A MULTISECTORAL MODEL OF PACIFIC AND MOUNTAIN INTERSTATE TRADE FLOWS

IWR

DEPARTMENT OF THE ARMY
CORPS OF ENGINEERS

January 1973

U.S.C.E.I.

IWR Report 73-2

A MULTISECTORAL MODEL OF PACIFIC AND MOUNTAIN INTERSTATE TRADE FLOWS

Prepared for

US Army Engineers Division, South Pacific

and

Institute for Water Resources

US Army Corps of Engineers

by

H. Craig Davis

School of Community and Regional Planning

University of British Columbia

and

Everard M. Lofting

Dry Lands Research Institute

University of California at Riverside

1973 IWR Report 73-2

FOREWORD

The research which produces the present report took off from an early study of the interregional interindustry relations of the eleven Western states. The original model, completed by H. Craig Davis (1968), a co-author of the present study, is an eight-region, fifteen-sector model of interregional and interindustry trade flows. The purpose of this study is to improve upon the early model and make it more suitable for impact studies. This was accomplished by providing information for each of the eleven Western states separately and by further disaggregating the industry sector details into 21 sectors for the year 1963.

Format

The basic concept of input-output analysis and the general methodology for the report are discussed in Part A. Part A also provides a brief review of the mathematics of regional and interregional input-output models.

The eleven Western states interindustry models are presented in Part B of the report. Basic assumptions and statistical procedures for the construction of the interregional interindustry model are explained. The 1963 interregional transaction for the eleven states is shown in a 6-part (Table B-5) oversize table which is inclosed in the back pocket of the report. Part B also contains an excellent review of the various forms of interregional interindustry models. It also includes the computer programs for the study. Major sources of data for the study include the regional tables for individual states, the 1958 national table and from Census statistics.

Findings

Many studies of the economic structure of single regions have been conducted in the structure of input-output analysis. Since regional economies depend upon a much higher level of trade between other regions than does the national economy with other nations, aggregation of trade flows inhibits the understanding of the structure of the regional economy. The basic difficulty in constructing interregional tables has been the lack of data on trade between regions of the nation.

This work and its predecessor by Davis (1968), applies to the gravity formulation suggested by Leontief and Strout to the estimation of the trade flows between the eleven states by the exact solution method. The National Tables for year 1963 were utilized to distribute imports between sectors in the formulation of Moses (1955). Substantial additional work was involved in estimating final demand vectors for Personal Income, State and Federal Government and Gross Private Fixed Capital Formation - each constructed on the basis of information derived from several sources.

Assessment

The study demonstrates some very valuable input-output information obtained from indirect sources by innovative techniques. Limitations of the analysis reflect simplifying assumptions regarding the regional characteristics of the industry sectors involved, the reliance on national tables for solution of data problems, the static nature of the model, and the time value of the data. Further research is needed to test and improve the procedures including those suggested in the study. Use of econometric models such as this input-output study requires caution because of the error terms around the estimated coefficients. Yet, with the inherent errors, the interindustry, interregional model provides powerful insights into the structure of the economies of the various states. the importance of trade between regions and the capability to demonstrate the substantial diffusion of direct impacts through the interdependence of interindustry and interregional economies.

Status

Over the years the Corps of Engineers has sponsored two other input-output studies in a continuous effort to improve methods for impact analysis. One is the "Preliminary Analysis: An Analytical System for the Measurement of Economic Impact in Appalachia" by Research and Development Corporation, Washington, D. C. This study contains 83 I/O sectors for three regions in Appalachia. The other is an interregional interindustry impact model which is presented as Chapter VI in IWR Report 69-1, "Development Benefits of Water Resources Investments" by University of Washington, St. Louis, Missouri. This is a 23-sector, 3-region model with the capability of reflecting 19 regions. Readers who are interested in the subject should, of course, refer to the Harvard University study by Karen Polenski entitled: "A Multiregional Input-Output Model for the United States," a 87-sector 44-region model.

