444051	1.286426	2.831336	0.1055926

C OF LEISURE	CBP METALS	CBP-ENERGY	CBP-MACH BLDG	CBP-CHEMICALS	CBP-WOOD PROD	CBP-CNSTR MAT	CBP-FOP SUPPORT
348958.7	9.186252	3125.291	5522.560	2089.992	3590.045	421.3950	95089.21

CBP AGRICULTURE	CBP-TRANS/COMM	CBP-TRADE/SER	CBP-MIL PROD	CBP-R. ESTATE	C BY MILITARY	C BY FED GOV'T	C BY LOC GOV'T
17947.57	2624.437	87239.61	206.0435	55275.20	102188.8	78411.25	75074.19

LISTING OF ARRAY UTILITY WEIGHT

NT METALS	NT-ENERGY	NT-MACH BLDG	NT-CHEMICALS	NT-WOOD PROD	NT-CNSTR MAT	NT-FOP SUPPORT	NT-CONSTRUCTION
0.2926358	0.2202518	0.3304380	0.3362996	0.3569658	0.2517576	0.6508700	0.4733970

NT AGRICULTURE	NT-TRANS/COMM	NT-TRADE/SER	NT-MIL PROD	NT-R. ESTATE	LABOR
0.1280503	0.2300740	0.2720989	0.2761840	0.1944268	1.000000

LISTING OF ARRAY RENTAL VALUE

P-METALS	P-ENERGY	P-MACH BLDG	P-CHEMICALS	P-WOOD PROD	P-CNSTR MAT	P-FOP SUPPORT	P-CONSTRUCTION
0.8966385	0.9858853	0.6785588	0.9291334	0.6372681	0.7297712	0.7607353	0.2480850

P-AGRICULTURE	P-TRANS/COMM	P-TRADE/SER	P-MIL PROD	P-R. ESTATE
0.8246236	0.8900484	0.2850784	0.8600607E-02	0.9840535

LISTING OF ARRAY FRACIN CAPITAL

POSITIVE GAMMA IMPLIES GAMMA FOR CAPITAL / NEGATIVE GAMMA IMPLIES GAMMA FOR LABOR

NT METALS	NT-ENERGY	NT-MACH BLDG	NT-CHEMICALS	NT-WOOD PROD	NT-CNSTR MAT	NT-FOP SUPPORT	NT-CONSTRUCTION
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

NT AGRICULTURE	NT-TRANS/COMM	NT-TRADE/SER	NT-MIL PROD	NT-R. ESTATE
1.000000	1.000000	1.000000	1.000000	1.000000

LISTING OF ARRAY CAPITAL FRACINS

NT METALS	NT-ENERGY	NT-MACH BLDG	NT-CHEMICALS	NT-WOOD PROD	NT-CNSTR MAT	NT-FOP SUPPORT	NT-CONSTRUCTION
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

NT AGRICULTURE	NT-TRANS/COMM	NT-TRADE/SER	NT-MIL PROD	NT-R. ESTATE
1.000000	1.000000	1.000000	1.000000	1.000000

LISTING OF ARRAY NORMALIZE (K-K)

P-METALS	P-ENERGY	P-MACH BLDG	P-CHEMICALS	P-WOOD PROD	P-CNSTR MAT	P-FOP SUPPORT	P-CONSTRUCTION
1.764912	0.6852305	2.673281	1.766022	2.466580	1.755182	7.181197	2.085403

P-AGRICULTURE	P-TRANS/COMM	P-TRADE/SER	P-MIL PROD	P-R. ESTATE
1.381725	0.7124543	1.633795	2.176102	0.2943166

LISTING OF ARRAY NORMALIZE (ACT)

UNIT VECTOR FOR STUDY ECONOMY

NOTE: POPULATION = 100 * (TODAY'S POPULATION) * EXP(0.015 * T)
 RE SOURCE'S, ETC., ARE NORMALIZED TO TODAY'S POPULATION

LISTING OF ARRAY UTILITY WEIGHTS

CBF METALS	CBF ENERGY	CBF MACH BLDG	CBF CHEMICALS	CBF WOOD PROD	CBF CONSTR MAT	CBF SUPP FOR
3489.87	31.75, 291	52,22, 560	2089, 992	3590, 045	421, 3950	95089, 21
CBF AGRICULTURE	CBF TRADE/SEK	CBF MIL PROD	CBF R. ESTATE	C BY MILITARY	C BY FED GOV'T	C BY LOC GOV'T
17947, 57	8239, 61	206, 0435	55275, 20	102188, 8	70411, 25	75074, 19

LISTING OF ARRAY INVENTORIES

NI METALS	NI ENERGY	NI MACH BLDG	NI CHEMICALS	NI WOOD PROD	NI CONSTR MAT	NI SUPP FOR
33454, 54	73531, 24	52991, 30	22865, 76	13379, 42	16911, 32	36276, 03
NI AGRICULTURE	NI TRANS/COMM	NI TRADE/SEK	NI MIL PROD	NI R. ESTATE	LABOR	
120742, 1	66271, 36	244155, 1	7762, 365	404983, 8	497917, 4	

LISTING OF ARRAY ACTIVITIES

PN1 METALS	PN1 ENERGY	PN1 MACH BLDG	PN1 CHEMICALS	PN1 WOOD PROD	PN1 CONSTR MAT	PN1 SUPP FOR
2582, 447	5292, 801	4368, 289	1757, 586	1348, 829	1441, 530	2724, 144
PN1 AGRICULTURE	PN1 TRANS/COMM	PN1 TRADE/SEK	PN1 MIL PROD	PN1 R. ESTATE	P-METALS	P-ENERGY
8481, 333	5051, 903	14838, 20	758, 0604	18952, 61	38436, 84	39046, 24
P-CHEMICALS	P-WOOD PROD	P-CONSTR MAT	P-POP SUPP FOR	P-CONSTRUCTION	P-AGRICULTURE	P-TRANS/COMM
20409, 47	21152, 92	16691, 57	186236, 9	68565, 36	118187, 2	29264, 96
P-MIL PROD	P-R. ESTATE	C OF LEISURE	CBP-METALS	CBP-ENERGY	CBP-MACH BLDG	CBP-CHEMICALS
30487, 19	106270, 4	345356, 4	10, 40350	3604, 783	6035, 782	2338, 897
CBP CONSTR MAT	CBP-POP SUPP FOR	CBP-AGRICULTURE	CBP-TRANS/COMM	CBP-TRADE/SEK	CBP-MIL PROD	CBP R. ESTATE
467, 2374	102658, 5	19013, 15	2956, 428	91658, 79	213, 8847	66366, 84
C BY FED GOV'T	C BY LOC GOV'T	EXPORTS				C BY MILITARY
85349, 49	79655, 17	-17843, 03				108082, 0

LISTING OF ARRAY CAPITAL VALUES

NI METALS	NI ENERGY	NI MACH BLDG	NI CHEMICALS	NI WOOD PROD	NI CONSTR MAT	NI SUPP FOR
0, 9219280	0, 9259411	0, 9222261	0, 9219158	0, 9205637	0, 9222541	0, 9225042
NI AGRICULTURE	NI TRANS/COMM	NI TRADE/SEK	NI MIL PROD	NI R. ESTATE	LABOR	
0, 9228258	0, 9206357	0, 9274817	0, 9249289	0, 9279127	1, 010431	

LISTING OF ARRAY CONSUMABLE VALUES

NI METALS	ENERGY	MACH BLDG	CHEMICALS	WOOD PROD	CONSTR MAT	POP SUPP FOR
0, 88, 9962	0, 8668045	0, 9149701	0, 8935801	0, 9166914	0, 9013075	0, 9262669
AGRICULTURE	TRANS/COMM	TRADE/SEK	MIL PROD	R. ESTATE		CONSTRUCTION
0, 9459, 66	0, 8877054	0, 9517866	0, 9633393	0, 8328737		0, 9298393

LISTING OF ARRAY RENTAL VALUES

NI METALS	NI ENERGY	NI MACH BLDG	NI CHEMICALS	NI WOOD PROD	NI CONSTR MAT	NI SUPP FOR
0, 1, 427942	0, 1791334	0, 2830599	0, 2815758	0, 2999498	0, 2066483	0, 5354940
NI AGRICULTURE	NI TRANS/COMM	NI TRADE/SEK	NI MIL PROD	NI R. ESTATE	LABOR	
0, 94019, 401	0, 1089505	0, 2255437	0, 2292777	0, 1559554	1, 010431	0, 4037497

Q RATIO	PH	PAIS	ENERGY	MACH BLDG	CHEMICALS	WOOD PROD	CNSIK MAT	FUP SUPPLFORI
0.01	0.305795E-01	0.3093104E-01	0.1912492E-01	0.4108076E-01	0.0008600E-01	0.2273637E-01	0.3700930E-01	0.3700930E-01
0.02	0.2714547E-01	0.6112959E-01	0.3784191E-01	0.7857974E-01	0.3956547E-01	0.4440676E-01	0.7000700E-01	0.7000700E-01
0.03	0.1359099	0.9049818E-01	0.5618015E-01	0.1132911	0.5852305E-01	0.6533165E-01	0.1003218	0.1003218
0.04	0.1737051	0.1190007	0.7415904E-01	0.1466066	0.7701012E-01	0.8551804E-01	0.1285312	0.1285312
0.05	0.2076055	0.1466246	0.9179434E-01	0.1760026	0.9506229E-01	0.1050739	0.1549945	0.1549945
0.06	0.2319267	0.1737111	0.1090997	0.2045586	0.1127078	0.1240554	0.1799443	0.1799443
0.07	0.2591473	0.1992504	0.260874	0.2315226	0.1299701	0.1425074	0.2036303	0.2036303
0.08	0.2845674	0.2242783	0.1427685	0.2570547	0.1468695	0.1604675	0.2261413	0.2261413
0.09	0.3084212	0.2484748	0.1591533	0.2812898	0.1634238	0.1779675	0.2476177	0.2476177
0.10	0.3308962	0.2718623	0.1752515	0.3043431	0.1796489	0.1950353	0.2681591	0.2681591
0.20	0.5047136	0.4667632	0.3222154	0.4869317	0.3264353	0.3462916	0.4360582	0.4360582
0.30	0.8276911	0.6071227	0.4480240	0.6138026	0.4509679	0.4711362	0.5600993	0.5600993
0.40	0.7117603	0.7108846	0.5572520	0.7084985	0.5587989	0.5773233	0.6579449	0.6579449
0.50	0.7833730	0.7897353	0.6531453	0.7824955	0.6535445	0.6694510	0.7382726	0.7382726
0.60	0.8403669	0.8511751	0.7381105	0.8422305	0.7377240	0.8225550	0.8059671	0.8059671
0.70	0.8892173	0.9001071	0.8139849	0.8916489	0.8131940	0.8327709	0.8641708	0.8641708
0.80	0.9311639	0.9398213	0.8822022	0.9333264	0.8813665	0.8076675	0.9149868	0.9149868
0.90	0.9677307	0.9725849	0.9439013	0.9690258	0.9433430	0.9464352	0.9599007	0.9599007
1.00	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2.00	1.197064	1.135740	1.370070	1.176536	1.384044	1.361219	1.253627	1.253627
3.00	1.296480	1.183995	1.567000	1.255965	1.598958	1.564077	1.386575	1.386575
4.00	1.358946	1.207956	1.690102	1.302056	1.738744	1.677337	1.471475	1.471475
5.00	1.402738	1.222049	1.774641	1.332469	1.837842	1.792821	1.531493	1.531493
6.00	1.435555	1.231238	1.836421	1.354177	1.912184	1.865187	1.576682	1.576682
7.00	1.461286	1.237659	1.883613	1.370519	1.970241	1.922241	1.612207	1.612207
8.00	1.482131	1.242377	1.920880	1.383306	2.016970	1.968564	1.641030	1.641030
9.00	1.499443	1.245976	1.951081	1.393607	2.055473	2.007042	1.664985	1.664985
10.00	1.514104	1.248803	1.976067	1.402099	2.087804	2.039592	1.685277	1.685277
20.00	1.593037	1.260755	2.099189	1.443910	2.255019	2.212508	1.794287	1.794287
30.00	1.67174	1.264327	2.145045	1.459740	2.321921	2.284696	1.841043	1.841043
40.00	1.677035	1.265992	2.169174	1.468337	2.358446	2.325422	1.88029	1.88029
50.00	1.669302	1.266939	2.184114	1.473592	2.382080	2.359138	1.885931	1.885931
60.00	1.669914	1.267545	2.194299	1.477300	2.398429	2.370734	1.898823	1.898823
70.00	1.67262	1.267963	2.201698	1.480031	2.410532	2.384034	1.908626	1.908626
80.00	1.683098	1.268267	2.207323	1.482133	2.419080	2.395847	1.916375	1.916375
90.00	1.687868	1.268497	2.211748	1.483804	2.427335	2.404715	1.922682	1.922682
100.00	1.691854	1.268677	2.215332	1.485167	2.433428	2.412077	1.927933	1.927933
Q RATIO	EDS INDUCTI	AGRICULTURE	TRANS/COMM	TRADE/SER	MIL PROD	R. ESTATE		
0.01	0.134045E-01	0.1063182E-01	0.2517589E-01	0.1241216E-01	0.1038182E-01	0.8515664		
0.02	0.2675385E-01	0.2126364E-01	0.4946949E-01	0.2480049E-01	0.2076362E-01	0.8768385		
0.03	0.4003594E-01	0.3189546E-01	0.7296128E-01	0.3715933E-01	0.3114540E-01	0.8911407		
0.04	0.5324931E-01	0.4252727E-01	0.9570714E-01	0.4948512E-01	0.4152709E-01	0.9010718		
0.05	0.6639167E-01	0.5315907E-01	0.1177529	0.6177514E-01	0.5190867E-01	0.9086509		
0.06	0.7946136E-01	0.6379084E-01	0.1391303	0.7402719E-01	0.6229006E-01	0.9147631		
0.07	0.9245713E-01	0.7442256E-01	0.1598986	0.8623942E-01	0.7267119E-01	0.9198744		
0.08	0.1053780	0.8505422E-01	0.1800655	0.9841024E-01	0.8305198E-01	0.9242602		
0.09	0.1182232	0.9568576E-01	0.1996680	0.1105383	0.9343333E-01	0.9280966		
0.10	0.1309923	0.1063172	0.2187457	0.1226223	0.1038121	0.9315078		
0.20	0.2544413	0.2126064	0.444658	0.2408483	0.2075503	0.9533128		
0.30	0.3707396	0.3187407	0.5156270	0.3539497	0.3110483	0.9655915		
0.40	0.4786918	0.4244128	0.6224355	0.4616830	0.4140534	0.9740905		
0.50	0.5802239	0.5290705	0.7113130	0.5640221	0.5162381	0.9805622		
0.60	0.6752909	0.6318704	0.7865494	0.6610573	0.6172130	0.9857723		
0.70	0.7643805	0.7317225	0.8511355	0.7529543	0.7165423	0.9901229		
0.80	0.8479152	0.8272739	0.9072730	0.8399207	0.8137672	0.9930516		
0.90	0.9263742	0.9171368	0.9564494	0.9221083	0.9084009	0.9971098		
1.00	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000		
2.00	1.543183	1.412257	1.261839	1.583648	1.608235	1.018440		
3.00	1.870662	1.479335	1.385542	1.936040	1.990761	1.028791		
4.00	2.084704	1.491015	1.458139	2.163531	2.124547	1.035933		
5.00	2.235020	1.497071	1.506039	2.319611	2.179090	1.041361		
6.00	2.343494	1.498151	1.540135	2.431691	2.270744	1.045725		
7.00	2.429760	1.499181	1.565649	2.515365	2.3221943	1.049364		

0.00	2.922762	1.492510	1.185491	2.572790	2.140631	1.074301
10.00	2.549190	1.499688	1.601377	2.630663	2.236059	1.055202
20.00	2.523876	1.499792	1.614391	2.671687	2.239633	1.057614
30.00	2.792487	1.499986	1.676882	2.854765	2.248531	1.072972
40.00	2.870298	1.499997	1.699490	2.911817	2.249534	1.081554
50.00	2.905353	1.499999	1.711248	2.938455	2.249793	1.087467
60.00	2.926058	1.500000	1.718483	2.953541	2.249890	1.091956
70.00	2.939641	1.500000	1.723394	2.963118	2.249934	1.095560
80.00	2.949196	1.500000	1.726953	2.968676	2.249958	1.098564
90.00	2.956260	1.500000	1.729652	2.974415	2.249971	1.101134
100.00	2.961683	1.500000	1.731773	2.977982	2.249979	1.103376
100.00	2.965968	1.500000	1.733483	2.980753	2.249985	1.105363

ENVIRONMENTAL ECONOMY:

*** RELATIVE STANDARD OF LIVING = 1.0037

*** TOTAL CONSUMPTION (EXCLUDING LEISURE) = 0.55467E106

*** TOTAL INVESTMENT = 70683.

*** GNP = 0.62536E106

*** TOTAL PRODUCTION = 0.11110E107

*** TOTAL CAPITAL INVENTORY = 0.11239E107

STARTING RESOURCES:

NOTE: RESOURCES ARE NORMALIZED TO TODAY'S POPULATION

LISTING OF ARRAY STARTING RES:

K1-MATLS	K1-ENERGY	K1-MACH BLDG	K1-CHEMICALS	K1-WOOD PROD	K1-CNSTR MAT	K1-FOP SUP(FORT K1-CONSTRUCTION
5900.059	44834.16	19996.86	5920.377	1586.325	12904.90	16204.02
161339.1	28853.10	168540.1	2362.720	301702.3	697917.4	10281.28

DSA ECONOMIC MODEL

FASS 1#	59.661 CPU SEC#	469 LOOF#	FOM = 1.00	0.475	1.00	
FASS 2#	60.317 CPU SEC#	482 LOOF#	FOM = 0.866E-02	0.256	1.00	
FASS 3#	59.771 CPU SEC#	465 LOOF#	FOM = 0.694E-02	0.188	1.00	
FASS 4#	65.052 CPU SEC#	517 LOOF#	FOM = 0.620E-02	0.198	1.00	
FASS 5#	56.433 CPU SEC#	441 LOOF#	FOM = 0.536E-02	0.167	1.00	
FASS 6#	58.743 CPU SEC#	465 LOOF#	FOM = 0.478E-02	0.156	1.00	