Dr. Ung-soo Kim of the Catholic University of America is currently engaged by IWR to apply interregional input-output techniques to the impact analysis of Arkansas River Multipurpose Project. His work, when completed in July 1974, will be an integrated part of a larger impact study of the water resources projects in the Arkansas River Basin.

PREFACE

This report is essentially an extension and refinement of the interregional study of the Western States undertaken by Davis.[1] In the original study the economies of Nevada, Montana, Idaho, and Wyoming were grouped and treated as a single economic region. Moreover, the earlier study was based on a minimal level of sector detail.

The present study deals with the economies of all of the eleven Western states explicitly and shows a finer level of detail for both the agriculture and mining sectors.

The outlines of the study were drawn while both authors were serving on the Economic Evaluation of Water Project at the Sanitary Engineering Research Facility, University of California, Berkeley. The computations and computer runs were carried out by Drs. Jona Bargur and M. Zaki Yacoub, now with The Technion, Israel, and Northern Michigan University, respectively.

The work was brought to completion under the auspices of the Center for Economic Studies, U.S. Army Corps of Engineers, Alexandria, Virginia. The authors express their appreciation to Mr. Nathanial Back, Mr. Robert Harrison, and Mr. James Tang for their cooperation and assistance. Similar expressions of thanks are made to Mr. Carl Quong and other members of the Mathematics and Computing Group of the Ernest O. Lawrence Berkeley Laboratory for their initiative and overall assistance in the final phases of the study. Special recognition is due Mr. J.G. Miller, LBL for his programming and related efforts in experimental video-composition techniques applied to the 231 order matrix of interstate trade relations.

H. C. D.
E. M. L.
Berkeley, California
June, 1972

[1] H.C. Davis, *Economic Evaluation of Water Part V: Multi-Regional Input-Output Techniques and Western Water Resources Development.* Sanitary Engineering Research Laboratory, College of Engineering and School of Public Health, University of California, Berkeley. Contribution No. 125, Water Resources Center, February 1968.

CONTENTS

INTRODUCTION

Regional investment decisions, if they are to be defensible in terms of regional comparative advantage and specialization, require some knowledge of the magnitude and value of commodity imports and exports. The so-called "terms of trade" and "national comparative advantage" were cornerstones of Classical economic thought, [1] but the applicability of similar considerations within national boundaries have, for the most part been neglected.

Historical accident or the "invisible hand" of the market, may have, in some instances, played a general role in determining a particular pattern of regional specialization. However, in the face of such pressing problems as environmental quality, scarce natural resources, and the depletion of resources at the regional level that can be considered to be renewable only over a planning period of future generations, it is questionable whether market forces or inertia will play a similar role in the future.

Since World War II, many important public decisions, including investment decisions, have increasingly been based on planning information generated by the newer techniques of mathematical economics, operations research, and systems analysis. [2] Transportation studies, land use planning, resource availability, model cities programs, plant site and location studies are much less subject to casual or intuitive judgment for the ultimate implementation of programs in which they may play a crucial role. This is not to say that the newer methods have gained full acceptance, [3] but it might be judged from the main stream of the literature in this area that broader acceptance may be a function of time and the increased availability of basic data whereby the effectiveness of the new techniques may be patently demonstrated.

For the specific problem of judging the appropriateness of regional resource development programs, trade flow information may be of vital importance. At the present time, however, there is virtually no reliable source from which such primary informa-

[1] See Eric Roll, *A History of Economic Thought,* Prentice-Hall, Inc. 1956.

[2] For example, the U.S. Emergency Model, discussed in H.B. Chenery and P.G. Clark, *Interindustry Economics*, John Wiley and Sons, New York, 1959.

[3] A Committee report of the National Academy of Sciences notes, "The gap between scientific knowledge of optimal methods and their application by farmers, manufacturers, and government officials is large and widening." *Alternatives in Water Management,* National Academy of Sciences, Washington, 1966.

tion can be obtained for counties, states, or the standard regional groupings of states.[1] The Bureau of the Census does provide information on certain categories of shipments for major regions of the nation,[2] but these latter groups yield information that is too gross in nature to be of use in most project area studies.

Because of the basic importance of regional [3] trade information for a variety of analytical purposes, substantial effort has been directed to the development of procedures for estimating trade flows in the absence of survey data. The most widely accepted procedure is that known as the "gravity flow" method.

Gravity flow, or trade, models have been used within a general equilibrium framework to analyze transportation investment feasibility problems in underdeveloped countries.[4, 5] They have also been used to study regional production in both Argentina [6] and Japan, [7] and are also in current use for the analysis of transportation requirements in the so-called "Northeast Corridor" of the United States.[8] A sizeable study based on gravity flow techniques has been undertaken at the Harvard Economic Research Project.[9,10] In the present volume a similar interregional model for the eleven Mountain and Pacific States has been formulated on the basis of interindustry relations and the gravity flow technique. The general methodology for the construction of the model is contained in Part A. The interstate model itself is presented in Part B together with the interstate trade impact multipliers derived from the model.

---

[1]See any Personal Income edition of the *Survey of Current Business*, U.S. Department of Commerce, Washinton, D.C.

[2]Summarized by Hugh O. Nourse, *Regional Economics*, McGraw-Hill, New York, 1968, Chapter 6.

[3]The use of the word regional here is intended to imply "small area," such as a county, or group of counties.

[4]David Kresge, "A Simulation Model for Development Planning," Harvard Transportation and Economic Development Seminar, Discussion Paper No. 32, November 1965 (unpublished).

[5]Paul O. Roberts, "The Role of Transport in Developing Countries: A development Planning Model," (paper presented December 1967, Pan American Highway Congress, Montevideo, Uruguay) (unpublished).

[6]Mario S. Brodersohn, *"A Multiregional Input-Output Analysis of the Argentine Economy,"* (Instituto Torcuato di Tella, Centro de Investigaciones Economicas, Buenos Aires, October 1965).

[7]Karen R. Polenske, "Empirical Implementation of a Multiregional Input-Output Gravity Trade Model," *Contributions to Input-Output Analysis*, ed. A.P. Carter and A. Brody, Amsterdam, North-Holland Publishing Company, 1970.

[8]This research is being supported by the U.S. Department of Transportation.

[9]Karen R. Polenske, "An Empirical Test of Interregional Input-Output Models: Estimation of 1963 Japanese Production," *American Economic Review*, Papers and Proceedings, Vol. LX, No. 2, May 1970, p. 76.

[10]Karen R. Polenske, "A Multiregional Input-Output Model for the United States," U.S. Department of Commerce, Economic Development Administration, Report No. 21, October 1970.

PART A: METHODOLOGY

THE REGIONAL INTERINDUSTRY ECONOMIC TRANSACTIONS MODEL

1. Partial vs. General Equilibrium Analysis

The economic activities within a region may at times be grouped into a number of industries or sectors in order to facilitate economic analysis. These sectors will not generally be independent of each other as there will usually be flows of goods and services between the sectors. Judgements as to the nature and extent of these flows will govern the choice between a partial or general equilibrium approach to the analysis of many regional economic issues.

Suppose the analyst is interested in the effects upon a particular sector of a change in the market demand for its production. It may be that the resulting change in the sector's production will have repercussions on other sectors in the regional economy. Moreover, these repercussions may in turn affect, or feed back upon, the sector in question. The basic assumption of partial equilibrium analysis is that such feedback to the sector under study is sufficiently small to be ignored.