*** BEGINNING PHASE 2 ***

FASS 7#	105.233 CPU SEC#	824 LOOF#	FOM = 0.523E-01	0.148		0.650E-01
FASS 8#	61.427 CPU SEC#	404 LOOF#	FOM = 0.183E-01	0.204		0.529E-01
FASS 9#	61.179 CPU SEC#	468 LOOF#	FOM = 0.179E-01	0.136		0.548E-01
FASS 10#	64.086 CPU SEC#	503 LOOF#	FOM = 0.176E-01	0.148		0.660E-01
FASS 11#	63.935 CPU SEC#	503 LOOF#	FOM = 0.184E-01	0.137		0.573E-01
FASS 12#	54.609 CPU SEC#	442 LOOF#	FOM = 0.110E-01	0.112		0.519E-01
FASS 13#	52.799 CPU SEC#	414 LOOF#	FOM = 0.922E-02	0.110		0.510E-01
FASS 14#	54.357 CPU SEC#	433 LOOF#	FOM = 0.868E-02	0.110		0.416E-01
FASS 15#	55.591 CPU SEC#	435 LOOF#	FOM = 0.851E-02	0.896E-01		0.353E-01
FASS 16#	54.539 CPU SEC#	421 LOOF#	FOM = 0.943E-02	0.793E-01		0.399E-01
FASS 17#	53.405 CPU SEC#	424 LOOF#	FOM = 0.908E-02	0.686E-01		0.339E-01
FASS 18#	50.952 CPU SEC#	402 LOOF#	FOM = 0.702E-02	0.665E-01		0.382E-01
FASS 19#	54.573 CPU SEC#	435 LOOF#	FOM = 0.878E-02	0.655E-01		0.438E-01
FASS 20#	55.690 CPU SEC#	428 LOOF#	FOM = 0.102E-01	0.578E-01		0.368E-01
FASS 21#	64.183 CPU SEC#	510 LOOF#	FOM = 0.909E-02	0.643E-01		0.368E-01
FASS 22#	59.127 CPU SEC#	484 LOOF#	FOM = 0.609E-02	0.575E-01		0.379E-01
FASS 23#	57.655 CPU SEC#	463 LOOF#	FOM = 0.607E-02	0.581E-01		0.339E-01
FASS 24#	55.207 CPU SEC#	432 LOOF#	FOM = 0.546E-02	0.486E-01		0.333E-01
FASS 25#	55.068 CPU SEC#	431 LOOF#	FOM = 0.548E-02	0.403E-01		0.295E-01
FASS 26#	58.475 CPU SEC#	458 LOOF#	FOM = 0.570E-02	0.301E-01		0.263E-01
FASS 27#	57.077 CPU SEC#	459 LOOF#	FOM = 0.464E-02	0.311E-01		0.264E-01

PASS	201	59,599 CPU SEC	480 LOUFS/ FOM =	0.408E-02	0.557E-01	0.323E-01
PASS	291	62,530 CPU SEC	497 LOUFS/ FOM =	0.592E-02	0.319E-01	0.220E-01
PASS	301	63,434 CPU SEC	428 LOUFS/ FOM =	0.315E-02	0.245E-01	0.192E-01
PASS	311	66,914 CPU SEC	537 LOUFS/ FOM =	0.399E-02	0.231E-01	0.207E-01
PASS	321	51,180 CPU SEC	403 LOUFS/ FOM =	0.263E-02	0.228E-01	0.206E-01
PASS	331	57,631 CPU SEC	459 LOUFS/ FOM =	0.369E-02	0.193E-01	0.212E-01
PASS	341	51,821 CPU SEC	411 LOUFS/ FOM =	0.275E-02	0.264E-01	0.261E-01
PASS	351	59,167 CPU SEC	474 LOUFS/ FOM =	0.357E-02	0.324E-01	0.256E-01
PASS	361	59,473 CPU SEC	454 LOUFS/ FOM =	0.345E-02	0.228E-01	0.331E-01
PASS	371	55,620 CPU SEC	429 LOUFS/ FOM =	0.347E-02	0.227E-01	0.165E-01
PASS	381	55,843 CPU SEC	448 LOUFS/ FOM =	0.289E-02	0.182E-01	0.119E-01
PASS	391	52,194 CPU SEC	417 LOUFS/ FOM =	0.204E-02	0.174E-01	0.130E-01
PASS	401	54,408 CPU SEC	422 LOUFS/ FOM =	0.233E-02	0.191E-01	0.140E-01
PASS	411	45,198 CPU SEC	348 LOUFS/ FOM =	0.176E-02	0.168E-01	0.169E-01
PASS	421	57,591 CPU SEC	448 LOUFS/ FOM =	0.213E-02	0.154E-01	0.224E-01
PASS	431	56,214 CPU SEC	437 LOUFS/ FOM =	0.320E-02	0.173E-01	0.140E-01
PASS	441	58,329 CPU SEC	452 LOUFS/ FOM =	0.198E-02	0.139E-01	0.117E-01
PASS	451	55,960 CPU SEC	432 LOUFS/ FOM =	0.156E-02	0.141E-01	0.980E-02
PASS	461	54,656 CPU SEC	433 LOUFS/ FOM =	0.145E-02	0.135E-01	0.115E-01
PASS	471	53,012 CPU SEC	428 LOUFS/ FOM =	0.223E-02	0.144E-01	0.114E-01
PASS	481	57,192 CPU SEC	446 LOUFS/ FOM =	0.149E-02	0.143E-01	0.953E-02
PASS	491	52,386 CPU SEC	405 LOUFS/ FOM =	0.192E-02	0.104E-01	0.821E-02
PASS	501	56,532 CPU SEC	430 LOUFS/ FOM =	0.109E-02	0.114E-01	0.696E-02
PASS	511	55,137 CPU SEC	429 LOUFS/ FOM =	0.133E-02	0.110E-01	0.722E-02
PASS	521	53,612 CPU SEC	401 LOUFS/ FOM =	0.123E-02	0.906E-02	0.815E-02

FINAL MODEL RESULTS:

CONVENIENCE ON PASS 52

LISTING OF ARRAY INVENTORY:

PER	1	N1-METALS	N1-ENERGY	N1-MACH BLDG	N1-CHEMICALS	K1-WOOD FROG	K1-CNSTR MAT	N1-FUP SUP/FORT
PER	2	59597.227	45283.74	20197.37	5979.759	1602.221	13034.29	16366.50
PER	3	9370.546	44143.25	19588.15	7505.920	2227.387	12623.96	15929.22
PER	4	14525.34	41118.94	21304.19	10017.25	3729.991	11607.73	14791.27
PER	5	18168.41	41299.85	26514.93	11840.16	5189.048	10673.30	16395.72
PER	6	21494.77	44832.60	32206.58	13397.71	6442.528	9814.100	19385.91
PER	7	24379.01	49997.17	37412.10	15039.87	7396.271	10342.15	22054.57
PER	8	26548.44	55298.25	40630.37	16460.99	8649.779	12629.55	24627.81
PER	9	28306.08	59417.26	43012.95	17637.19	10159.92	15090.41	25916.59
PER	10	30485.44	62832.38	45601.81	18577.53	10883.53	16460.52	26733.05
PER	11	31987.94	65012.09	47660.84	19258.08	11416.63	16988.19	27583.79
PER	12	32506.15	66718.60	49246.97	19837.88	11943.75	17404.94	28650.63
PER	13	32967.20	68374.79	50138.30	20388.83	12249.77	17514.80	29761.33
PER	14	33309.08	69545.15	50829.71	20912.09	12502.34	17632.87	30715.88
PER	15	33547.56	70342.12	51466.79	21291.70	12677.41	17601.78	31634.59
PER	16	33542.73	71438.36	51904.85	21561.98	12794.53	17457.71	32563.25
PER	17	33568.70	71725.54	52099.16	21828.81	12900.64	17348.50	33261.74
PER	18	33372.66	71997.16	52358.46	22048.19	12968.78	1736.38	33751.75
PER	19	33105.33	72338.39	52527.13	22211.67	13035.28	17094.40	34211.35
PER	20	33219.41	72597.29	52549.64	22337.40	13089.98	16969.88	34427.33
PER	21	33259.56	72755.68	52676.83	22512.99	13098.86	16872.30	34768.46
PER	22	33251.56	72747.46	52739.83	22576.41	13189.88	16801.26	35159.89
PER	23	33288.85	72983.58	52794.94	22601.90	13266.63	16783.76	35327.99
PER	24	33317.11	73256.89	52796.49	22671.17	13338.52	16838.97	35446.92
PER	25	33359.68	73275.66	52743.17	22743.28	13333.29	16902.07	35472.47
PER	26	33317.32	73282.04	52835.45	22712.05	1334.00	16897.49	35520.74
PER	27	33217.67	73327.51	52836.49	22733.45	13243.03	16866.55	35622.63
PER	28	33168.01	73334.63	52635.65	22780.58	13256.93	16867.24	35620.33
PER	29	33142.62	73244.12	52489.41	22601.25	13250.99	16824.50	35721.92
PER	30	33101.07	73106.14	52459.58	22670.94	13231.27	16742.81	35814.22
PER	31	33097.52	7332.79	52300.21	22618.48	13225.41	16671.13	35805.90
PER	32	33092.32	73193.90	52391.89	22785.03	13232.62	16614.56	35714.13
PER	33	33033.62	73272.76	52217.76	22625.00	13137.57	16571.17	35638.54
PER	34	33004.77	73100.83	52005.66	22459.72	13209.19	16560.95	35755.93

[illegible]

PER 14	2461.711	5613.167	4594.006	1711.067	1413.600	1355.946	3346.111
PER 15	2673.114	5678.773	4475.949	1775.481	1396.953	1418.850	3040.715
PER 16	2566.142	5237.335	4348.085	1855.585	1354.583	1304.924	3001.365
PER 17	2313.238	5671.896	4790.030	1888.418	1404.932	1333.450	3050.738
PER 18	2541.300	5434.951	4219.683	1803.381	1345.623	1318.623	2566.786
PER 19	2655.638	5541.712	4491.511	1829.096	1317.387	1367.958	3324.162
PER 20	2563.318	5267.439	4451.216	1811.103	1517.031	1359.405	2748.922
PER 21	2558.577	5209.971	4377.360	1794.366	1275.402	1436.091	2898.784
PER 22	2655.717	5769.690	4442.120	1741.512	1353.192	1455.014	2677.456
PER 23	2590.534	5339.876	4377.761	1886.928	1343.309	1523.127	2714.988
PER 24	2607.419	5266.915	4322.052	1721.877	1319.362	1483.415	2746.205
PER 25	2460.093	5314.355	4572.855	1757.488	1353.505	1391.608	2678.137
PER 26	2476.864	5350.342	4157.727	1789.061	1336.485	1424.142	2798.012
PER 27	2549.128	5242.737	4137.851	1815.212	1364.719	1453.449	2781.879
PER 28	2532.298	5145.633	4237.691	1742.358	1301.155	1335.312	2793.238
PER 29	2512.485	5542.029	4359.568	1813.598	1331.757	1360.638	2587.632
PER 30	2591.123	5136.470	4385.390	1702.121	1327.628	1344.696	2611.770
PER 31	2513.014	5109.491	4470.229	1744.890	1357.632	1377.007	2608.445
PER 32	2476.604	5603.889	4517.228	1850.722	1325.788	1365.067	2837.761
PER 33	2779.231	5034.142	4333.796	1743.978	1287.633	1438.093	2704.890
PER 34	2653.610	5482.030	4315.937	1731.931	1370.142	1429.225	2630.724
PER 35	2638.805	5498.794	4453.483	1716.697	1370.856	1556.947	2747.647
PER 36	2806.935	5125.977	4238.545	1887.176	1311.429	1607.748	2651.318
PER 37	2352.140	5314.118	4150.925	1767.283	1508.834	1463.219	2609.338
PER 38	2300.060	5445.210	4242.971	1686.740	1248.056	1464.698	2758.435
PER 39	2305.949	4793.508	4468.048	1732.160	1334.631	1308.048	2081.467
PER 40	2618.885	5132.333	4269.672	1649.452	1293.953	1412.410	2642.489

	PK1-CONSTRUCTIO	PK1-AGRICULTURE	PK1-TRANS/COMM	PK1-TRADE/SER	PK1-MIL PROD	PK1-K. ESTATE	F-ME IAS
PER 1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	11648.89
PER 2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	1510.297	0.000000E+00	17221.76
PER 3	0.000000E+00	0.000000E+00	8626.304	0.000000E+00	737.2565	0.000000E+00	24158.20
PER 4	6424.879	0.000000E+00	9058.523	0.000000E+00	810.8573	0.000000E+00	27262.75
PER 5	5295.643	0.000000E+00	9449.494	8528.841	886.6762	0.000000E+00	30708.79
PER 6	6929.112	0.000000E+00	8713.880	22074.36	971.0863	6807.323	34008.27
PER 7	5851.199	0.000000E+00	7610.626	21544.04	878.1558	20967.55	36153.29
PER 8	4301.569	0.000000E+00	8461.251	23852.26	878.3019	31842.42	37948.17
PER 9	3876.049	6038.481	6103.678	21382.29	937.9410	35656.55	39171.97
PER 10	3279.707	9839.836	6097.504	22150.72	897.8966	37461.27	40041.09
PER 11	3417.939	11114.69	5968.939	20424.73	905.0095	3122.40	40117.72
PER 12	2882.438	11280.43	5630.320	20147.62	860.9176	29423.30	40133.07
PER 13	3205.403	9899.318	5697.405	19342.45	794.9481	2830.98	39789.69
PER 14	2926.936	11432.26	5591.344	17543.00	835.4207	26733.64	39732.78
PER 15	2971.017	9610.522	5237.468	16280.96	836.9086	25474.71	39468.05
PER 16	3011.602	10655.57	5199.990	17401.56	788.6476	24034.28	39332.85
PER 17	2804.949	9318.773	5232.689	12300.84	796.7430	23476.12	39083.31
PER 18	2990.078	10078.68	4899.978	15543.05	776.8997	22140.89	38976.56
PER 19	2909.318	8771.555	4995.693	16105.99	811.9257	20761.67	38786.15
PER 20	2854.406	8653.155	5666.093	15493.63	729.6087	20718.13	38739.76
PER 21	3703.701	9662.922	4897.079	13335.58	778.9035	19522.44	38712.24
PER 22	2996.481	8526.791	5321.386	15155.04	716.6491	19568.70	38607.73
PER 23	3195.791	8324.465	5303.670	15308.59	786.6342	19451.84	38582.72
PER 24	3030.276	9457.175	4918.498	14747.03	780.5433	19573.39	38573.89
PER 25	3132.575	8549.341	4952.294	15069.93	734.6480	19706.11	38539.17
PER 26	3035.806	8732.778	4952.703	15270.15	763.3790	19468.46	38525.61
PER 27	3018.535	8640.001	5017.313	15352.56	756.7076	19042.06	38498.35
PER 28	3076.018	8548.108	4957.986	15274.54	756.4150	19031.51	38414.47
PER 29	2942.195	8527.538	5040.947	14219.05	746.0174	19463.51	38374.37
PER 30	2970.650	8118.474	5022.881	14478.13	739.0350	19700.91	38328.14
PER 31	2875.561	8352.075	5018.795	14677.06	751.7112	19161.26	38286.51
PER 32	2960.703	7943.214	5077.947	15219.50	770.1917	19271.69	38324.14
PER 33	3125.703	9060.173	4976.672	14173.50	730.7461	18456.08	38367.10
PER 34	2965.978	9198.379	4936.942	14494.15	732.8301	17920.97	38382.41
PER 35	3205.248	7951.529	5176.404	14632.09	748.7919	18183.09	38364.58
PER 36	3262.540	8476.172	5141.541	14519.50	763.2985	18298.61	38451.49
PER 47	3432.439	8480.579	5481.045	14380.41	772.9472	19293.15	38489.91

69

FEB 1	1801.136	819.48.38	96.11132	5201.790	49810.74	62760.90	45551.61
FEB 2	797.7061	82703.83	118.1919	4627.22	63381.40	65795.24	55799.67
FEB 3	792.1809	79839.82	149.5228	43580.44	72433.93	65132.08	56451.08
FEB 4	1162.746	75799.58	161.9096	38143.48	77036.34	61120.51	57465.05
FEB 5	1439.630	72862.45	172.0630	35993.25	79817.95	60861.17	58592.08
FEB 6	1654.085	73968.13	180.6901	35702.12	83901.60	62146.69	60877.59
FEB 7	1769.212	76512.31	185.9557	38105.66	87438.13	65053.80	63521.51
FEB 8	1985.278	79080.72	191.0035	42059.10	91204.66	68554.79	66428.75
FEB 9	2172.959	81374.13	194.9986	45912.85	94043.84	71635.47	68790.67
FEB 10	2287.938	83202.82	198.9083	49202.59	96826.90	74179.05	70731.35
FEB 11	2400.797	84798.53	201.7582	52104.38	98708.75	76324.06	72383.10
FEB 12	2510.608	86080.53	204.5802	54634.67	100710.7	78046.99	73825.50
FEB 13	2584.549	87104.35	205.8422	56913.34	101968.6	79569.68	74880.77
FEB 14	2659.514	88019.70	207.4689	58686.74	103217.8	80661.85	75768.74
FEB 15	2714.027	88613.31	208.5545	60163.33	103997.0	81614.68	76427.77
FEB 16	2757.923	89185.53	209.5670	61250.50	104800.5	82368.90	77059.73
FEB 17	2789.588	89228.27	210.2045	62065.60	105407.0	82893.38	77528.12
FEB 18	2810.276	90056.05	210.7911	63248.08	105864.3	83309.79	77871.07
FEB 19	2822.230	90301.07	211.2293	63681.21	106181.6	83443.73	78100.74
FEB 20	2858.036	90524.00	211.5981	64169.43	106474.4	83688.56	78373.60
FEB 21	2881.456	90616.94	211.8195	64362.84	106681.5	84201.07	78477.79
FEB 22	2892.995	90802.15	212.0493	64594.14	106902.2	84409.60	78688.54
FEB 23	2916.330	90832.84	212.1431	64927.42	107000.4	84581.22	78788.65
FEB 24	2914.367	90890.95	212.3869	64960.13	107094.2	84707.51	78829.87
FEB 25	2914.367	90917.43	212.4825	65128.50	107142.4	84802.73	78864.79
FEB 26	2911.021	91015.43	212.4827	65185.85	107212.2	84931.53	78977.90
FEB 27	2910.993	91113.70	212.5658	65171.61	107294.6	84957.42	79058.39
FEB 28	2913.212	91175.20	212.5672	65146.67	107337.7	84937.69	79068.16
FEB 29	2908.123	91249.13	212.6838	65146.67	107438.4	84931.53	79104.80
FEB 30	2914.881	91202.01	212.7871	65185.85	107442.7	84957.42	79115.73
FEB 31	2926.262	91210.17	212.7523	65171.61	107404.7	84937.69	79130.41
FEB 32	2929.874	91248.45	212.8244	65146.67	107406.9	84937.69	79079.48
FEB 33	2918.276	91178.66	212.7439	65122.02	107406.9	84937.69	79079.48
FEB 34	2924.771	91193.58	212.7009	65109.43	107415.5	84937.69	79079.48
FEB 35	2921.297	91163.87	212.6995	65055.45	107395.5	84733.93	79149.01
FEB 36	2933.637	91056.11	212.7208	65109.43	107387.0	84712.38	79104.65
FEB 37	2937.553	91094.33	212.5046	65337.60	107331.1	84784.24	79089.47
FEB 38	2937.936	91337.30	212.9599	65512.82	107560.2	85000.45	79070.61
FEB 39							79236.45
FEB 40							

EXPORTS

FEB 1	-5546.577
FEB 2	-8059.942
FEB 3	-11395.31
FEB 4	-13163.71
FEB 5	-14743.12
FEB 6	-16063.21
FEB 7	-16543.40
FEB 8	-16938.24
FEB 9	-17421.73
FEB 10	-17939.42
FEB 11	-18034.62
FEB 12	-18135.64
FEB 13	-18039.39
FEB 14	-18093.89
FEB 15	-18003.34
FEB 16	-18023.50
FEB 17	-17934.53
FEB 18	-17933.74
FEB 19	-17901.54
FEB 20	-17880.95
FEB 21	-17930.44
FEB 22	-17868.91
FEB 23	-17873.57
FEB 24	-17876.01
FEB 25	-17849.03

PER 29 1.041.56
 PER 27 1.039.10
 PER 26 1.017.17
 PER 29 1.770.16
 PER 30 1.733.63
 PER 31 1.770.75
 PER 32 1.795.77
 PER 33 1.820.32
 PER 34 1.844.30
 PER 35 1.857.97
 PER 36 1.865.37
 PER 37 1.868.47
 PER 38 1.801.73
 PER 39 1.782.21
 PER 40 1.757.47

LISTING OF ARRAY SHADOW VAL T-01

NI METALS NI ENERGY NI-MACH BLDG NI-CHEMICALS NI-WOOD PROD NI-CNSTR MAT NI-POF SUPPRT NI-CNSTRUCTION
 40.2914 1.760/94 2.915018 14.72885 19.94475 0.9982331 2.391352 1.613506
 0.693178 NI-TRANS/COMM NI-TRD/SEK NI-R. ESTATE
 3.059422 1.314283 4.245614 1.413736

LISTING OF ARRAY SHADOW VALUE1

PER 1 NI-METALS NI-ENERGY NI-MACH BLDG NI-CHEMICALS NI-WOOD PROD NI-CNSTR MAT NI-POF SUPPRT
 PER 2 9.641700 2.055558 3.540085 9.469913 10.47888 1.201736 2.969617
 PER 3 3.632661 2.247988 3.383217 3.634356 3.834554 1.413231 3.038299
 PER 4 2.357656 2.195917 2.541117 2.501052 2.515344 1.618826 2.491089
 PER 5 1.977317 1.943936 1.998459 2.050480 2.167924 1.758078 2.007953
 PER 6 1.730185 1.644158 1.674015 1.743512 1.871439 1.709659 1.696089
 PER 7 1.540540 1.467848 1.474463 1.540494 1.576531 1.531843 1.500383
 PER 8 1.365180 1.337837 1.351277 1.395385 1.386527 1.374347 1.375419
 PER 9 1.238789 1.243915 1.283411 1.291506 1.274590 1.261979 1.276840
 PER 10 1.149789 1.174018 1.189667 1.215114 1.188169 1.178894 1.200367
 PER 11 1.093548 1.116961 1.126368 1.153164 1.124754 1.116997 1.145295
 PER 12 1.047836 1.074328 1.077914 1.102802 1.081452 1.071976 1.103605
 PER 13 1.016681 1.044812 1.039974 1.064466 1.048085 1.037623 1.067388
 PER 14 0.987183 1.023040 1.012918 1.037435 1.023772 1.014733 1.034246
 PER 15 0.9791525 1.005319 0.9920356 1.015233 1.003987 0.9978882 1.009914
 PER 16 0.9750649 0.9908475 0.9755847 0.9960210 0.9868224 0.9832604 0.9940414
 PER 17 0.9727947 0.9787900 0.9614166 0.9813804 0.9730925 0.9731271 0.9821271
 PER 18 0.9671518 0.9684575 0.9513307 0.9706903 0.9628584 0.9649859 0.9726381
 PER 19 0.9608144 0.9603655 0.943093 0.9626749 0.9553191 0.9580521 0.9635455
 PER 20 0.9555358 0.9546262 0.9396369 0.9566892 0.9482164 0.9521278 0.9532508
 PER 21 0.9516311 0.9502783 0.9348819 0.9533721 0.9437807 0.9475569 0.9468737
 PER 22 0.9493576 0.9456033 0.9310943 0.9487793 0.9430070 0.9436080 0.9447092
 PER 23 0.9445473 0.9414929 0.9294047 0.9446667 0.9425249 0.9194177 0.9436037
 PER 24 0.9494042 0.9397257 0.9278141 0.9409838 0.9419076 0.9368563 0.9429250
 PER 25 0.9308827 0.9377910 0.9262788 0.9387092 0.9412800 0.9356940 0.9417521
 PER 26 0.9307824 0.9368691 0.9258996 0.9359986 0.9394220 0.9353684 0.9404135
 PER 27 0.9495520 0.9363667 0.9263667 0.9324834 0.9375083 0.9360687 0.9387383
 PER 28 0.9485866 0.9363773 0.9261796 0.9363536 0.9380670 0.9372451 0.9372451
 PER 29 0.9475736 0.9355599 0.9252824 0.9276116 0.9361387 0.9398155 0.9370004
 PER 30 0.9468881 0.9347975 0.9253694 0.9266615 0.9358370 0.9398445 0.9379401
 PER 31 0.9460193 0.9352001 0.9265326 0.9285602 0.9350100 0.9386528 0.9389701
 PER 32 0.9425270 0.9352473 0.9271712 0.9259417 0.9404777 0.9375881 0.9382335
 PER 33 0.9389331 0.9346855 0.9272434 0.9250425 0.9340634 0.9362971 0.9371854
 PER 34 0.9331103 0.9343929 0.9276884 0.9259218 0.9344903 0.9341680 0.9379124
 PER 35 0.9316455 0.9338929 0.9284914 0.9271679 0.9332946 0.9315279 0.9386055
 PER 36 0.9317464 0.9340645 0.9289096 0.9266451 0.9323190 0.9298215 0.9377779
 PER 37 0.9316678 0.9348090 0.9284118 0.9257007 0.9314888 0.9298672 0.9367931
 PER 38 0.9316678 0.9348090 0.9284118 0.9257007 0.9314888 0.9298672 0.9367931
 PER 39 0.9316678 0.9348090 0.9284118 0.9257007 0.9314888 0.9298672 0.9367931
 PER 40 0.9316678 0.9348090 0.9284118 0.9257007 0.9314888 0.9298672 0.9367931

11K 20	0.9240705	0.9052025	0.9100033	0.9300814	0.9455201	0.9452210	0.9724565
11K 21	0.9106177	0.9000518	0.9157407	0.9241515	0.9368071	0.9472144	0.9472470
11K 22	0.9170256	0.9009102	0.9214964	0.9211617	0.9310734	0.9268761	0.9445567
11K 23	0.9070643	0.8990011	0.9149648	0.9191771	0.9231704	0.9217006	0.9416530
11K 24	0.9018178	0.8977046	0.9217125	0.9133989	0.9299306	0.9166070	0.9416036
11K 25	0.8991773	0.8931403	0.9156073	0.9114721	0.9216263	0.9146545	0.9407388
11K 26	0.9017758	0.8929037	0.9142839	0.9129672	0.9215887	0.9141077	0.9393607
11K 27	0.9049807	0.8980879	0.9140095	0.9109493	0.9202575	0.9126594	0.9381869
11K 28	0.9030091	0.8978119	0.9253195	0.9081288	0.9284387	0.9106576	0.9368903
11K 29	0.9014531	0.8932935	0.9156244	0.9067378	0.9282103	0.9139818	0.9366765
11K 30	0.9013059	0.89773495	0.9229001	0.9043348	0.9284712	0.9155406	0.9367715
11K 31	0.8994612	0.8942725	0.9223054	0.9026880	0.9276173	0.9152712	0.9367034
11K 32	0.9013639	0.8976818	0.9244471	0.9055920	0.9278382	0.9141510	0.9380515
11K 33	0.9017355	0.8977348	0.9246085	0.9033688	0.9254340	0.9156080	0.9383668
11K 34	0.9051302	0.89773937	0.9240270	0.9005142	0.9270146	0.9151416	0.9369214
11K 35	0.8976415	0.8978409	0.9228192	0.9030981	0.9286710	0.9155384	0.9370801
11K 36	0.8951146	0.89739890	0.9234385	0.9045179	0.9255825	0.9101802	0.9364046
11K 37	0.8912260	0.8957036	0.9231964	0.9026380	0.9277208	0.9098784	0.9366085
11K 38	0.8909295	0.8978012	0.9213285	0.8996095	0.9261348	0.9106176	0.9375618
11K 39	0.8959241	0.8964193	0.9234553	0.9027629	0.9258201	0.9121677	0.9380891
11K 40	0.8989947	0.8958867	0.9226192	0.9039317	0.9239050	0.9088833	0.9358476

K. ESTATE

MIL FROM

TRADE/SER

TRANS/COMM

AGRICULTURE

CONSTRUCTION

11K 1	0.9243801	3.732946	1.345105	1.391321	1.350495	6.1098920	0.9724565
11K 2	1.062619	2.143801	1.064698	1.457101	1.254677	6.675949	0.9472470
11K 3	1.182934	1.743296	1.054844	1.579101	1.254677	6.675949	0.9445567
11K 4	1.268349	1.378008	1.092820	3.289980	0.9678129	1.876932	0.9416530
11K 5	1.347935	1.272584	1.150925	3.312093	0.9172756	1.796021	0.9407388
11K 6	1.441580	1.272584	1.150925	2.257103	0.9071299	1.615955	0.9393607
11K 7	1.140314	1.197319	1.179422	1.822994	0.9153048	1.439641	0.9381869
11K 8	1.548233	1.108025	1.140204	1.483393	0.9168002	1.333799	0.9368903
11K 9	1.450577	1.078742	1.103172	1.321949	0.9735143	1.256123	0.9366765
11K 10	1.314227	1.056641	1.072080	1.207771	1.022201	1.173708	0.9367715
11K 11	1.203916	1.035882	1.048517	1.147075	1.060614	1.119779	0.9367034
11K 12	1.123421	1.021240	1.028787	1.093152	1.061480	1.079659	0.9380515
11K 13	1.060855	1.007152	1.013465	1.045339	1.046285	1.049117	0.9383668
11K 14	1.011724	1.000978	1.001533	1.015433	1.024581	1.021389	0.9369214
11K 15	0.9712170	0.9931390	0.9911374	0.9868109	0.9992469	0.9913258	0.9370801
11K 16	0.9418686	0.9879600	0.9844978	0.9669899	0.9809064	0.9799048	0.9364046
11K 17	0.9187523	0.9831865	0.9781812	0.9515992	0.9804200	0.9680368	0.9366085
11K 18	0.9024448	0.9802051	0.9722645	0.9407974	0.9726711	0.9640922	0.9375618
11K 19	0.8905932	0.9774773	0.9687257	0.9338717	0.945677	0.9590610	0.9380891
11K 20	0.8807415	0.9754496	0.9660972	0.9299160	0.9580896	0.9565867	0.9358476
11K 21	0.8739427	0.9737491	0.9637181	0.9182660	0.9582895	0.9512812	
11K 22	0.8679986	0.9727315	0.9627296	0.9108025	0.9560833	0.9512812	
11K 23	0.8651035	0.9716772	0.9607659	0.9071696	0.9533257	0.9460664	
11K 24	0.8613946	0.9712479	0.9604413	0.8999108	0.9529060	0.9430846	
11K 25	0.8588061	0.9701359	0.9598272	0.9005171	0.9510219	0.9479742	
11K 26	0.8557309	0.9696963	0.9595477	0.9015520	0.9497399	0.9434431	
11K 27	0.8516128	0.9696953	0.9585145	0.9015608	0.9486115	0.9401776	
11K 28	0.8513384	0.9693163	0.9574807	0.9008739	0.9477027	0.9393957	
11K 29	0.8509096	0.9693101	0.9566250	0.9024506	0.9483871	0.9410099	
11K 30	0.8487059	0.9687786	0.9560596	0.9003581	0.9484709	0.9411712	
11K 31	0.8447976	0.9683084	0.9565536	0.8987202	0.9483781	0.9403457	
11K 32	0.8419443	0.9684666	0.9564681	0.8968565	0.9490503	0.9397607	
11K 33	0.8433827	0.9684725	0.9562108	0.8957509	0.9514087	0.9411362	
11K 34	0.8437072	0.9681384	0.9560667	0.8950378	0.9505591	0.9402792	
11K 35	0.8461955	0.9685049	0.9567985	0.8993106	0.9479652	0.9405573	
11K 36	0.8481485	0.9687008	0.9566420	0.8973138	0.9470748	0.9380069	
11K 37	0.8496629	0.9687070	0.9569538	0.8983808	0.9483457	0.9390399	
11K 38	0.8489584	0.9686100	0.9577707	0.8946017	0.9489451	0.9377598	
11K 39	0.8459938	0.9695954	0.9576843	0.8934092	0.9497742	0.9381409	
11K 40	0.8437310	0.9675224	0.9551367	0.8932928	0.9499862	0.9379847	

LISTING 11.10 *Continued*

[illegible]

NR 23	0.414001	0.92273030	0.1941517	0.2324913	0.2304734	0.1620494	1.007400
NR 24	0.414696	0.90059768	0.1901991	0.2319911	0.2325945	0.1616091	1.006172
NR 25	0.415737	0.9726306E-01	0.1904612	0.2311663	0.2341191	0.1609971	1.006635
NR 26	0.4140992	0.9559097E-01	0.1909950	0.2310437	0.2335693	0.1602853	1.006939
NR 27	0.4074995	0.9530057E-01	0.1910443	0.2300195	0.2337706	0.1594290	1.006076
NR 28	0.4077040	0.9470754E-01	0.1908291	0.2280238	0.2331222	0.1594323	1.006917
NR 29	0.4097171	0.9515066E-01	0.1913804	0.2275096	0.2323394	0.1592946	1.007237
NR 30	0.4105003	0.9531271E-01	0.1906737	0.2270560	0.2319196	0.1587663	1.006647
NR 31	0.4105511	0.9554273E-01	0.1899125	0.2285101	0.2327629	0.1577577	1.006917
NR 32	0.4086917	0.9602850E-01	0.1892097	0.2281823	0.2326880	0.1570282	1.006409
NR 33	0.4118473	0.9764940E-01	0.1886633	0.2278365	0.2319424	0.1571424	1.006596
NR 34	0.4099997	0.9729605E-01	0.1888338	0.2277021	0.2324719	0.1574265	1.006487
NR 35	0.4104518	0.9511524E-01	0.1902072	0.2284090	0.2326384	0.1581154	1.006635
NR 36	0.4069553	0.9467409E-01	0.1895504	0.2284331	0.2337641	0.1588285	1.006491
NR 37	0.4099110	0.9547213E-01	0.1898954	0.2282770	0.2325176	0.1588444	1.006976
NR 38	0.4055534	0.9561185E-01	0.1881492	0.2290631	0.2323404	0.1587578	1.007365
NR 39	0.4074978	0.9595678E-01	0.1874962	0.2286661	0.2339834	0.1580131	1.007740
NR 40	0.4079242	0.9679742E-01	0.1876362	0.2265183	0.2298162	0.1574728	1.006894

LISTING OF ARRAY COST OF PRODUCING

	NI METALS	NI-ENERGY	NI-HACH BUDG	NI-CHEMICALS	NI-WOOD FRUP	NI-UNSTR HAI	NI-FUE SUB-TOTAL
NR 1	9.651214	7.940739	9.286329	9.588198	10.23690	9.481953	9.366625
NR 2	3.615649	3.133720	3.561100	3.636373	3.776793	3.592807	3.394000
NR 3	2.523511	2.195408	2.462114	2.523395	2.641287	2.500737	2.483175
NR 4	2.046125	1.919481	2.017373	2.045358	2.088924	2.034352	2.027001
NR 5	1.726201	1.669119	1.712414	1.725450	1.743758	1.719580	1.711392
NR 6	1.515334	1.476060	1.506821	1.515528	1.527647	1.511678	1.509466
NR 7	1.364295	1.348763	1.360479	1.364663	1.378954	1.363536	1.361751
NR 8	1.276092	1.265958	1.273224	1.276224	1.278900	1.274760	1.274041
NR 9	1.185170	1.179354	1.183663	1.185354	1.186773	1.184483	1.184121
NR 10	1.135171	1.127359	1.133515	1.135202	1.137750	1.134472	1.133909
NR 11	1.087543	1.083542	1.086708	1.087590	1.088832	1.087181	1.086927
NR 12	1.055606	1.052358	1.055024	1.055705	1.056819	1.055414	1.055197
NR 13	1.020224	1.020816	1.020394	1.020317	1.019961	1.020293	1.020357
NR 14	1.002152	1.002928	1.002357	1.002213	1.001864	1.001242	1.001303
NR 15	0.985082	0.988269	0.9857309	0.985173	0.9838410	0.9852982	0.9855398
NR 16	0.9752807	0.9776200	0.9757688	0.9753123	0.9744192	0.9754589	0.9756208
NR 17	0.9627648	0.9658407	0.9634331	0.9628263	0.9616656	0.9630339	0.9633500
NR 18	0.9570665	0.9606232	0.9577735	0.9570856	0.9557629	0.9573206	0.9572479
NR 19	0.9508053	0.9549847	0.9516161	0.9508385	0.9492739	0.9511103	0.9513808
NR 20	0.9476540	0.9521757	0.9485255	0.9476813	0.9459835	0.9497669	0.9485574
NR 21	0.9438620	0.9474487	0.9445742	0.9438395	0.9425116	0.9441148	0.9443352
NR 22	0.9499431	0.9449766	0.9409082	0.9399363	0.9380994	0.9402768	0.9403851
NR 23	0.9368249	0.9415072	0.9377493	0.9368341	0.9351350	0.9371614	0.9374633
NR 24	0.9445159	0.9408549	0.9353881	0.9345083	0.9329815	0.9340420	0.9351196
NR 25	0.9336553	0.9383773	0.9346043	0.9336668	0.9319673	0.9340116	0.9343210
NR 26	0.9326743	0.9375695	0.9336518	0.9327070	0.9309230	0.9330515	0.9333795
NR 27	0.9302337	0.9361551	0.9328672	0.9320390	0.9305807	0.9333581	0.9335430
NR 28	0.9318797	0.9356886	0.9326427	0.9310754	0.9305579	0.9321811	0.9324242
NR 29	0.9327369	0.9369265	0.9333424	0.9327270	0.9312418	0.9330375	0.9332938
NR 30	0.9313300	0.9333108	0.9322961	0.9313335	0.9295194	0.9316810	0.9319944
NR 31	0.9306431	0.9355548	0.9316074	0.9306444	0.9280720	0.9305979	0.9310267
NR 32	0.9315204	0.9356956	0.9313572	0.9313137	0.9300496	0.9310374	0.9320992
NR 33	0.9351885	0.9367135	0.9330570	0.9321515	0.9305332	0.93254919	0.9325266
NR 34	0.9347063	0.9359459	0.9325602	0.9314797	0.9299091	0.9318150	0.9320833
NR 35	0.9310265	0.9350517	0.9319272	0.9310316	0.9297894	0.9333796	0.9316872
NR 36	0.930167	0.9341644	0.9310043	0.9302041	0.9280466	0.9305748	0.9307214
NR 37	0.9305384	0.9340749	0.9314100	0.9307567	0.9290436	0.9300884	0.9311602
NR 38	0.9308385	0.9334351	0.9299856	0.9298907	0.9274357	0.9297418	0.9296248
NR 39	0.9307864	0.9347654	0.9314712	0.9307653	0.9291237	0.9309047	0.9311651
NR 40	0.9307102	0.9348909	0.9310335	0.9296827	0.9280277	0.9300150	0.9307697

NR 1	10.60206	9.419913	10.13341	9.734064	10.141809	9.734064	9.734064
------	----------	----------	----------	----------	-----------	----------	----------

[illegible][illegible]