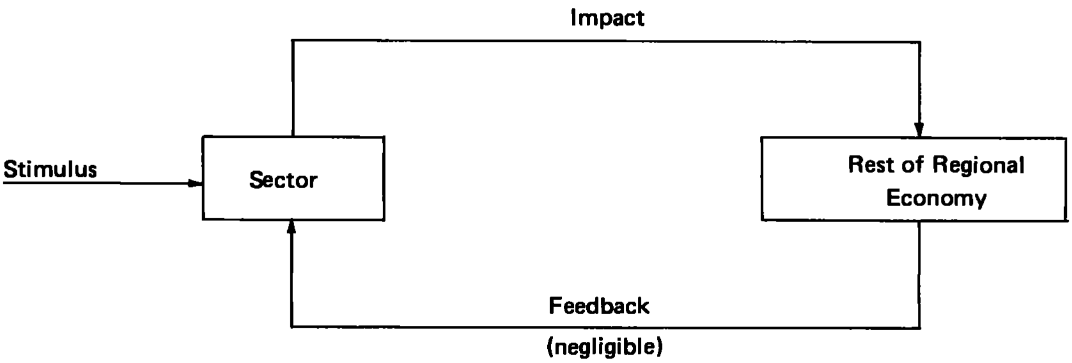


Figure A-1. Conditions for Partial Equilibrium Analysis

On the other hand general equilibrium analysis is appropriate if feedback upon the sector is significant or if the analyst is interested in the effects (impact) of the initial stimulus upon other sectors of the regional economy. Under general equilibrium analysis, relationships between sectors of the economy are explicitly taken into account and the entirety of the economy is brought under examination.

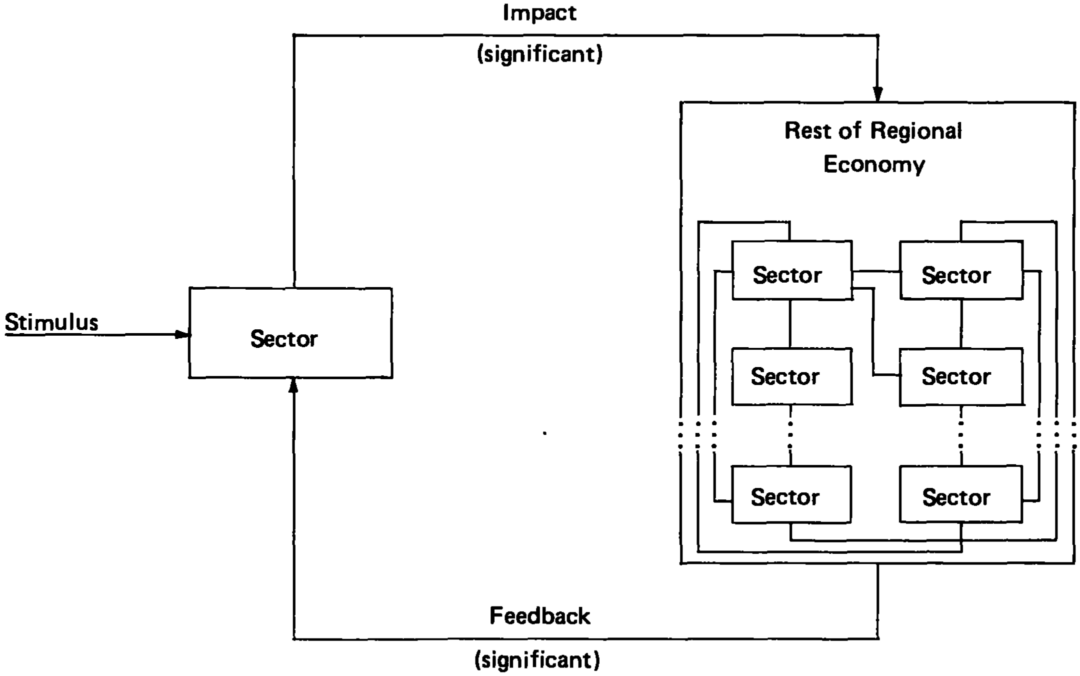


Figure A-2. Conditions for General Equilibrium Analysis

In a pure equilibrium model each variable in the model depends only upon other variables within the model and hence all variables are endogenous. That is, values for all variables are yielded by the model and none must be determined exogenously.

In the 1930's Wassily Leontief formulated a mathematical description of the U.S. economy constructed from data regarding economic transactions between industries. The effort stands as the first operational economic model to embody the essense of general equilibrium. The model is not strictly a pure equilibrium model since, as we shall see, some of the key variables of the model are exogenous, i.e., values for these variables must be determined outside the model itself and then "plugged in."

Leontief's model came to be labelled "input-output" as it records each transaction between firms in double entry fashion as both a sale of output and a purchase of input. The model represents a general equilibrium approach to the study of an economy and focuses upon the interdependencies between the various economic sectors. Although initially applied on a national level, the model has been used in a proliferation of regional applications in recent years.

2. The Regional Input-Output Model

The basic I-O model may be explained in terms of its three associated tables:

a. Table of Interindustry Transactions
b. Table of Direct Requirements
c. Table of Direct Plus Indirect Requirements

a. Table of Interindustry Transactions

As a first step in the construction of the Transactions Table the regional economy is segmented into a number of sectors (or industries). Just what sectors the analyst chooses to represent the economy depends upon 1) the nature of the regional economy with which he is working, 2) the nature of the problem in which he is interested, and 3) the resources at his disposal. The number of sectors of the typical regional I-O model will be in the neighborhood of 30. Each of these sectors is generally defined in terms of the Standard Industrial Classification (SIC) codes.[1] Every type of economic activity may be represented by an SIC code number at the 2-digit, 3-digit or 4-digit level. For example, a fertilizer manufacturer would be classified at the 4-digit level as belonging to the category: 2871 Fertilizers. If we were to aggregate our SIC classification to the 3-digit level the firm would be classified under the heading of 287 Agricultural Chemicals. If we were to collapse our categories to the 2-digit level our firm would fall into the 28 Chemicals and Allied Products classification. There are currently 79 two-digit classifications, which comprise 10 divisions. For example, SIC 28 is contained within Division D-Manufacturing.

However the factors are defined, it is essential that taken together they encompass the entirety of economic activity within the region. For illustrative purposes, the following interindustry Transactions Table contains only 3 sectors or industries.

Table A-1.
TRANSACTIONS TABLE ($1000)

		Purchasers			Final Demand	Gross Output
		Agriculture	Manufacturing	Services		
Sellers	Agriculture	10	5	5	50	70
	Manufacturing	20	30	25	25	100
	Services	5	10	10	55	80
Imports		5	15	5		
Value Added		30	40	35		
Gross Outlay		70	100	80		

Since each sector both buys from and sells to other sectors within the economy for further processing, each of the three sectors is listed both at the left of the table as a seller and at the top of the table as a purchaser. The 3 x 3 matrix formed by these sectors is referred to as the "processing" matrix.

The final demand sectors represent all sales that are made not for further processing within the region but for final consumption by Households, Government, Investment, and Exports. If a farmer sells milk to a restaurant, the transaction is from Agriculture to Services; if the farmer sells milk to a household the transaction

[1] Technical committee on Industrial Classification, Office of Statistical Standards, *Standard Industrial Classification Manual*, Washington D.C., 1957.

is from Agriculture to Final Demand (Households). If a steel mill sells to a metal fabricating plant within the region, it is a transaction between Manufacturing and Manufacturing; if the mill sells to a metal fabricating plant outside the region, it is a transaction from Manufacturing to Final Demand (Export). Thus each sector's sales are recorded as satisfying either intermediate (processing) or final demand. Gross Output (total sales) of each sector is the sum of intermediate and final sales. Reading along any of the first three rows, we can see how each sector distributed its output over the period. Manufacturing, for example, sold \$20,000 of its \$100,000 total output to Agriculture, \$30,000 to Manufacturing, \$25,000 to Services, and the remaining \$25,000 to Final Demand.

Reading down any of the first three columns we may see how the particular sector purchased input. For example, Manufacturing purchased \$5,000 from Agriculture, \$30,000 from Manufacturing, \$10,000 from Services, \$15,000 from Imports (goods and services produced outside the region), and \$40,000 from the Value Added sector (roughly wages and salaries, rents, interest, depreciation, dividends, and profit). Gross Outlay (total purchases) must equal Gross Output (total sales) as profits are considered to be the remuneration to management and thus serve as the balancing item. That is, total sales revenue is equal to total cost plus profit. This point is emphasized below by the diagram representing dollar flows into and out of a sector.