LINE	CONSTRUCTION	AGRICULTURE	TRANSPORTATION	TRADE/SECTOR	MANUFACTURING	ESTATE
1	3.794313	11.24651	1.082925	3.961957	1.008751	3.253729
2	0.425611	3.167856	0.1110564	0.9555909	0.31378941-01	0.7579597
3	0.001034	1.691169	-0.14051435-01	0.2705450	0.1325334F-01	0.2394357
4	0.022146	1.023509	0.1322408E-01	0.1008547	0.2744950E-02	0.1762637
5	0.2413605E	0.5003109	0.7019105E-02	0.2163974F-01	0.1209781E-01	0.8007453E-01
6	0.2046733E	0.3152994	0.2149100E-01	0.1757214E-03	0.1014833E-01	0.7202622E-02
7	0.3427030E	1.100925	0.2623809F-01	0.8558166E-03	0.1833307E-02	0.5054800E-02
8	0.1407830F	0.5130761	-0.5885212E-02	0.8488932E-02	0.333211E-02	0.3726664E-02
9	0.2306711E-01	0.6257909E	0.1363896E-01	0.1867654E-02	0.47464431E-02	0.28243479E-02
10	0.1237655E	0.1330450E	0.1467719E-01	0.677119E-02	0.3428495E-02	0.1645137E-02
11	0.1434919E	0.1623140E-01	0.5211405E-02	0.901517E-03	0.4384029E-03	0.5201465E-03
12	0.1024015E	0.1123355E-01	-0.5681127E-03	0.1450812E-02	0.9715303E-03	0.1099788E-02
13	0.1937164E	0.1570525E-01	0.8607437E-02	0.3572257E-02	0.9165349E-02	0.1805985E-02
14	0.1396930E	0.8565052E-02	0.5043254E-02	-0.1625276E-02	0.51198392E-02	0.6979569E-03
15	0.1256381E	0.5019412E-02	0.6510915E-02	0.5445026E-03	0.3990309E-02	0.92336017E-03
16	0.7055530E	0.1799711E-02	0.3541615E-02	0.1669406E-02	0.1898134E-02	0.2317944E-03
17	0.1205300E	0.2479250E-02	0.6612613E-02	0.5509941F-03	0.4505577E-02	0.3005228E-02
18	0.1049701E	0.6220302E-02	0.4021731E-02	0.4623739E-02	0.2310425E-02	0.1596379E-02
19	0.1027145E	0.5269448E-02	0.2203729E-02	0.2090310E-04	0.2672525E-02	0.1767291E-02
20	0.8277923E	0.6196223E-02	0.1422411E-02	0.9015408E-03	0.3255950E-02	0.3055394E-03
21	0.1408475E	0.6546055E-02	0.1972314E-02	0.4389501E-03	0.3195121E-02	0.2639171E-02
22	0.5770160E	0.5191692E-02	0.2486869E-03	0.5312546E-03	0.3970033E-02	0.1463877E-02
23	0.3642143E	0.3799550E-02	0.1676584E-02	0.276911E-02	0.2579903E-02	0.2093051E-02
24	0.3115908E	0.3308343E-02	0.3089900E-02	-0.2392087E-02	0.2624291E-02	0.3066071E-02
25	0.2405907E	0.3260885E-02	0.4408524E-02	0.8082035E-03	0.2496518E-02	0.27217970E-02
26	0.2127760E	0.2108434E-02	0.4567149E-02	0.3201634E-03	0.2530875E-02	0.1592443E-02
27	0.5694973E	0.7917401E-03	0.3950504E-02	0.1653426E-03	0.2650193E-02	0.741974E-02
28	0.4457912E	0.1914013E-03	0.2622742E-02	0.1631594E-03	0.2655683E-02	0.2303350E-02
29	0.3070039E	0.1911911E-03	0.2351024E-03	0.2431171E-03	0.1045444E-02	0.3030430E-03
30	0.5621235E	0.1624711E-02	0.277824E-03	0.7947144E-03	0.3017168E-02	0.4070620E-03
31	0.2023691E	0.2340011E-02	0.0465477E-03	0.1080215E-02	0.3377861E-02	0.4476406E-03
32	0.3940098E	0.3605039E-03	0.4845305E-03	0.7247739E-03	0.2692263E-02	0.1664158E-02
33	0.2044731E	0.2627252E-02	0.9311909E-03	0.1344508E-03	0.2013737E-02	0.7195597E-03
34	0.1197573E	0.231713E-02	0.765470E-03	0.8178311E-03	0.2469560E-02	0.1042407E-02
35	0.122454E	0.1279221E-02	0.1611026E-02	0.3844870E-03	0.1766963E-02	0.1139760E-02
36	0.0082440E	0.6900161E-04	0.2377766E-02	0.1526262E-02	0.1893714E-02	0.2078236E-02
37	0.4304670E	0.3525656E-02	0.3525656E-02	0.1481986E-04	0.2684274E-03	0.7985518E-03
38	0.2127479E	0.1034470E-02	0.1629404E-02	0.9081506E-02	0.2964676E-03	0.1564518E-02
39	0.3059069E	0.1520555E-02	0.2903305E-02	0.1475016E-02	0.2190714E-02	0.2052423E-04
40	0.8349170E	0.1328356E-02	0.1500935E-02	0.1016132E-03	0.5236053E-03	0.9016030E-03

[illegible]

RECEIVED FOR THIS RUN - 5310.271 SECONDS

APPENDIX B

DATA AGGREGATOR--AGGRAT

The user will usually want to aggregate the economic data into sectors of interest. Perhaps he would like to keep the energy sectors separate, but combine many manufacturing sectors into one sector. This can most easily be accomplished through the use of the DSA data aggregator, AGGRAT, which also performs the task of calculating or asking for missing data such as depreciation rates, discount rates, etc., and putting the final data into model format.

B.1 DSA FORMAT

A primary data file must exist for AGGRAT to use. This file consists of a series of records, where each record contains four entries labeled I, J, K, and DATA. These four entries constitute the raw economic data in DSA format, as shown in Fig. B-1. The index K refers to "pages" 1-5. Indices I and J refer usually to industry number, and are the column and row indices for each page. The file must be sorted by K, J, and I.

B.1.1 PAGE 1 - PRODUCTION

The first page of the file is the production transactions matrix. If there are N production industries, this page consists of N columns and N + 6 rows. Index I refers to the production industry. Index J refers to the commodity used by industry I. DATA is the amount of commodity J used by industry I in the production of its product.

The indices I and J need not be integral. For example, the Bureau of Economic Analysis classifies the United States economy into 468 sectors, but only 86 primary industries. For this data base there are 468 values of I ranging for 1.0 to 86.9999 and 474 values of J.

Six values of J (0 to -5) are added to this page. These rows are defined as:

J = 0	K = 1 PRODUCTION	K = 2 INVESTMENT	K = 3 CONSUMPTION	K = 4 TRADE	K = 5 DOMESTIC TRANSFERS
J = -1	UNDIFFERENTIATED	FOR INDUSTRY I FROM INDUSTRY J	1 PERSONAL	1 EXPORTS	FROM INDUSTRY J
J = -2	WAGES		2 MILITARY	2 IMPORTS	TO INDUSTRY I
J = -3	BURDEN		3 FEDERAL GOVERNMENT		
J = -4	RENT (PTI)		4 LOCAL GOVERNMENT		
J = -5	CAPITAL INVENTORY		5 INVENTORY CHANGE		
	DEPRECIATION		6 UNDIFFERENTIATED		

Figure B-1. Description of DSA Data Format

<u>J</u>	<u>Interpretation</u>
0	Undifferentiated Imports
-1	Wages to Labor
-2	Burden (Taxes)
-3	Rental on Capital (Property-Type Income)
-4	Capital Inventory
-5	Depreciation (Total)

The term "undifferentiated imports" refers to imports which have no domestic counterpart or which are consumed by final users in the form in which they are imported.

The sum of each column of page 1 down through $J = -3$ is the total output of that industry. It must equal the total use for that product, i.e., it must be equal to the sum of the corresponding row across pages 1-4. The data for $J = -4$ and $J = -5$ are not required. They are used only if the user wants the final aggregated data put into model format and even then are not required (see Sec. B.2).

B.1.2 PAGE 2 - INVESTMENT

The second page of the file is the investment, or capital formation page. The N values of I refer to the industry which uses the capital. The index J refers to the commodity required in the manufacture of the capital. No labor or capital is required in the manufacture of the capital since this matrix merely describes the mix of items comprising the capital for each industry. One extra row, $J = 0$ for undifferentiated imports, is provided.

B.1.3 PAGE 3 - CONSUMPTION

Consumption by final users is represented by page 3. There are six final users to which the data may be applied. The user may define these six in any way he wishes, but the aggregator normally assumes the following categories:

<u>I</u>	<u>Final User</u>
1	General Population
2	Military
3	Federal Government
4	Local Government
5	Inventory Change
6	Undifferentiated

If the user wishes to add more consuming sectors he will have to modify the model. In general, however, these six have been sufficient for all data bases used to date. The rows of page 3 are once again the industry codes plus $J = 0$ for undifferentiated imports.

B.1.4 PAGE 4 - TRADE

The trade matrix contains just two columns. $I = 1$ refers to exports while $I = 2$ refers to imports. The DATA value for $J = 0$, $I = 2$ must in this case equal the sum of the undifferentiated imports row, i.e., all of the undifferentiated imports are "imported" at $I = 2$, $J = 0$, $K = 4$.

B.1.5 PAGE 5 - DOMESTIC TRANSFERS

Some data bases have a quantity called domestic transfers. When a particular industry produces a completely secondary product, the secondary product is "sold" to the primary industry for that product, which distributes it. This fictitious transaction is called a domestic transfer. After the data is aggregated, the diagonal elements of this matrix must be subtracted from the diagonal elements of the production matrix, page 1, since these elements represent fictitious transactions between an aggregated industry and itself.

B.2 USING THE AGGREGATOR

The aggregator is primarily self-explanatory, i.e., the user need only answer the questions asked by the aggregator. The first question asked is:

HAS DATA BEEN AGGREGATED SUFFICIENTLY?

The aggregator has two uses: (1) to aggregate economic sectors, and (2) to put the aggregated data into model format. If the data were previously aggregated and put onto a file and the user just wants to use it in the model, or to change it slightly, he may answer yes. If the data are to be further aggregated, he should answer no.

The aggregator next asks the user for the name of the file which contains the data. If the data has been sufficiently aggregated, it is read in and the program informs the user how many industries are contained in the data. If it has not, the program needs more information. In particular, the user will be asked to supply the following:

1. The number of industries which will comprise the aggregated data base.
2. The industry codes which go into each aggregated sector.

No industry code can be in more than one sector. If this happens the aggregator will inform the user. The codes are input as a series of code ranges. An illustrative sequence is:

Example 1:

ECONOMIC MODEL DATA PREPARATION--HAS DATA BEEN AGGREGATED
SUFFICIENTLY? (Y OR N)

yes

INPUT FILE NAME

filename

BEGINNING TO READ IN DATA:

FINISHED PAGE 1

FINISHED PAGE 2

FINISHED PAGE 3

FINISHED PAGE 4

FINISHED PAGE 5

DATA CONTAINS (number) PRODUCTION INDUSTRIES

Example 2:

ECONOMIC MODEL DATA PREPARATION--HAS DATA BEEN AGGREGATED
SUFFICIENTLY? (Y OR N)
no
INPUT FILE NAME
filename
INPUT NUMBER OF RESOURCES (MAX OF 100)
10
DO YOU WISH DATA TO BE INTEGRALLY AGGREGATED (Y OR N)
"Y (ES)" WILL PUT DATA INTO INTEGRAL CODES 1-10 by INTEGERIZING
ALL INDUSTRY CODES¹
no
INPUT BEGINNING AND ENDING CODES FOR INDUSTRY 1
1 2.999
ANY OTHER SEGMENTS FOR INDUSTRY 1? (Y OR N)
yes
INPUT BEGINNING AND ENDING CODES FOR INDUSTRY 1
3 3.9999
ANY OTHER SEGMENTS FOR INDUSTRY 1? (Y OR N)
no
INPUT BEGINNING AND ENDING CODES FOR INDUSTRY 2
4.0 5.4001
ANY OTHER SEGMENTS FOR INDUSTRY 2? (Y OR N)
no
. . .
. . .
ANY OTHER SEGMENTS FOR INDUSTRY 10? (Y OR N)
no

¹This is useful for data bases containing a large number of sectors divided into primary industries, such as the BEA US data base. Instead of the user having to input code ranges 1.0 - 1.9999, 2.0 - 2.999, etc., the model automatically sets the code ranges. This will aggregate the data into the primary industries which can be further aggregated later.

BEGINNING AGGREGATION:

FINISHED PAGE 1

FINISHED PAGE 2

FINISHED PAGE 3

FINISHED PAGE 4

FINISHED PAGE 5

After the aggregation, the user has the option of listing any of the aggregated data by page number, and of changing any data element. After the user is satisfied with the data, the program tests for consistency between row and column sums. If inconsistencies are found, the program informs the user of the problem industries and allows the user to look at and change the data before testing again. If the data is satisfactory, the user may copy it to a separate file. This aggregated data will have integral industry codes as opposed to the primary data base which need not have integral codes.

B.3 MODEL FORMAT

After aggregation, the user may decide to have the data put into model format. Assuming he answers the question

WOULD YOU LIKE DATA PUT INTO MODEL FORMAT? (Y OR N)

with "yes," the aggregator will inquire as to his preferences regarding the number of capital types, the names of consumables, the number and definition of consumption activities, and primary data arrays.

B.3.1 CAPITAL TYPES

The following question is asked:

CAPITAL MAY BE DIVIDED INTO TWO TYPES BY INDUSTRY GROUP IN ORDER TO DISTINGUISH TWO DEPRECIATION RATES (E.G., FOR STRUCTURES VS. EQUIPMENT)
--DO YOU WISH THIS DIVISION? (Y OR N).

If the user answers with "yes," he will be asked to supply the aggregated industries contained in each type of capital. Some industries may use only one type of capital. The aggregator will determine if this

is so and delete any capital not required. As only two types of capital are allowed, the total number of capital items will be between $N + 1$ and $2N + 1$ where N is the number of aggregated industries. Every industry must use at least one capital type. If an industry is found which does not use any capital, the model will inform the user and abort.

B.3.2 INDUSTRY NAMES

The user has the option of letting the aggregator provide names for the aggregated industries (IND 1, IND 2, etc.) or naming them himself. The industry name may not be greater than 15 characters.

B.3.3 CONSUMPTION ACTIVITIES

The user may break page 3 into a set of consumption activities in any way desired. The most general way to define a consumption activity is by giving the industries and consumer groups (see Sec. B.1.3) included in the activity:

```
INPUT GROUP NUMBER WHICH IS INCLUDED IN CONSUMPTION ACTIVITY
(number); "S" TO STOP
1
INPUT INDUSTRY NUMBER OR RANGE OF INDUSTRY NUMBERS INCLUDED IN
GROUP 1'S PARTICIPATION IN ACTIVITY (number)
5
ANY MORE INDUSTRIES FOR GROUP 1 IN CONSUMPTION ACTIVITY (number)?
no
ANY MORE GROUPS IN CONSUMPTION ACTIVITY (number)?
no
INPUT NAME FOR THIS ACTIVITY
```

Contiguous groups may be listed together. For example, if the above activity were to be total consumption of industry 5, the user could have specified groups 1-6 in the first question by typing in "1 6" or "1,6." It is not necessary to list each group separately.

Conversely, if the user would like to break out a single group or combination of groups for each of a list of industries, he may answer the second question as follows:

INPUT INDUSTRY NUMBER OR RANGE OF INDUSTRY NUMBERS INCLUDED IN
GROUP 1'S PARTICIPATION IN ACTIVITY (number)

5 10 I

The "I" informs the model that industries 5 through 10 constitute six independent consumption activities by the same group or groups.

B.3.4 MODEL PARAMETERS

Besides the activity levels and direct requirements matrix, the economic model requires the following data:

- (1) Capital inventory levels
- (2) Capital depreciation rates
- (3) Current level of inventory growth
- (4) Current value trends (used in calculating the discount rates--not used directly)
- (5) Discount rates
- (6) Gestation times
- (7) Population growth

The aggregator considers which of the above it can calculate and which it needs to ask for. For example, if rows -4 and -5 of page 1 contain data, then the total capital inventory and overall depreciation rate is known. If the industry contains only one type of capital, its inventory and depreciation is therefore also known. If two types of capital are used, the model will split the total inventory according to how the user wants to split the depreciation rates of the two types.

The aggregator will list the above data for the user, inserting default values where necessary (e.g., gestation times are defaulted to one year). It will then ask the user to change any of the data he wishes with the exception of capital inventory and activity levels. Some of the data, e.g., discount rates or inventory when two types of capital are used or when row -4 is blank, are listed as question marks. These data will be calculated by the aggregator after the user has specified the remaining parameters. Value trends are defaulted to the growth rate. If the growth rate is to be calculated, i.e., it is listed

as a question mark, the value trend will be listed as zero. The actual value used in the calculation of the discounts will be either the growth rate, which is calculated first, or the value input by the user.

After the user has changed the data as he wishes, the aggregator will calculate the remaining data (previously listed as "?") and will relist the updated data arrays. The data are then output onto file ECO.DAT for later use by the economic model.

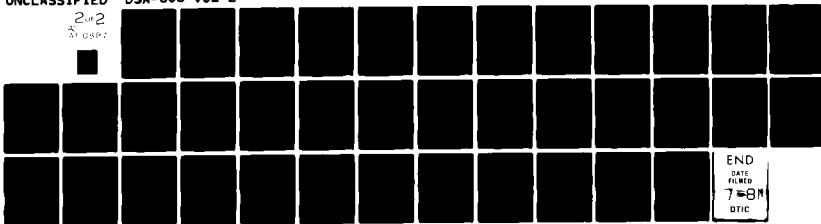
AD-A100 897

DECISION-SCIENCE APPLICATIONS INC ARLINGTON VA F/G 5/3
THE DYNEVAL (DYNAMIC ECONOMIC VALUES) MODEL. VOLUME II. DOCUMENT--ETC(U)
MAY 81 G E PUGH, M. T NUNENKAMP, J C KRUPP AC00C104
DSA-306-VOL-2 NL

UNCLASSIFIED

202

31 0887



END
DATE
FILMED
7-81
DTIC

APPENDIX C

GLOSSARY OF VARIABLES USED IN DSA ECONOMIC RECOVERY MODEL

ACON = Direct requirements matrix -- ACON(I,J) is the amount of commodity I used per unit of activity J
 ACTLAB = Labels (names) for activities
 ARENT = Amount spent on capital and labor per unit of activity for the production of consumables in the current economy (first NCAPM1 is amount spent on capital (each capital is used for only one activity specified in ITRADE); next NCON is amount spent on labor)
 B = Exponent for utility and translog functions (first NCAPM1 for translog exponent of capital/capital trade-offs; next NCON for translog exponent of capital/labor trade-offs, next NCONAC for utility exponent)
 CAPLAB = Labels (names) for capital resources
 CONLAB = Labels (Names) for consumables
 D = Depreciation for the NCAPM1 capital resources (D(NCAP) = population growth)
 DISCNT = Discount rate for the NCAPM1 capital resources
 DLAMB = Rental value for capital resources (each time period)
 DTAU = Difference between TPER and TLAG (DTAU(J) = $TPER * (1.0 - AMOD(TLAG(J)/TPER, 1.0))$)
 EFFCAP = Effective capital inventory (used when two types of capital are used to make one consumable)
 EFFCST = Effective cost of producing capital
 EFFLAB = Labor used in production of particular consumable (not constant but varies with time period)
 EPSO = Epsilon parameters used in Everett algorithm for phase two
 FLAM = Shadow value (offset by one time period) (FLAM(K=1)) contains horizon DLAMB values for capital

FUDGE	=	Normalization coefficient for translog function (first NCAPM1 for capital/capital trade-offs; next NCON for capital/labor trade-offs)
GAMMA	=	Coefficient for utility and translog functions (first NCAPM1 for translog function of capital/ capital trade-offs; next NCON for translog function of capital/labor trade-offs (if GAMMA .GT. 0 IT IS GK -- if .LT. 0 ABS(GAMMA) is GL, GK + GL = 1.0); next NCONAC are utility coefficients (activity = GAMMA/(cost**BO)))
GROWTH	=	Fractional growth of industries producing capital in current economy
ITRADE	=	Array stating for which industry capital J is used
K	=	General index for time period
NACT	=	Number of activities
NCAP	=	Number of capital resources (including labor) (NCAP must be .GT. NCON .AND. .LE. 2*NCON + 1)
NCAPM1	=	Number of capital resources (excluding labor) = NCAP - 1
NCON	=	Number of consumable resources
NCONAC	=	Number of consumption activities
NCST	=	Total number of resources = NCAP + NCON
NPER	=	Number of time periods
NPRDAC	=	Number of production activities = NCON + NCAP -1
PERLAB	=	Labels used when printing out arrays with time period indices
PRMLAM	=	Lambda prime - gross profit in the manufacture of commodity I which is distributed between capital & labor
R	=	Resource level for capital
REQ	=	Equilibrium levels for capital

RFUT	=	Resource level one gestation time in the future if no investment activity occurs (used in phase one only)
RLAST	=	Resource level achieved in previous pass for given time period and capital item
RORGNL	=	Starting level of resources
RSCALE	=	Scaling factor used in phase one method of determining investment activity levels
RTRGT	=	Target resource level for one gestation time in the future (used in phase one only)
SCRATCH	=	Scratch pad array -- used for a variety of functions
SUMLAB	=	Labels used in summary tables at end of run
SUMMARY	=	Summary tables array
T	=	Time at center of time period K
TCONS	=	Total consumption of commodity I
TLAG	=	Gestation time for capital (years)
TPER	=	Length of time period in years
W	=	Storage array which contains every other array; W is written onto insurance file after each pass for restarts
X	=	Activity level
Z	=	Minimum consumption activity level

APPENDIX D

SOURCE CODE LISTING


```

01300 IF(NPER .GE. 2) GO TO 50
01310 WRITE(UNIT,5)
01320 NPER = 2
01330 GO TO 50
01340 40 CONTINUE
01350 READ(UNIT) NPER
01360 50 CONTINUE
01370 NCST = NCAP + NCUN
01380 NPKDAC = NCST - 1
01390 NCUNAC = NACT - NPKDAC - 1
01400 NCAPH1 = NCAP - 1
01410 NCUNP1 = NCUN + 1
01420 NPERP1 = NPER + 1
01430 NPERP2 = NPERP1 + NPER
01440 IF((PLOT.II.O) GO TO 55
01450 NRECSZ = 5 + (NCAP*NPERP1) + (NCST*NPERP1)
01460 * + (NCAPH1*NPER) + (NCAP*NPER)
01470 * + (NPER*NACT) + (3*(NCST+NACT+1))
01480 NRECNO = 1
01490 OPEN(UNIT=5,DEVICE='DISK',ACCESS='RANDOM',MODE='BINARY',
01500 * FILE='PLOT.DAT',RECORD SIZE=NRECSZ)
01510 C..... ASSIGN STORAGE
01520 C
01530 C
01540 55 CONTINUE
01550 IFLAM = 1
01560 IT = IFLAM + NCST*NPERP1
01570 ID = IT + NPERP2
01580 IFLAG = ID + NCAP
01590 IDLMB = IFLAG + NCAPH1
01600 ITRAD = IDLMB + NCAP*NPER
01610 IFUUG = ITRAD + NCAPH1
01620 ICAPEF = IFUUG + NPKDAC
01630 ILABEF = ICAPEF + NCON
01640 ISCK = ILABEF + NCON
01650 NSCK = MAXO(NCST,IB)
01660 IRTGT = ISCK + NSCK
01670 IRFUT = IRTGT + NCAPH1
01680 ICON = IRFUT + NCAPH1
01690 IAKN = ICON + NCON
01700 IACON = IAKN + NFRDAC
01710 IZ = IACON + NCUNP1*NACT
01720 IGAM = IZ + NCONAC
01730 IB = IGAM + NACT
01740 IICAP = IB + NACT
01750 ILCUN = IICAP + 3*NCAP
01760 IIACT = ILCUN + 3*NCUNP1
01770 ILPER = IIACT + 3*NACT
01780 IIRKIG = ILPER + 3*NPER
01790 IRLST = IIRKIG + NCAP
01800 IASCU = IRLST + NCAP*NPERP2
01810 IKSUM = IASCU + NCAPH1
01820 IDISC = IKSUM + NCAPH1
01830 INEO = IDISC + NCAPH1
01840 IEFILM = INEO + NCAP
01850 IEF50 = IEFILM + NCON
01860 IDTAU = IEF50 + NPER*NCAFM1
01870 IEFCSZ = IDTAU + NCAPH1

```



```

0.2910 C
0.2970 F0 = SCRICH(NCAP)
0.3000 DO 110 J = 1, NCAPI
0.3010 KURBN(J) = R(J,1) * SCRICH(J)/F0
0.3020 110 CONTINUE
0.3030 KURBN(NCAP) = R(NCAP,1)
0.3040 C
0.3050 C..... CALCULATE CURRENT ECONOMY'S UTILITY AND PRODUCTION PARAMETERS
0.3060 C
0.3070 CALL CURRNT(NCON,NCAP,NCAPM1,NFNUAL,NCUNAC,NACT,
0.3080 1 X,Z,K,SCRICH,SOLNOM,GAMMA(NCST),B(NCST),
0.3090 2 GAMMA,B,ITRADE,B,AMB,ITRADE,FUDGE,ARENT,CAPLAB,ACTLAB)
0.3100 C
0.3110 C..... CALCULATE EQUILIBRIUM STATE
0.3120 C
0.3130 WRITE(LP,3) F0,B(NCAP)
0.3140 CALL XMIT(NCST,1,0,FLAM)
0.3150 CALL XMIT(NCAP,K,REG)
0.3160 CALL EQUIM(NCON,NCUNF1,NCAP,NCST,NCAPM1,NFNUAL,NCUNAC,NACT,REG,
0.3170 1 X,Z,B,AMB,FLAM,EFFFLAM,FUDGE,EFFCAP,EFFLAB,GAMMA(NCST),B(NCST),
0.3180 2 GAMMA,B,ITRADE,B,ACON,DISCNT,BTAU,FLAG,TPER,SCRICH,
0.3190 3 CAPLAB,CONLAB,ACTLAB,ARENT(NCAP),TCONS,FRLAM)
0.3200 SOL = SOLF(NCONAC,X(NCST,1),Z,GAMMA(NCST),B(NCST))
0.3210 SOL = SOL / SOLNOM
0.3220 WRITE(LP,5) SOL
0.3230 CALL SUMMARY(NCST,NCAPM1,NACT,X,REG,SUMMARY)
0.3240 WRITE(LP,10) (SUMRY(I,1),I=1,5)
0.3250 C
0.3260 C..... STARTING SITUATION
0.3270 C
0.3280 WRITE(LP,7)
0.3290 CALL FRARI(ROKGNL,NCAP,15H STARTING RES,CAPLAB,LP,132)
0.3300 C
0.3310 C..... INITIALIZATION
0.3320 C
0.3330 CALL EFFNAP(EFFCAP,FUDGE,GAMMA,B,ITRADE,ROKGNL,NCON,NCAPM1)
0.3340 CALL EFFNAP(SCRICH,FUDGE,GAMMA,B,ITRADE,REG,NCUN,NCAPM1)
0.3350 DO 120 I = 1,NCON
0.3360 II = I + NCAPM1
0.3370 X(II,1) = X(II,1) * EFFCAP(I)/SCRICH(I)
0.3380 120 CONTINUE
0.3390 DO 130 N = 2,NPERF2
0.3400 CALL XMIT(NCAP,REG,RLAST(1,N))
0.3410 IF(N.GT,NPERF1) GO TO 130
0.3420 CALL XMIT(NCST,FLAM(1,1),FLAM(1,N))
0.3430 130 CONTINUE
0.3440 CALL XMIT(NCAP,D,AMB(1,1),FLAM(1,1))
0.3450 CALL XMIT(NCAP,REG,RLAST(1,1))
0.3460 CALL XMIT(NCAPM1,X,RSSCALE)
0.3470 C
0.3480 C
0.3490 C
0.3500 C
0.3510 C
0.3520 C

```

NOTE: FLAM(J,1) IS USED ABOVE TO STORE THE HORIZON RENTAL
VALUE FOR CAPITAL ITEMS. SINCE THE MODEL USES FLAM(J,N+1)
IN ITS CALCULATION FOR TIME PERIOD N, FLAM(J,1) IS AVAILABLE
FOR THIS.

```

0.3530      CLOSE = 1
0.3540      EVNRT = 1.0E-5
0.3550      FAKTIO = 0.005
0.3560      WRITE (1F,2)
0.3570      CLOSE (UNIT=1F,DEVICE='DISK',FILE='PRINT.DAT')
0.3580      150 CONTINUE
0.3590      ICONS = 150
0.3600      MFLTA = 1.0
0.3610      FUM3 = 1.0
0.3620      SLPMLT = SLOPLO
0.3630      MAXITER = MAXITIO
0.3640      REFMIX = 5.0
0.3650      MPASS = MINO(20,MPASS/2)
0.3660      IF (RESTART) READ (IDAT) IFER, SIN NOM, GANSUM, ENKRTOL, EVERKEI, IPASS
0.3670      IF (RESTART) CLOSE (UNIT=INS, DEVICE='DISK', FILE='INS.DAT')
0.3680      IF (IPLOT.EQ.0 .AND. .NOT.REWIND) NRECNO = 1PASS
0.3690      C
0.3700      C..... MAIN DRIVER --- PASS LOOP
0.3710      C
0.3720      200 CONTINUE
0.3730      CALL STOP11
0.3740      OPEN (UNIT=1F, DEVICE='DISK', ACCESS='APPEND',
0.3750      1  MORE='ASCII', FILE='PRINT.DAT')
0.3760      CALL UTILIZ(CPU1)
0.3770      CALL TIMFR(NCON, NCAF, NCAF1, NEST, NFRKAC, NCONAC, NACT, NFER, NCONF1,
0.3780      2  NFRF1, NFER2, R, KLAST, GRWTH, D, I, TLAG, DTAU, ILAMB, FLAG, RTROT,
0.3790      3  SCRTCH, EFFCST, EFFLAM, TCONS, REQ, PKMLAN, KSCALE)
0.3800      CALL UTILIZ(CPU2)
0.3810      CPU1 = CPU2 - CPU1
0.3820      CPU2 = 1.0
0.3830      FUM = 1.0
0.3840      IF (IPASS .EQ. 1) GO TO 230
0.3850      SUM = 0.0
0.3860      SUM1 = 0.0
0.3870      DO 220 K = 1, NFER
0.3880      210 SUM = SUM + (KLAST(J,N) - R(J,N))**2 / (0.5*(KLAST(J,N) + R(J,N)))
0.3890      SUM1 = SUM1 + 0.5*(KLAST(J,N) + R(J,N))
0.3900      220 CONTINUE
0.3910      FUM = SORT(SUM/SUM1)
0.3920      GO TO 270
0.3930      230 CONTINUE
0.3940      DO 250 J = 1, NCAF1
0.3950      IF (R(J,NFER) .LT. 0.95*REQ(J)) GO TO 260
0.3960      250 CONTINUE
0.3970      SLPMLT = SLOPLO
0.3980      DO 240 N = 1, NFER
0.3990      CALL XRTI(NCAF1, X(1,N), EPSO(1,N))
0.4000      240 CONTINUE
0.4010      GO TO 270
0.4020      260 CONTINUE
0.4030      SLPMLT = SLPMLT / 2.0
0.4040      WRITE (11F,3)
0.4050      WRITE (1F,3)
0.4060      GO TO 200
0.4070      270 CONTINUE
0.4080      IF (.NOT.EVERKEI) GO TO 310
0.4090      SUM = 0.0
0.4100      SUM1 = 0.0
0.4110      DO 305 K = 1, NFER
0.4120      305 SUM = SUM + (KLAST(J,N) - R(J,N))**2 / (0.5*(KLAST(J,N) + R(J,N)))
0.4130      SUM1 = SUM1 + 0.5*(KLAST(J,N) + R(J,N))
0.4140      310
0.4150

```

```

04140      DO 300 J = 1,NCAFMI
04150      SUM = SUM + EFCS(J,N)*X(J,N)*EF50(J,N)**2
04160      SUM1 = SUM1 + EFCS1(J,N)*X(J,N)
04170      CONTINUE
04180      SUM = SUM1
04190      EN100 = AMIN1(EN100,1.0E-20)
04200      EN100 = AMAX1(EN100,0.0001)
04210      FOM3 = SUM1*(SUM/SUM1)
04220      CONTINUE
04230      WRITE(ITY,4) IPASS,CPU,1.00F,FOM,FOM2,FOM3
04240      WRITE(ITY,4) IPASS,CPU,1.00F,FOM,FOM2,FOM3
04250      IFLAG = 11
04260      IF(IPASS.GE.NPASS .OR. FOM2.LE.FOMSTP) IFLAG = -1
04270      C
04280      C
04290      C
04300      C
04310      C
04320      CALL LAMBDA(NPER,NCAF,NCAFMI,NCFST,NCFPT,NCFP2,T,D,FLAN,DLAMB,
04330      * IFLAG,BTAP,DISCONT,EFFECT,R,REQ,IFLAG)
04340      IF(IFLAG.LE.0) GO TO 350
04350      IF(EVENET) GO TO 330
04360      DO 320 N = 1,NPER
04370      N1 = N + 1
04380      EF50(J,N) = 0.3*EF50(J,N) + 0.7*X(J,N)
04390      CONTINUE
04400      315 CONTINUE
04410      IF(FOM.GT.0.005 .AND. IPASS.LT.NPASS) GO TO 345
04420      EVENET = .TRUE.
04430      WRITE(ITY,9)
04440      WRITE(ITY,9)
04450      DO 325 N = 1,NPER
04460      N1 = N + 1
04470      DO 324 J = 1,NCAFMI
04480      X(J,N) = AMAX1(EF50(J,N),0.001*X(J,NPER))
04490      DIF = FLAN(J,N1) - EFCS1(J,N)
04500      EF50(J,N) = SIGN(0.05,DIF)
04510      CONTINUE
04520      324 CONTINUE
04530      CONTINUE
04540      330 CONTINUE
04550      DO 340 N = 1,NPER
04560      N1 = N + 1
04570      DO 335 J = 1,NCAFMI
04580      DIF = FLAN(J,N1) - EFCS1(J,N)
04590      IF(X(J,N).GT.0.0) GO TO 332
04600      IF(DIF.LE.0.0) GO TO 335
04610      X(J,N) = 0.0005*X(J,NPER)
04620      EF50(J,N) = 0.05
04630      CONTINUE
04640      332 CONTINUE
04650      F0 = -DELTA
04660      IF(DIF*EF50(J,N).GT.0.0) F0 = FCONS*DELTA
04670      EF50(J,N) = ABS(EF50(J,N)) * (1.01F0)
04680      EF50(J,N) = AMIN1(EF50(J,N),0.9)
04690      EF50(J,N) = SIGN(EF50(J,N),DIF)
04700      X(J,N) = X(J,N)*(1.01EF50(J,N))
04710      IF(EF50(J,N).GT.-0.8 .OR. X(J,N).GT.1.0E-04*X(J,NPER))
04720      GO TO 335
04730      X(J,N) = 0.0
04740      EF50(J,N) = 0.0

```


106

```

05000 READ(DAT) Z
05010 READ(DAT) ALPH
05020 READ(DAT) ITRNGR
05030 CONTINUE
05040 WRITE(111Y,2)
05050 READ(111Y,4) IPER
05060 FLAG2 = 2.811ALPH
05070 IF(IPER .LT. FLAG2) GO TO 65
05080 WRITE(111Y,3) FLAG2, FLAG2, FLAG2, FLAG2
05090 GO TO 60
05100 CONTINUE
05110 DO 70 N = 1, NPERF2
05120 XN = N - 1
05130 I(XN) = IPER * (XN + 0.5)
05140 CONTINUE
05150 NPERMX = (NPERF2 + 1.5) * ITRNGR / IPER + 0.001
05160 IF(NPERMX .GT. NPERF2) CALL ZABORT(
05170 1 'INCREASE SIZE OF NPERMX IN STORAGE' FOR MAX TEST. TIME!)
05180 DO 75 J = 1, NPERMX
05190 XLAG = ILAG(J) / IPER
05200 JLAG = XLAG + 1.0E-10
05210 YLAG = JLAG
05220 ITAG(J) = (1.0 - XLAG + YLAG) * IPER
05230 CONTINUE
05240 C..... PRINT BASIC ARRAYS
05250 C
05260 80 CONTINUE
05270 WRITE(111Y,15)
05280 READ(111Y,8) Y
05290 IF(Y.EQ.NO) GO TO 82
05300 IF(Y.EQ.YES) GO TO 81
05310 WRITE(111Y,11) BELL
05320 GO TO 80
05330 81 CONTINUE
05340 WRITE(LP,1)
05350 WRITE(LP,4) NCAP, NCON, NPERMAC, NCONMAC, NACT
05360 WRITE(LP,7)
05370 CALL PRN1(Z, NCONMAC, 15MINIMUM ACTIVITY, ACTLAB(1, NCST), LP, 132)
05380 CALL PRN1(8CONMAC, NCONMAC, 15SUB(UTILITY-ELAS), ACTLAB(1, NCST),
05390 1 LP, 132)
05400 CALL PRN1(GAMMAC, NCONMAC, 15H WEIGHT FACTOR,
05410 1 ACTLAB(1, NCST), LP, 132)
05420 CALL PRN1(BCONCAP, NCON, 15BETA F-CONSUMER, ACTLAB(1, NCAP), LP, 132)
05430 CALL PRN1(BCONCAP, NCON, 15BETA CAP-TRADE, CAPLAB, LP, 132)
05440 CALL PRN1(SCRICH, NCAP, 15H FRACTIONAL RES, CAPLAB, LP, 132)
05450 CUSE (UNIT-LF, DEVICE='PSK', FILE='PRINT.DAT', DISPOSE='PRINT')
05460 COPEN(UNIT-LF, DEVICE='PSK', FILE='PSK', ACCESS='SEQUENT',
05470 1 MODIF='ASCII')
05480 WRITE(111Y,6)
05490 GO TO 93
05500 82 CONTINUE
05510 WRITE(111Y,16)
05520 READ(111Y,8) Y
05530 IF(Y.EQ.NO) GO TO 93
05540 IF(Y.EQ.YES) GO TO 84
05550 WRITE(111Y,11) BELL
05560 GO TO 82
05570 83 CONTINUE
05580 WRITE(111Y,9)
05590

```

```

07100 07100 07100
07110 07110 07110
07120 07120 07120
07130 07130 07130
07140 07140 07140
07150 07150 07150
07160 07160 07160
07170 07170 07170
07180 07180 07180
07190 07190 07190
07200 07200 07200
07210 07210 07210
07220 07220 07220
07230 07230 07230
07240 07240 07240
07250 07250 07250
07260 07260 07260
07270 07270 07270
07280 07280 07280
07290 07290 07290
07300 07300 07300
07310 07310 07310
07320 07320 07320
07330 07330 07330
07340 07340 07340
07350 07350 07350
07360 07360 07360
07370 07370 07370
07380 07380 07380
07390 07390 07390
07400 07400 07400
07410 07410 07410
07420 07420 07420
07430 07430 07430
07440 07440 07440
07450 07450 07450
07460 07460 07460
07470 07470 07470
07480 07480 07480
07490 07490 07490
07500 07500 07500
07510 07510 07510
07520 07520 07520
07530 07530 07530
07540 07540 07540
07550 07550 07550
07560 07560 07560
07570 07570 07570
07580 07580 07580
07590 07590 07590
07600 07600 07600
07610 07610 07610
07620 07620 07620
07630 07630 07630
07640 07640 07640
07650 07650 07650

03 CONTINUE
WRITE(ITYY,19)
WRITE(ITYY,13)
CALL CURAND(4,NITY,CUM,ITYY)
DO 85 I = 1,8
  IF(CUM(I).EQ. XCOM(1)) GO TO 86
85 CONTINUE
WRITE(ITYY,11) BELL
GO TO 84
86 CONTINUE
IF(I.EQ. 1) GO TO 83
IF(I.EQ. 2) CALL FRAR(B(NCST),NCONAC,15H UTILITY ELASTIC,
1 ACILAB(1,NCST),ITY,80)
IF(I.EQ. 3) CALL FRAR(B(NCAP),NCON,15H BETA CAPL/LABOR,
1 ACILAB(1,NCAP),ITY,80)
IF(I.EQ. 4) CALL FRAR(B(NCAPM),15H BETA CAPL/CAPL,CAPLAB,
1 ITY,80)
IF(I.EQ. 5) CALL FRAR(GAMMA(NCST),NCONAC,15H WEIGHT FACTOR,
1 ACILAB(1,NCST),ITY,80)
IF(I.EQ. 6) CALL FRAR(SCRATCH(NCAP,15H FRACTIONAL RES,
1 CAPLAB,ITY,80)
IF(I.EQ. 7) CALL FRAR(Z,NCONAC,15H MINIMUM ACTIVITY,
1 ACILAB(1,NCST),ITY,80)
IF(I.NE. 8) GO TO 84
93 CONTINUE
WRITE(ITYY,17)
READ(ITYY,8) Y
IF(Y.EQ. NO) GO TO 130
IF(Y.EQ. YES) GO TO 95
WRITE(ITYY,11) BELL
GO TO 93
94 CONTINUE
WRITE(ITYY,9)
WRITE(ITYY,18)
GO TO 96
95 CONTINUE
WRITE(ITYY,18)
WRITE(ITYY,14)
96 CONTINUE
CALL CURAND(4,NITY,CUM,ITYY)
IF(ICOM(4).EQ. 0) ICOM(4) = ICOM(3)
J1 = ICOM(3)
J2 = ICOM(4)
DO 98 I = 1,8
  IF(CUM(I).EQ. XCOM(1)) GO TO 99
98 CONTINUE
WRITE(ITYY,11) BELL
GO TO 95
99 CONTINUE
GOTO 94,100,105,110,115,120,125,80), I
100 CONTINUE
IF(J1.GT.0 .AND. J2.GT.0 .AND. J2.LE.NCONAC) GO TO 101
WRITE(ITYY,11) BELL
IF(J1.LE.0 .OR. J2.GT.NCONAC) WRITE(ITYY,12) NCONAC
GO TO 96
101 CONTINUE

```

```

07600      DO 102 J = J1,J2
07610      N = NEXT(J-1)
07620      BCK = LOG(N)
07630      102 CONTINUE
07640      WRITE(ITY,13)
07650      WRITE(ITY,14)
07660      GO TO 96
07670
07680      103 CONTINUE
07690      IF(J1.GT.0 .AND. J2.LE.NCON) GO TO 106
07700      WRITE(ITY,11) BELL
07710      IF(J1.LE.0 .OR. J2.GT.NCON) WRITE(ITY,12) NCON
07720      GO TO 95
07730
07740      106 CONTINUE
07750      DO 107 J = J1,J2
07760      N = NCAP(J-1)
07770      BCK = CON(2)
07780      107 CONTINUE
07790      WRITE(ITY,13)
07800      WRITE(ITY,14)
07810      GO TO 96
07820
07830      110 CONTINUE
07840      IF(J1.GT.0 .AND. J2.LE.NCAPM1) GO TO 111
07850      WRITE(ITY,11) BELL
07860      IF(J1.LE.0 .OR. J2.GT.NCAPM1) WRITE(ITY,12) NCAPM1
07870      GO TO 95
07880
07890      111 CONTINUE
07900      DO 112 J = J1,J2
07910      BCK = CON(2)
07920      112 CONTINUE
07930      WRITE(ITY,13)
07940      WRITE(ITY,14)
07950      GO TO 96
07960
07970      115 CONTINUE
07980      IF(J1.GT.0 .AND. J2.LE.NCONAC) GO TO 116
07990      WRITE(ITY,11) BELL
08000      IF(J1.LE.0 .OR. J2.GT.NCONAC) WRITE(ITY,12) NCONAC
08010      GO TO 95
08020
08030      116 CONTINUE
08040      DO 117 J = J1,J2
08050      N = NEXT(J-1)
08060      GAMMA(N) = CON(2)
08070      117 CONTINUE
08080      WRITE(ITY,13)
08090      WRITE(ITY,14)
08100      GO TO 96
08110
08120      120 CONTINUE
08130      IF(J1.GT.0 .AND. J2.LE.NCAP) GO TO 121
08140      WRITE(ITY,11) BELL
08150      IF(J1.LE.0 .OR. J2.GT.NCAP) WRITE(ITY,12) NCAP
08160      GO TO 95
08170
08180      121 CONTINUE
08190      DO 122 J = J1,J2
08200      SCATCH(J) = CON(2)
08210      122 CONTINUE
08220      WRITE(ITY,13)
08230      WRITE(ITY,14)
08240      GO TO 96

```

```

001300 120 CONTINUE
001310 IF (J.EQ.0 .AND. J.PAGE.NE.0 .AND. J2.EE.NCONAC) GO TO 126
001320 WRITE (UNIT,11) B11
001330 IF (J.EE.0 .OR. J2.EE.NCONAC) WRITE (UNIT,12) NCONAC
001340 GO TO 95
001350 125 CONTINUE
001360 DO 127 J = J1,J2
001370 Z(J) = GM(2)
001380 127 CONTINUE
001390 WRITE (UNIT,13)
001400 WRITE (UNIT,14)
001410 GO TO 96
001420 130 CONTINUE
001430 WRITE (LP,1)
001440 WRITE (LP,5) NFEK,IFER
001450 WRITE (LP,4) NEAF,NCON,NFEBAC,NCONAC,NACT
001460 WRITE (LP,7)
001470 CALL FRK1(2,NCONAC,15HINITIUM ACTIVITY,ACTLAB(1,NCST),LP,132)
001480 CALL FRK1(3,NCONAC,15H(UTILITY-ELAS),ACTLAB(1,NCST),
001490 1 LP,132)
001500 1 CALL FRK1(GAMMA(NCST),NCONAC,15H WEIGHT FACTOR,
001510 1 ACTLAB(1,NCST),LP,132)
001520 CALL FRK1(B(NCAF),NCON,15HETA F-CONSUMBL,ACTLAB(1,NCAP),LP,132)
001530 CALL FRK1(B,NCAF,15HBETA CAPL-TRADE,CAPLAB,LP,132)
001540 CALL FRK1(SCRATCH,NCAF,15H FRACTIONAL RES,CAPLAB,LP,132)
001550 WRITE (LP,10)
001560 CALL FRK1(X,NACT,15H TOWAY'S ACTIVITY,ACTLAB,LP,132)
001570 CALL FRK1(R,NCAF,15H RESOURCES,CAPLAB,LP,132)
001580 CALL FRK1(FLAG,NCAF,15H GESTATION TIME,CAPLAB,LP,132)
001590 B(NCAF) = -B(NCAF)
001600 CALL FRK1(D,NCAF,15H DEPRECIATION,CAPLAB,LP,132)
001610 B(NCAF) = -B(NCAF)
001620 CALL FRK1(GROWTH,NCAF,15H GROWTH FACTOR,CAPLAB,LP,132)
001630 CALL FRK1(DISCNT,NCAF,15H DISCOUNTS,CAPLAB,LP,132)
001640 CALL FRK2(ACON,NCON,15HCONSUMPT MATRIX,CONLAB,
001650 1 ACTLAB,LP,132)
001660 CALL FRK1(AGENT,NCAF,15H RENTAL MATRIX,CAPLAB,LP,132)
001670 CALL FRK1(AGENT(NCAF),NCON,15H LABOR RENTAL,
001680 1 ACTLAB(1,NCAP),LP,132)
001690 REMIND IDAT
001700 WRITE (IDAT) NACT,NCAF,NCON
001710 WRITE (IDAT) CAPLAB,CONLAB
001720 WRITE (IDAT) ACTLAB
001730 WRITE (IDAT) X
001740 WRITE (IDAT) R
001750 WRITE (IDAT) FLAG
001760 WRITE (IDAT) B
001770 WRITE (IDAT) DISCNT
001780 WRITE (IDAT) GROWTH
001790 WRITE (IDAT) ARENT
001800 WRITE (IDAT) (SCRATCH(J),J=1,NCAP)
001810 WRITE (IDAT) B
001820 WRITE (IDAT) (GAMMA(1),1-NCST,NACT,1)
001830 WRITE (IDAT) Z
001840 WRITE (IDAT) ACUN
001850 WRITE (IDAT) ITRADE
001860 CLOSE (UNIT=IDAT,DEVICE='DISK',FILE='ECO.DAT')
001870 RETURN

```



```

09350 CONTINUE
09360 J1 = INDX(1)
09370 J2 = INDX(2)
09380 IF (BETA(J1).GT. 1.0) GO TO 75
09390 FACTOR = (ILAMB(J1)*K(J1)) / (ILAMB(J2)*K(J2))
09400 FACTOR = FACTOR / (FACTOR + 1.0)
09410 BETA(J1) = LOG(FACTOR) / LOG(-BETA(J1))
09420 CONTINUE
09430 BETA(J2) = BETA(J1)
09440 BETA0 = -BETA(J1)
09450 BETA1 = BETA(J1) + 1.0
09460 BETA2 = -1.0 / BETA(J1)
09470 GAMMA = (K(J1)/K(J2))*BETA1 * (ILAMB(J1)/ILAMB(J2))
09480 GAMMA(J1) = GAMMA / (1.0 + GAMMA)
09490 GAMMA(J2) = 1.0 / (1.0 + GAMMA)
09500 FUDGE(J1) = (K(J1)/K(J2)) / ((GAMMA(J1)*K(J1))*BETA0 +
* GAMMA(J2)*K(J2))*BETA2)
09510 FUDGE(J2) = FUDGE(J1)
09520 CAPEFF = K(J1) + K(J2)
09530 EFFLAM = (ILAMB(J1)*K(J1) + ILAMB(J2)*K(J2))
09540 EFFLAM = EFFLAM / CAPEFF
09550 CONTINUE
09560 XLABOR = X(JCON)*ARENT(JCON)
09570 ARENT(JCON) = XLABOR / CAPEFF
09580 IF (BETA(JCON).GE.-1.0) GO TO 85
09590 FACTOR = EFFLAM * CAPEFF / XLABOR
09600 FACTOR = FACTOR / (FACTOR + 1.0)
09610 BETA(JCON) = -LOG(FACTOR) / LOG(-BETA(JCON))
09620 CONTINUE
09630 BETA0 = -BETA(JCON)
09640 BETA1 = 1.0 + BETA(JCON)
09650 BETA2 = -1.0 / BETA(JCON)
09660 GAMMA = EFFLAM * (CAPEFF/XLABOR)*BETA1
09670 GAMMA(JCON) = GAMMA / (1.0 + GAMMA)
09680 IF (GAMMA(JCON).GT. 0.999) GAMMA(JCON) = -1.0/(GAMMA+1.0)
09690 GN = GAMMA(JCON)
09700 G1 = 1.0 - GAMMA(JCON)
09710 IF (GAMMA(JCON).GT. 0.0) GO TO 100
09720 G1 = 1.0 - G1
09730 CONTINUE
09740 FUDGE(JCON) = X(JCON) / ( (GN*CAPEFF)*BETA0 +
1 G1*XLABOR*BETA2)
09750 CONTINUE
09760 ILAMB(NCAP) = 1.0
09770 C..... PRINT TODAY'S ECONOMY
09780 C
09790 WRITE(1,2)
09800 CALL PRK1(BETA(NCAP),NCON,15H BETA P-CONSUMBL,
1 ACTLAB(1,NCAP),LP,132)
09810 CALL PRK1(SK(1),NCONAC,15H UTILITY WEIGHT,ACTLAB(1,NCST),LP,132)
09820 CALL PRK1(15H AMB,NCAP,15H RENTAL VALUE,CAFLAB,LP,132)
09830 CALL PRK1(GAMMA(NCAP),NCON,15H FRACN CAPITAL,
1 ACTLAB(1,NCAP),LP,132)
09840 WRITE(1,3)

```



```

10510 C
10520 C
10530 C
10540 C
10550 C
10560 C
10570 C
10580 C
10590 C
10600 C
10610 C
10620 C
10630 C
10640 C
10650 C
10660 C
10670 C
10680 C
10690 C
10700 C
10710 C
10720 C
10730 C
10740 C
10750 C
10760 C
10770 C
10780 C
10790 C
10800 C
10810 C
10820 C
10830 C
10840 C
10850 C
10860 C
10870 C
10880 C
10890 C
10900 C
10910 C
10920 C
10930 C
10940 C
10950 C
10960 C
10970 C
10980 C
10990 C
11000 C
11010 C
11020 C
11030 C
11040 C
11050 C
11060 C
11070 C

CALCULATE LABUK
CSTLAB = DLAMB(NCAP)
SUML = 0.0
DO 40 I = 1, NCON
  II = 1 + NCAPM1
  GN = GAMMAP(II)
  GL = 1.0 - GN
  IF (GN .GT. 0.0) GO TO 45
  GA = -GN
  GK = 1.0 - GL
  CONTINUE
45 BETA0 = BETA(II)
  BETA1 = 1.0 / (BETA0+1.0)
  BETA2 = (BETA0+1.0) / BETA0
  RATIO = ( (GL*EFFLAB(II)) / (GN*CSTLAB) ) ** BETA1
  EFFLAB(II) = RATIO*EFFCAP(II)
  SUML = SUML + EFFLAB(II)
  FRMLAM(II) = (EFFLAB(II) / (GN*FRUGRE(II))) *
    (GL / (RATIO**BETA0) + GN) ** BETA2
  DO 50 J = 1, NCON
    FRMLAM(J) = FRMLAM(J) + ACON(J,II)*FLAM(J+NCAP)
  50 CONTINUE
  FRMLAM(1) = FRMLAM(1) + ACON(NCON+1,II) -
    ACON(1,II)*FLAM(1+NCAP)
  60 CONTINUE
  X(NCST) = N(NCAP) - SUML
  DLAMB(NCAP) = GAMMA(1) / ((X(NCST)-Z(1))**B(1))
  CHECK FOR CONSISTENCY IN FLAM'S
  USE FRMLAM ARRAY TO STORE NEW DERIVED FLAM
  IFLAG = -1
  DIF = DLAMB(NCAP) - CSTLAB
  AVG = (DLAMB(NCAP) + CSTLAB) / 2.0
  IF (ABS(DIF)/AVG .LE. ERRTOR) GO TO 65
  IFLAG = 11
  DLAMB(NCAP) = AVG
  65 CONTINUE
  DO 75 I = 1, NCON
    II = 1 + NCAPM1
    JJ = 1 + NCAP
    XIAM = FRMLAM(II) / (1.0 - ACON(1,II))
    DIF = FLAM(JJ) - XIAM
    AVG = (FLAM(JJ) + XIAM) / 2.0
    IF (ABS(DIF)/AVG .GT. 1.0E-04) IFLAG = 11
    FRMLAM(II) = XIAM
  75 CONTINUE
  IF (IFLAG .NE. 0) GO TO 85
  DO 80 I = 1, NCON
    JJ = 1 + NCAP
    FLAM(JJ) = 0.5 * (FRMLAM(1) + FLAM(JJ))
  80 CONTINUE
  GO TO 25
  85 CONTINUE

```

```

11080 C          CALCULATE PRODUCTION OF CONSUMABLES ACTIVITIES
11090 C
11100 C          CALL XMT1(C,NCON(0,0),ICONS)
11110 C
11120 C          DO 100 I = 1,NCON
11130 C             I1 = I + NCAFM1
11140 C             GN = GAMMA(I1)
11150 C             GI = 1.0 - GN
11160 C             IF (GN .GT. 0.0) GO TO 90
11170 C             GI = GN
11180 C             GN = 1.0 - GI
11190 C             CONTINUE
11200 C             X(I1) = 0.0
11210 C             IF (EFFECT(C1).LE.0.0 .OR. EFFLAB(I1).LE.0.0) GO TO 100
11220 C             X(I1) = GN * EFFECT(I1)**(-BETA(I1))
11230 C             X(I1) = X(I1) + GI * EFFLAB(I1)**(-BETA(I1))
11240 C             X(I1) = FUDGE(I1) * (X(I1)**(-1.0/BETA(I1)))
11250 C             DO 95 J = 1,NCON
11260 C                ICONS(J) = ICONS(J) + ACONC(J,I1)*X(I1)
11270 C             CONTINUE
11280 C
11290 C          95 CONTINUE
11300 C
11310 C          NEXT CALCULATE CONSUMPTION ACTIVITIES
11320 C
11330 C          DO 120 JJ = 1,NCONAC
11340 C             J = JJ + NPRODAC
11350 C             X(J) = 0.0
11360 C             IF (GAMMA(JJ).LE.0.0) GO TO 120
11370 C             COST = 0.0
11380 C             DO 110 I = 1,NCON
11390 C                COST = COST + ACONC(I,J)*ELAM(I1)
11400 C             CONTINUE
11410 C             COST = COST + ACONC(NCON+1,J)
11420 C             IF (COST .LE. 0.0) GO TO 120
11430 C             X(J) = (GAMMA(JJ)/COST)**(1.0/B(JJ)) + Z(JJ)
11440 C             DO 115 I = 1,NCON
11450 C                ICONS(I) = ICONS(I) + X(J)*ACONC(I,J)
11460 C             CONTINUE
11470 C
11480 C          115 CONTINUE
11490 C
11500 C          END USED FOR PRODUCTION OF CATION ACTIVITIES
11510 C
11520 C          DO 140 I = 1,NCAFM1
11530 C             I1 = I + NCON
11540 C             ICONS(I) = ICONS(I) + X(I)*ACONC(I1)
11550 C             CONTINUE
11560 C
11570 C          140 CONTINUE
11580 C
11590 C          UNDIFFERENTIATED INPUTS
11600 C
11610 C

```

```

11700 UNDIFF = 0.0
11710 NACTM = 0.0
11720 DO 150 I = 1, NACTM
11730 UNDIFF = UNDIFF + ACUM(NCONP, I) * X(I)
11740 CONTINUE
11750 X(NACT) = 0.0
11760 IF (UNDIFF .GE. 0.0) GO TO 170
11770 X(NACT) = UNDIFF / ACUM(NCONP, NACT)
11780 DO 160 I = 1, NCON
11790 ICONS(I) = ICONS(I) + ACUM(I, NACT) * X(NACT)
11800 CONTINUE
11810 CONTINUE
11820 CHECK OUT ACTIVITIES TO SEE IF PRODUCED ENOUGH
11830 IF LAG = -1,
11840 DO 100 I = 1, NCON
11850 II = I + NCAPI
11860 DIF = ICONS(I) - X(II)
11870 AVG. = X(II) + ICONS(I)
11880 IF (ABS(DIF)/AVG .GT. ENRTOL) IF LAG = 11
11890 DO 175 J = 1, NCAPI
11900 IF (ITRABE(J) .NE. I) GO TO 175
11910 R(J) = R(J) * (1.0 + DIF/AVG)
11920 CONTINUE
11930 CONTINUE
11940 IF (IFLAG.GT.0) GO TO 20
11950 CONVERGENCE
11960 C
11970 C
11980 C
11990 C
12000 C
12010 C
12020 C
12030 C
12040 C
12050 C
12060 C
12070 C
12080 C
12090 C
12100 C
12110 C
12120 C
12130 C
12140 C
12150 C
12160 C
12170 C
12180 C
12190 C
12200 C
12210 C
12220 C
12230 C
12240 C
12250 C
12260 C
12270 C
12280 C
12290 C
12300 C
12310 C
12320 C
12330 C
12340 C
12350 C
12360 C
12370 C
12380 C
12390 C
12400 C
12410 C
12420 C
12430 C
12440 C
12450 C
12460 C
12470 C
12480 C
12490 C
12500 C
12510 C
12520 C
12530 C
12540 C
12550 C
12560 C
12570 C
12580 C
12590 C
12600 C
12610 C
12620 C
12630 C
12640 C
12650 C
12660 C
12670 C
12680 C
12690 C
12700 C
12710 C
12720 C
12730 C
12740 C
12750 C
12760 C
12770 C
12780 C
12790 C
12800 C
12810 C
12820 C
12830 C
12840 C
12850 C
12860 C
12870 C
12880 C
12890 C
12900 C
12910 C
12920 C
12930 C
12940 C
12950 C
12960 C
12970 C
12980 C
12990 C
13000 C
13010 C
13020 C
13030 C
13040 C
13050 C
13060 C
13070 C
13080 C
13090 C
13100 C
13110 C
13120 C
13130 C
13140 C
13150 C
13160 C
13170 C
13180 C
13190 C
13200 C
13210 C
13220 C
13230 C
13240 C
13250 C
13260 C
13270 C
13280 C
13290 C
13300 C
13310 C
13320 C
13330 C
13340 C
13350 C
13360 C
13370 C
13380 C
13390 C
13400 C
13410 C
13420 C
13430 C
13440 C
13450 C
13460 C
13470 C
13480 C
13490 C
13500 C
13510 C
13520 C
13530 C
13540 C
13550 C
13560 C
13570 C
13580 C
13590 C
13600 C
13610 C
13620 C
13630 C
13640 C
13650 C
13660 C
13670 C
13680 C
13690 C
13700 C
13710 C
13720 C
13730 C
13740 C
13750 C
13760 C
13770 C
13780 C
13790 C
13800 C
13810 C
13820 C
13830 C
13840 C
13850 C
13860 C
13870 C
13880 C
13890 C
13900 C
13910 C
13920 C
13930 C
13940 C
13950 C
13960 C
13970 C
13980 C
13990 C
14000 C
14010 C
14020 C
14030 C
14040 C
14050 C
14060 C
14070 C
14080 C
14090 C
14100 C
14110 C
14120 C
14130 C
14140 C
14150 C
14160 C
14170 C
14180 C
14190 C
14200 C
14210 C
14220 C
14230 C
14240 C
14250 C
14260 C
14270 C
14280 C
14290 C
14300 C
14310 C
14320 C
14330 C
14340 C
14350 C
14360 C
14370 C
14380 C
14390 C
14400 C
14410 C
14420 C
14430 C
14440 C
14450 C
14460 C
14470 C
14480 C
14490 C
14500 C
14510 C
14520 C
14530 C
14540 C
14550 C
14560 C
14570 C
14580 C
14590 C
14600 C
14610 C
14620 C
14630 C
14640 C
14650 C
14660 C
14670 C
14680 C
14690 C
14700 C
14710 C
14720 C
14730 C
14740 C
14750 C
14760 C
14770 C
14780 C
14790 C
14800 C
14810 C
14820 C
14830 C
14840 C
14850 C
14860 C
14870 C
14880 C
14890 C
14900 C
14910 C
14920 C
14930 C
14940 C
14950 C
14960 C
14970 C
14980 C
14990 C
15000 C
15010 C
15020 C
15030 C
15040 C
15050 C
15060 C
15070 C
15080 C
15090 C
15100 C
15110 C
15120 C
15130 C
15140 C
15150 C
15160 C
15170 C
15180 C
15190 C
15200 C
15210 C
15220 C
15230 C
15240 C
15250 C
15260 C
15270 C
15280 C
15290 C
15300 C
15310 C
15320 C
15330 C
15340 C
15350 C
15360 C
15370 C
15380 C
15390 C
15400 C
15410 C
15420 C
15430 C
15440 C
15450 C
15460 C
15470 C
15480 C
15490 C
15500 C
15510 C
15520 C
15530 C
15540 C
15550 C
15560 C
15570 C
15580 C
15590 C
15600 C
15610 C
15620 C
15630 C
15640 C
15650 C
15660 C
15670 C
15680 C
15690 C
15700 C
15710 C
15720 C
15730 C
15740 C
15750 C
15760 C
15770 C
15780 C
15790 C
15800 C
15810 C
15820 C
15830 C
15840 C
15850 C
15860 C
15870 C
15880 C
15890 C
15900 C
15910 C
15920 C
15930 C
15940 C
15950 C
15960 C
15970 C
15980 C
15990 C
16000 C
16010 C
16020 C
16030 C
16040 C
16050 C
16060 C
16070 C
16080 C
16090 C
16100 C
16110 C
16120 C
16130 C
16140 C
16150 C
16160 C
16170 C
16180 C
16190 C
16200 C
16210 C
16220 C
16230 C
16240 C
16250 C
16260 C
16270 C
16280 C
16290 C
16300 C
16310 C
16320 C
16330 C
16340 C
16350 C
16360 C
16370 C
16380 C
16390 C
16400 C
16410 C
16420 C
16430 C
16440 C
16450 C
16460 C
16470 C
16480 C
16490 C
16500 C
16510 C
16520 C
16530 C
16540 C
16550 C
16560 C
16570 C
16580 C
16590 C
16600 C
16610 C
16620 C
16630 C
16640 C
16650 C
16660 C
16670 C
16680 C
16690 C
16700 C
16710 C
16720 C
16730 C
16740 C
16750 C
16760 C
16770 C
16780 C
16790 C
16800 C
16810 C
16820 C
16830 C
16840 C
16850 C
16860 C
16870 C
16880 C
16890 C
16900 C
16910 C
16920 C
16930 C
16940 C
16950 C
16960 C
16970 C
16980 C
16990 C
17000 C
17010 C
17020 C
17030 C
17040 C
17050 C
17060 C
17070 C
17080 C
17090 C
17100 C
17110 C
17120 C
17130 C
17140 C
17150 C
17160 C
17170 C
17180 C
17190 C
17200 C
17210 C
17220 C
17230 C
17240 C
17250 C
17260 C
17270 C
17280 C
17290 C
17300 C
17310 C
17320 C
17330 C
17340 C
17350 C
17360 C
17370 C
17380 C
17390 C
17400 C
17410 C
17420 C
17430 C
17440 C
17450 C
17460 C
17470 C
17480 C
17490 C
17500 C
17510 C
17520 C
17530 C
17540 C
17550 C
17560 C
17570 C
17580 C
17590 C
17600 C
17610 C
17620 C
17630 C
17640 C
17650 C
17660 C
17670 C
17680 C
17690 C
17700 C
17710 C
17720 C
17730 C
17740 C
17750 C
17760 C
17770 C
17780 C
17790 C
17800 C
17810 C
17820 C
17830 C
17840 C
17850 C
17860 C
17870 C
17880 C
17890 C
17900 C
17910 C
17920 C
17930 C
17940 C
17950 C
17960 C
17970 C
17980 C
17990 C
18000 C
18010 C
18020 C
18030 C
18040 C
18050 C
18060 C
18070 C
18080 C
18090 C
18100 C
18110 C
18120 C
18130 C
18140 C
18150 C
18160 C
18170 C
18180 C
18190 C
18200 C
18210 C
18220 C
18230 C
18240 C
18250 C
18260 C
18270 C
18280 C
18290 C
18300 C
18310 C
18320 C
18330 C
18340 C
18350 C
18360 C
18370 C
18380 C
18390 C
18400 C
18410 C
18420 C
18430 C
18440 C
18450 C
18460 C
18470 C
18480 C
18490 C
18500 C
18510 C
18520 C
18530 C
18540 C
18550 C
18560 C
18570 C
18580 C
18590 C
18600 C
18610 C
18620 C
18630 C
18640 C
18650 C
18660 C
18670 C
18680 C
18690 C
18700 C
18710 C
18720 C
18730 C
18740 C
18750 C
18760 C
18770 C
18780 C
18790 C
18800 C
18810 C
18820 C
18830 C
18840 C
18850 C
18860 C
18870 C
18880 C
18890 C
18900 C
18910 C
18920 C
18930 C
18940 C
18950 C
18960 C
18970 C
18980 C
18990 C
19000 C
19010 C
19020 C
19030 C
19040 C
19050 C
19060 C
19070 C
19080 C
19090 C
19100 C
19110 C
19120 C
19130 C
19140 C
19150 C
19160 C
19170 C
19180 C
19190 C
19200 C
19210 C
19220 C
19230 C
19240 C
19250 C
19260 C
19270 C
19280 C
19290 C
19300 C
19310 C
19320 C
19330 C
19340 C
19350 C
19360 C
19370 C
19380 C
19390 C
19400 C
19410 C
19420 C
19430 C
19440 C
19450 C
19460 C
19470 C
19480 C
19490 C
19500 C
19510 C
19520 C
19530 C
19540 C
19550 C
19560 C
19570 C
19580 C
19590 C
19600 C
19610 C
19620 C
19630 C
19640 C
19650 C
19660 C
19670 C
19680 C
19690 C
19700 C
19710 C
19720 C
19730 C
19740 C
19750 C
19760 C
19770 C
19780 C
19790 C
19800 C
19810 C
19820 C
19830 C
19840 C
19850 C
19860 C
19870 C
19880 C
19890 C
19900 C
19910 C
19920 C
19930 C
19940 C
19950 C
19960 C
19970 C
19980 C
19990 C
20000 C

```

```

12110      F = 1
12120      IF (1 - LE, 10) F = F / 100.
12130      IF (1 - GI, 10 AND, 1 - LE, 19) F = (F - 9.0) / 10.
12140      IF (1 - GI, 19 AND, 1 - LE, 26) F = F - 10.0
12150      IF (F - GI, 20) F = (F - 27.0) * 10.
12160      DO 480 J = JSTART, JSTOP
12170          JJ = J + NCAPM1
12180          GN = GAMMAF(JJ)
12190          GL = 1.0 - GN
12200          IF (GN - GI, 0.0) GO TO 450
12210          GL = -GN
12220          GN = 1.0 - GL
12230          CONTINUE
12240          XNDR = (GN + GL * RENTB(J) * (-BETA(JJ))) * (-1.0 / BETA(JJ))
12250          XFAC = (GN + GL * RENTB(J)) * (-BETA(JJ)) * (-1.0 / BETA(JJ))
12260          SGRICH(J) = XFAC / XNDR
12270          480 CONTINUE
12280          WRITE(LP,3) F, (SGRICH(J), J = JSTART, JSTOP)
12290          500 CONTINUE
12300          600 CONTINUE
12310          RETURN
12320          END
12330          SUBROUTINE TIMEP(NCON, NCAP, NCAPM1, NCST, NPERFAC, NCUNAC, NACT, NPER,
12340              1 NCONF, NPERF1, NPERF2, K, RLAST, GRDTH,
12350              2 D, T, FLAG, DIA, FLAM, FLAMRT, NGT, NFUT, X, Z, GAMMA, B, LTRAD, FUDGE,
12360              3 EFFCAP, EFFLAB, ACON, KSTART, SCRICH,
12370              4 EFFCT, EFFLAM, TCINS, RED, PMLAN, NSCALE)
12380              C
12390              C
12400              C
12410              DIMENSION KCONC(NPERF2), RLAST(NCAP, NPERF2), GRDTH(NCAPM1),
12420                  1 KCONC(NPERF2), TLAB(NCAPM1), DIA(NCAPM1), RED(NCAP),
12430                  2 DIA(NCAP, NPER), FLAM(NST, NPERF1), RINGT(NCAPM1), X(NACT, NPER),
12440                  3 Z(NCONAC), GAMMA(NACT), R(NACT), NSTART(NCAP), SGRICH(NST),
12450                  4 LTRAD(NCAPM1), FUDGE(NPERAC), EFFCAP(NCON), LEFTLAB(NCON),
12460                  5 ACON(NCON), L(NACT), PMLAN(NCON), NFUT(NCAPM1), NSCALE(NCAPM1),
12470                  6 FFLAN(NCON), TCINS(NCON), LEFTST(NCAPM1, NPER)
12480              C
12490              LOGICAL LVERKE1
12500              LOGICAL FINITEF5
12510              C
12520              COMMON /EVLN1/ LVERKE1
12530              COMMON /LUMP/ LUMP1, LUMP2
12540              COMMON /MKT1/ MKT1, MKT2
12550              COMMON /MIXER/ NCMIX
12560              COMMON /PASS/ IPASS
12570              COMMON /PERIOD/ TPER
12580              COMMON /FRONT/ FRNTF5
12590              COMMON /SLOPE/ SLP, INS, DIA1, LP, INEEND
12600              COMMON /SLOPE/ SLP1, MAXLIK
12610              COMMON /TOR INC/ ENRDI
12620              C

```

```

12630 1 FORMAT(10F10.3, 1X, 'END OF PERIOD', 13)
12640 2 FORMAT(1H, 6X, 'R', 10G12.4)
12650 3 FORMAT(1H, 6X, 'FLAM', 10G12.4)
12660 4 FORMAT(1H, 6X, 'TURNS', 10G12.4)
12670 5 FORMAT(1H, 6X, 'KING', 10G12.4)
12680 6 FORMAT(1H, 6X, 'XING', 10G12.4)
12690 7 FORMAT(1H, 6X, 'XING', 10G12.4)
12700 8 FORMAT(1H, 6X, 'XING', 10G12.4)
12710 9 FORMAT(1H, 6X, 'XING', 10G12.4)
12720 10 FORMAT(1H, 6X, 'XING', 10G12.4)
12730 11 FORMAT(1H, 6X, 'XING', 10G12.4)
12740 12 FORMAT(1H, 6X, 'XING', 10G12.4)
12750 13 FORMAT(1H, 6X, 'XING', 10G12.4)
12760 14 FORMAT(1H, 6X, 'XING', 10G12.4)
12770 15 FORMAT(1H, 6X, 'XING', 10G12.4)
12780 16 FORMAT(1H, 6X, 'XING', 10G12.4)
12790 C
12800 LIMP = 0
12810 SUM1 = 0.0
12820 SUM2 = 0.0
12830 NCAFF1 = NCAF + 1
12840 DO 220 N = 1, NPERF*2
12850 DO 200 J = 1, NCAFF1
12860 R(J,N) = RSTART(J)*EXP((KROTH(J)*T(N))
12870 IF(T(N).GT. TLAB(J)) R(J,N) = RSTART(J) * EXP( GROWTH(J) *
12880 * (TLAB(J) - (BCJ)*D(NCAF))*(T(N)-TLAB(J)) )
12890 200 CONTINUE
12900 R(NCAF,N) = RSTART(NCAF)
12910 220 CONTINUE
12920 IF(PRINTPS) WRITE(1P,0) IPASS
12930 C
12940 DO 500 N = 1, NPER
12950 N1 = N + 1
12960 IF(EVERET) GO TO 305
12970 C
12980 DO 300 J = 1, NCAFF1
12990 IF(IPASS.NE.1) GO TO 240
13000 IF(N.EQ.1) GO TO 250
13010 CALL XHIT(NCON,X(NCAF,N1),X(NCAF,N))
13020 CALL XHIT(NCON,LAH(NCAF+1,N),FLAH(NCAF+1,N1))
13030 GO TO 250
13040 C
13050 CONTINUE
13060 R(J,N) = (R(J,N) + RLAST(J,N)*(RESMIX-1.0)) / RESMIX
13070 CONTINUE
13080 C..... SET FUTURE RESOURCE LEVEL ...
13090 NN = (T(N1) + 1.5)*T(N) + TLAB(J)/T(N) + 0.001
13100 NFUT(J) = R(J,N) * EXP(-(BCJ)*D(NCAF))*(T(N1)/2.0 + DTAU(J))
13110 C..... SET RESOURCE TARGET
13120 RTF = RLAST(J,N) - RLAST(J,N-1)
13130 IF(DIF.GE.0.0) RINGT(J) = RLAST(J,N)
13140 IF(DIF.LE.0.0) RINGT(J) = RLAST(J,N-1)
13150 IF(DIF.LE.0.0) RINGT(J) = RLAST(J,N) * EXP(
13160 * (LOG(RLAST(J,N-1)/RLAST(J,N)) * (T(N1)/2.0 + DTAU(J)))
13170 * (RTF - EXP(-(BCJ)*D(NCAF))*(1.0 EXP(-(BCJ)*D(NCAF))*(T(N1)/

```

```

1190 FACTR = ITER / FACTR
1200 GO TO 1000
1210 GO TO 1000
1220 GO TO 1000
1230 GO TO 1000
1240 GO TO 1000
1250 GO TO 1000
1260 GO TO 1000
1270 GO TO 1000
1280 GO TO 1000
1290 GO TO 1000
1300 GO TO 1000
1310 GO TO 1000
1320 GO TO 1000
1330 GO TO 1000
1340 GO TO 1000
1350 GO TO 1000
1360 GO TO 1000
1370 GO TO 1000
1380 GO TO 1000
1390 GO TO 1000
1400 GO TO 1000
1410 GO TO 1000
1420 GO TO 1000
1430 GO TO 1000
1440 GO TO 1000
1450 GO TO 1000
1460 GO TO 1000
1470 GO TO 1000
1480 GO TO 1000
1490 GO TO 1000
1500 GO TO 1000
1510 GO TO 1000
1520 GO TO 1000
1530 GO TO 1000
1540 GO TO 1000
1550 GO TO 1000
1560 GO TO 1000
1570 GO TO 1000
1580 GO TO 1000
1590 GO TO 1000
1600 GO TO 1000
1610 GO TO 1000
1620 GO TO 1000
1630 GO TO 1000
1640 GO TO 1000
1650 GO TO 1000
1660 GO TO 1000
1670 GO TO 1000
1680 GO TO 1000
1690 GO TO 1000
1700 GO TO 1000
1710 GO TO 1000
1720 GO TO 1000
1730 GO TO 1000
1740 GO TO 1000
1750 GO TO 1000
1760 GO TO 1000
1770 GO TO 1000
1780 GO TO 1000
1790 GO TO 1000
1800 GO TO 1000
1810 GO TO 1000
1820 GO TO 1000
1830 GO TO 1000
1840 GO TO 1000
1850 GO TO 1000
1860 GO TO 1000
1870 GO TO 1000
1880 GO TO 1000
1890 GO TO 1000
1900 GO TO 1000
1910 GO TO 1000
1920 GO TO 1000
1930 GO TO 1000
1940 GO TO 1000
1950 GO TO 1000
1960 GO TO 1000
1970 GO TO 1000
1980 GO TO 1000
1990 GO TO 1000
2000 GO TO 1000

```


121

```

14960 IF (COST(I)) GO TO 85
14970 SUM = SUM + ACUN(I,J)*EFFECT(I,J)
14980 CONTINUE
14990 SUM = SUM + ACUN(CONF1,J)
15000 XLAN = (PRMAM(I)*SUM) / (1.0*ACUN(I,J))
15010 DIFF = FLAM(I,J) - XLAN
15020 AVG = FLAM(I,J) + XLAN
15030 IF (ABS(DIFF)/AVG .GT. ENKTOL) IFLAG = 11
15040 FLAM(I,J) = 0.5*(FLAM(I,J) + XLAN)
15050 CONTINUE
15060 IF (IFLAG.GT.0.0) GO TO 80
15070 C
15080 C FIND EFFECTIVE COST FOR PRODUCING CAPITAL
15090 C
15100 DO 105 J = 1,NCAPM1
15110 EFFECT(J) = 0.0
15120 DO 100 I = 1,NCUN
15130 II = I + NCAP
15140 EFFECT(J) = EFFECT(J) + ACUN(I,J)*FLAM(II)
15150 CONTINUE
15160 EFFECT(J) = EFFECT(J) + ACUN(CONF1,J)
15170 CONTINUE
15180 C
15190 C NEXT CALCULATE CONSUMPTION ACTIVITIES
15200 C
15210 DO 120 JJ = 2,NCUNAM
15220 J = JJ + NFRDAM
15230 X(J) = 0.0
15240 IF (GAMMA(I,J) .LE. 0.0) GO TO 120
15250 COST = 0.0
15260 DO 110 I = 1,NCUN
15270 II = I + NCAP
15280 COST = COST + ACUN(I,J)*FLAM(II)
15290 CONTINUE
15300 COST = COST + ACUN(CONF1,J)
15310 IF (COST .LE. 0.0) GO TO 120
15320 X(J) = (GAMMA(I,J)/COST)*(1.0/B(J,J)) + Z(J,J)
15330 DO 115 I = 1,NCUN
15340 ICONS(I) = ICONS(I) + X(J)*ACUN(I,J)
15350 CONTINUE
15360 CONTINUE
15370 C
15380 C FIND USED FOR PRODUCTION OF CAPITAL ACTIVITIES
15390 C
15400 DO 140 I = 1,NCAPM1
15410 IF (EVERET) GO TO 125
15420 RSC1 = RSCAM(I) * SLPN I*(I)/REDC(I)
15430 X(I) = RTRGT(I)*R(I) - (RSC1*AVG(EFFECT(I)/FLAM(I)))
15440 X(I) = X(I) / IFEK
15450 X(I) = AMAX1(X(I),0.0)
15460 CONTINUE
15470 DO 130 J = 1,NCUN
15480 TCONS(J) = TCONS(J) + X(I)*ACUN(I,J)
15490 CONTINUE
15500 CONTINUE
15510 C
15520 C UNDIFFERENTIATED IMPORTS

```

```

15340 C
15340 UNIMP = 0.0
15340 NACTM1 = NACT
15340 DO 150 I = 1, NACTM1
15340 UNIMP = UNIMP + ACON(NLUMP+1, I)*X(I)
15340
15340 C
15340 X(NACT) = 0.0
15340 IF (UNIMP .LT. 0.0) GO TO 160
15340 X(NACT) = UNIMP / ACON(NCON+1, NACT)
15340 DO 155 I = 1, NCON
15340 ICUNS(I) = ICUNS(I) + ACON(I, NACT)*X(NACT)
15340
15340 C
15340 CONTINUE
15340
15340 C
15340 CALCULATE DELTA LAMIDAS (RENTAL OF CAPITAL)
15340
15340 C
15340 DO 170 I = 1, NCON
15340 DO 165 J = 1, NCAPI
15340 IF (ITRARE(J).NE.1) GO TO 165
15340 BLANK(J) = EFFLAP(I)*FUNCE(J)*GAMMAP(J)*((EFFCAP(I)/
15340 (FUNCE(I)*K(J))**BETA(J)+1.0))
15340
15340 C
15340 CONTINUE
15340
15340 C
15340 CHECK UNIT ACTIVITIES TO SEE IF PRODUCED ENOUGH
15340
15340 C
15340 IFLAG = 1
15340 JFLAG = -1
15340 SUML = 0.0
15340 DO 200 I = 1, NCON
15340 II = I + NCAPI
15340 GN = GAMMAP(II)
15340 IF (GN .LT. 0.0) GN = 1.0 + GN
15340 REFAO = -1.0 / BETA(II)
15340 ACTMAX = 0.9*AFUNCE(II)*EFFCAP(I)*GN**BETAO
15340 IIF = ICUNS(I) * X(II)
15340 AVG = ICUNS(I) + X(II)
15340 IF (ABS(IIF)/AVG.GT.ERRTOL) IFLAG = +1
15340 IF (ABS(SCRICH(I)).GT.ERRTOL) JFLAG = +1
15340 FO = -BETA
15340 IF (SCRICH(I)*IIF .GT. 0.0) FO = FUNCE*DELTA
15340 IF (IIF .GT. 1) FO = 0.0
15340 SCRICH(I) = ABS(SCRICH(I)) * (1.0/FO)
15340 SCRICH(I) = AMIN(SCRICH(I), 0.9)
15340 SCRICH(I) = SIGN(SCRICH(I), IIF)
15340
15340 C
15340 CONTINUE
15340 IF (JFLAG.NE.1.0) IFLAG = -1
15340 IF (IFLAG.NE.0.0) IIFER, IIFER.GT.MAXIIFR) GO TO 250
15340 DO 201 I = 1, NCON
15340 FRMAM(I) = FRMAM(I) + (1.0/SCRICH(I))
15340
15340 C
15340 CONTINUE
15340 DO 220 I = 1, NCON
15340 II = I + NCAPI
15340 GN = GAMMAP(II)
15340 IF (GN .LT. 0.0) GN = 1.0
15340 IF (GN .GT. 0.0) GO TO 200
15340 GN = GN
15340 GN = 1.0 - GL
15340
15340 C
15340 CONTINUE
15340
15340 C

```

```

16130 FACTOR = (INDEX(1) + G28K(NCON,AM(1)/14 + MB(NCAP)
16140 B1 = BETA(1) / (BETA(1) + 1.0)
16150 B2 = 1.0/BETA(1)
16160 FACTOR = (FACTOR**B1) - B1/B2
16170 EFFLAB(1) = 0.0
16180 IF (FACTOR .GT. 0.0) EFFLAB(1) = EFFCAP(1) *
16190 FACTOR**B2
16200 SUM = SUM + EFFLAB(1)
16210 CONTINUE
16220 IF (SUM .LT. K(NCAP)) GO TO 20
16230 FACTOR = 0.9**K(NCAP)/SUM
16240 DO 225 I = 1,NCON
16250 EFFLAB(I) = FACTOR*EFFLAB(I)
16260 CONTINUE
16270 GO TO 20
16280 CONTINUE
16290 RETURN
16300 END
16310 SUBROUTINE EFFCAP(EFFCAP,FUDGE,GAMMA,BETA,ITRADE,K,NCON,NCAPM1)
16320 DIMENSION EFFCAP(NCON),GAMMA(NCAPM1),BETA(NCAPM1),ITRADE(NCAPM1),
16330 I K(NCAPM1),FUDGE(NCAPM1),INDEX(2)
16340 C
16350 C THIS SUBROUTINE OBTAINS AN EFFECTIVE LEVEL OF CAPITAL FOR USE
16360 C IN THE PRODUCTION OF THE NCON CONSUMABLES BY USING A TRANSLOG
16370 C FUNCTION TO TRADE OFF TWO TYPES OF CAPITAL.
16380 C
16390 DO 100 J = 1,NCON
16400 N = 0
16410 DO 20 I = 1,NCAPM1
16420 IF (ITRADE(I) .NE. J) GO TO 20
16430 N = N + 1
16440 INDEX(N) = I
16450 CONTINUE
16460 EFFCAP(J) = 0.0
16470 IF (N .EQ. 0) GO TO 100
16480 IF (N .EQ. 2) GO TO 40
16490 J1 = INDEX(1)
16500 EFFCAP(J) = R(J1)
16510 GO TO 100
16520 CONTINUE
16530 J1 = INDEX(1)
16540 J2 = INDEX(2)
16550 G1 = GAMMA(J1)
16560 G2 = GAMMA(J2)
16570 F = FUDGE(J)
16580 C.....NOTE THAT G2 ALSO EQUALS 1.0-G1.....
16590 C.....BETA = BETA(J1) + I*TA(J2).....
16600 EFFCAP(J) = F*(G1**R(J1)**(BETA(J1) + G28K(J2)**(BETA(J2)))
16610 I
16620 **(-1.0/BETA(J1))
16630 CONTINUE
16640 RETURN
16650 END

```

```

16640 SURKUTLINE LAMBDA(NFLR,NCAP,NCAPM1,NST,NFLP1,NFLP2,I,B,FLAM,
16650 1 ILAMB,FLAG,BTAN,DISCNT,EFFEST,REQ,IFLAG)
16660
16670 DIMENSION I(NFLP1),I(NCAP),FLAM(NST,NFLP1),DLAMB(NCAP,NFLR),
16680 1 ILAG(NCAPM1),BTAN(NCAPM1),DISCNT(NCAPM1),EFFEST(NCAPM1,NFLR),
16690 2 K(NCAP,NFLR),REQ(NCAPM1)
16700
16710 COMMON /PERIOD/ IPER
16720
16730
16740 C PROPAGATES LAMBDA BACKWARDS THROUGH TIME --
16750 C NOTE: IFLAG CONTROLS WHETHER LAMBDA ARE PROPAGATED ALL THE WAY BACK
16760 C OR ONLY TO FLAM(J,2). NORMALLY, FLAM(J,1) IS USED TO STORE THE
16770 C HORIZON LAMB, BUT IN THE LAST PASS, THIS IS NO LONGER
16780 C NECESSARY, AND FLAM FOR FIRST TIME PERIOD IS CALCULATED FOR PRINTING
16790 C IN SURKUTLINE DRIVER. THIS FIRST FLAM REPRESENTS THE IMMEDIATE
16800 C FIRST ATTACK (POST RECONSTITUTION) FLAM ....
16810
16820 ANLAST = NFLR - 1
16830 IF (IFLAG .E. 0) ANLAST = NFLR
16840 DO 200 J = 1,NCAPM1
16850 C ..... FIRST CALCULATE HORIZON FLAM
16860 100 FL = FLAM(J,1)
16870 100 = DISCNT(J) + D(NCAP) + D(J)
16880 FLAM1 = DLH * PROPL(1,0,1,0,1,0,0,0,0,ILAG(J),IPER,BTAN(J),
16890 100 D(J),D(NCAP),DISCNT(J))/(1.0-EXP(-RHO*IPER))
16900 C ..... NEXT ESTIMATE FLAM FOR LAST TIME PERIOD
16910 100 = (ILAG(J) + 0.0001) / IPER
16920 100 = NFLR + NDEL + 2
16930 100 = R(J,NFLR) * EXP((D(J)+D(NCAP))*(IPER/2. + D(NCAP)))
16940 C ..... ABOVE SAME AS CALCULATION OF 'RFL' IN IIPER
16950 FLAM(J,NFLP1) = FLAM1 * REQ(J) / RHO
16960 C ..... NEXT CALCULATE OTHER FLAMS BY PROPAGATING KENTAL VALUES
16970 100 150 NK = 1,ANLAST
16980 N = NFLP1 - NK
16990 100 = (N 1) + NDEL
17000 N1 = NO + 1
17010 N2 = NO + 2
17020 100 = DLH * NLH(J) / R(J,N0)
17030 100 = DLH * NLH(J) / R(J,N1)
17040 100 = DLH * NLH(J) / R(J,N2)
17050 IF (N0 .E. NFLR) DL1 = DLAMB(J,N0)
17060 IF (N1 .E. NFLR) DL1 = DLAMB(J,N1)
17070 IF (N2 .E. NFLR) DL2 = DLAMB(J,N2)
17080 FLAM(J,N) = PROPL(DL1,DL2,FLAM(J,N1),ILAG(J),IPER,
17090 100 CONTINUE
17100 200 CONTINUE
17110 RETURN
17120
17130 END

```

```

17140 C      FUNCTION FNU(L,R,DEL,DEL2,FL,TAU,DEL1,DTAU,B,P,K)
17150 C      FNR = 1 + B + K
17160 C      FNR1 = EXP(-FNR*DEL1)
17170 C      FNR2 = EXP(-FNR*DTAU)
17180 C      RT = EXP(-FNR*DEL1)
17190 C      RTAU = EXP(-FNR*DTAU)
17200 C      FACTOR = EXP(-FNR*DTAU) / ((P1D)*(1.0 RT)/R)
17210 C
17220 C      FNR1 = FNR*RT + FACTOR*(
17230 C      1  IN 1(1.0 RTAU)/K - (1.0-FNR2AU)/FNR ) +
17240 C      2  IN 1( RTAU*(1.0 FNR1) - FNR1 - RT)/R +
17250 C      3  IN 1( RTAU*(1.0-FNR2AU) + FNR1 - FNR2AU)/FNR ) +
17260 C      4  IN 1( RTAU*(FNR2AU-FNR1)/FNR + FNR2(RT-RTAU)/R ) )
17270 C
17280 C      RETURN
17290 C
17300 C      ENB
17310 C
17320 C      SUBROUTINE PLU(LI,NCAP,NCUN,NACT,NFER,
17330 C      1  SCKICH,R,FLAM,EFFCS,DEAMB,X,CAPLAB)
17340 C      DIMENSION SCKICH(5),R(1),FLAM(1),DEAMB(1),X(1),CAPLAB(1),EFFCS(1)
17350 C      COMMON /PERIOD/ TPER
17360 C      COMMON /DEVICE/ ITTY,INS,ITAT,LP,NKRECNO
17370 C
17380 C      CREATE FILE FOR PLOTTING ROUTINES
17390 C
17400 C      SCKICH(1) = NCAP
17410 C      SCKICH(2) = NCUN
17420 C      SCKICH(3) = NACT
17430 C      SCKICH(4) = NFER
17440 C      SCKICH(5) = TPER
17450 C      NFER1 = NFER + 1
17460 C      NCST = NCAP + 1
17470 C      NCST1 = NCST + 1
17480 C      NCAP1 = NCAP + 1
17490 C      N1 = NCUN*ENTNPF1
17500 C      N2 = NCST*ENTNPF1
17510 C      N3 = NCAP1*ENTNPF1
17520 C      N4 = NCUN*ENTNPF1
17530 C      N5 = NCST*ENTNPF1
17540 C      N6 = J0(NCST*ENTNPF1)
17550 C      UNTIL ALL LABELS ARE TOGETHER NCAP + NCUN*1 + NACT
17560 C      UNTIL (SCKICH(1),1-1,N3),(SCKICH(1),1-1,N1),
17570 C      1  (SCKICH(1),1-1,N2),(EFFCS(1),1-1,N3),(DEAMB(1),1-1,N4),
17580 C      2  (X(1),1-1,N5),(CAPLAB(1),1-1,N6)
17590 C      N3 UNO = N3*NCUN + 1
17600 C      RETURN
17610 C
17620 C      ENB
17630 C      FUNCTION SULF(NCUNAC,X,Z,GAMMA,B)
17640 C
17650 C      C..... CALCULATES STANDARD IN LIVING BASED ON ACTIVITIES AND UTILITY
17660 C      FUNCTION PARAMETERS
17670 C

```

```

1/660 DIMENSION X(NCINAC),Z(NCUNAC),GAMMA(NCUNAC),B(NCUNAC)
1/670 SURF = 0.0
1/680 DO 100 I = 1,NCUNAC
1/690 IF (GAMMA(I).LE.0.0) GO TO 100
1/700 IF (X(I)-Z(I).LE.0.0) GO TO 200
1/710 IF (B(I).EQ.1.0) GO TO 20
1/720 SOLO = ( (X(I)-Z(I))*((1.0-B(I)) - 1.0) / (1.0 - B(I))
1/730 GO TO 50
1/740 CONTINUE
1/750 SOLO = ALDB(X(I)-Z(I))
1/760 CONTINUE
1/770 SOLO = SURF + GAMMA(I)*SOLO
1/780 CONTINUE
1/790 RETURN
1/800 CONTINUE
1/810 SOLO = 0.0
1/820 RETURN
1/830 END
1/840 SUBROUTINE STOPIT
1/850 IMPLICIT INTEGER(A-Z)
1/860 LOGICAL TYPEIN
1/870 COMMON /DEVICE/ I11Y,INS,IDAT,LP,NKECNO
1/880 DATA I11Y,ICS/'615004020100','715004020100/
1/890 DATA IFC,UF,S/SHC ,SHS /
1/900 DATA BE1/'0340000000000/
1/910
1/920 C
1/930 1 FORMAT(' YOU HAVE INTERRUPTED ME --',/
1/940 * , ' I am C to let me continue or S to stop me.')
1/950 2 FORMAT(A1)
1/960 3 FORMAT('X',/,' EXCUSE ME?',A1)
1/970
1/980 C
1/990 IF (.NOT. TYPEIN(MINNY)) RETURN
100 CONTINUE
1/990 WRITE (TTY,3) BE1L
1/990 WRITE (TTY,1)
1/800 REACTIVITY,2) REPLY
1/800 IF (REPLY.EQ.UF) .OR. (REPLY.EQ.ICS) STOP
1/800 IF (REPLY.EQ.IFC .OR. REPLY.EQ.IC) RETURN
1/800 GO TO 100
1/800 END
1/800 SUBROUTINE SUMMARY (NCST,NCAT,M1,NACT,X,R,SUMMARY)
1/800 DIMENSION X(NACT),R(NCAT,M1),SUMMARY(S)
1/800

```

```

18070      NESTP1 = NEST + 1
18100      NFKDAC = NEST - 1
18110      NCAP = NCAPH + 1
18120      C
18130      CALL XNII(-5.0.0,SUMMY)
18140      C
18150      DO 100 I = NESTP1,NACT
18160      SUMMY(1) = SUMMY(1) + X(I)
18170      100 CONTINUE
18180      DO 200 I = 1,NCAPH1
18190      SUMMY(2) = SUMMY(2) + X(I)
18200      200 CONTINUE
18210      SUMMY(3) = SUMMY(1) + SUMMY(2)
18220      DO 300 I = NCAP,NFKDAC
18230      SUMMY(4) = SUMMY(4) + X(I)
18240      300 CONTINUE
18250      DO 400 I = 1,NCAPH1
18260      SUMMY(5) = SUMMY(5) + R(I)
18270      400 CONTINUE
18280      C
18290      RETURN
18300      END
18310      *

```

```

1111  HAST3.DAT.1
1111  HAST3 XXX
1111

```