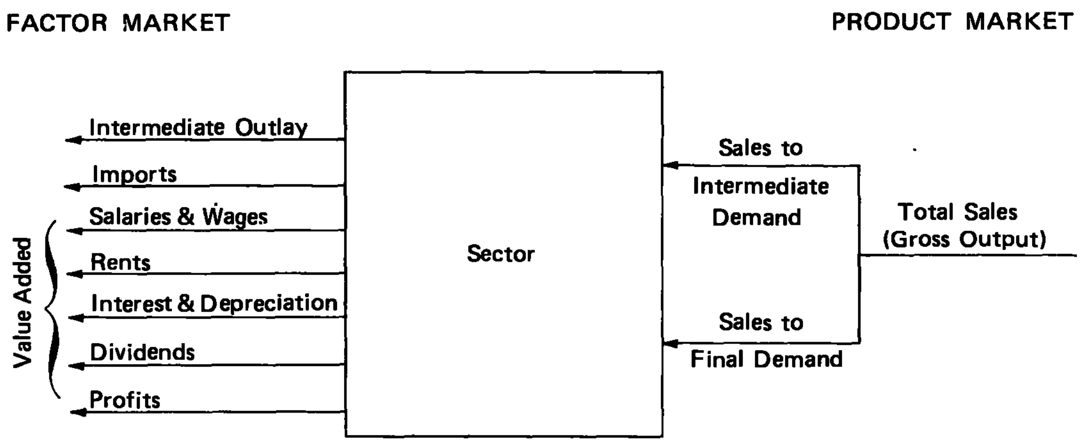


Figure A-3. Sectoral Monetary Flows

The dollar flows are normally those that have been recorded over a period of one year and data pertaining to the flows are gathered by personal and/or mail interviews with each of the firms or a sampling of the firms within each of the sectors.

b. Table of Direct Requirements

The Table of Direct Requirements or coefficients is formed by dividing the entries in each column of the processing matrix by their respective column total (Gross Outlay). The Direct Coefficients Table of our illustrative I-O model is Table A-2.

Table A-2.
TABLE OF DIRECT REQUIREMENTS PER DOLLAR OF GROSS OUTLAY

	Agriculture	Manufacturing	Services
Agriculture	.14	.05	.06
Manufacturing	.29	.30	.31
Services	.07	.10	.12

Table A-2 reveals that for the average dollar spent by, say, Agriculture during the period, 14¢ were spent on Agricultural inputs, 29¢ upon inputs from Manufacturing, and 7¢ on Services. Under the assumption that these coefficients remain fixed, we can forecast the effect upon the regional economy resulting from an increase in Final Demand.

To illustrate, let us assume that the export demand for the output of our regional manufacturing sector increases by $10,000. The manufacturing sector will increase its output by $10,000 to meet this rise in final demand and to do so will have to make the following purchases:

Manufacturing: $10,000	
Agriculture	$10,000 x .05 = $ 500
Manufacturing	$10,000 x .30 = $3,000
Services	$10,000 x .10 = $1,000

However, in order to produce this supporting output, each sector will require the following inputs:

	Agriculture: $500	Manufacturing: $3,000	Services: $1,000	Total
Agriculture	500 x .14 = $ 70	3000 x .05 = $150	1000 x .06 = $ 60	$ 280
Manufacturing	500 x .29 = $145	3000 x .30 = $900	1000 x .31 = $310	$1,355
Services	500 x .07 = $ 35	3000 x .10 = $300	1000 x .12 = $120	$ 455

These requirements will set off a third round of spending:

	Agriculture: $280	Manufacturing: $1355	Services: $455	Total
Agriculture	280 x .14 = $39.20	1355 x .05 = $ 67.75	455 x .06 = $ 27.30	$124.25
Manufacturing	280 x .29 = $81.20	1355 x .30 = $406.50	455 x .31 = $141.05	$628.75
Services	280 x .07 = $19.60	1355 x .10 = $135.50	455 x .12 = $ 54.60	$209.70

These rounds of spending will continue with each round becoming weaker in its effects. The accumulated increases in total sales of each sector resulting from the stimulus to the Manufacturing sector of $10,000 in export demand can be computed from the increase in sales of each round.

While such series of calculations are helpful in understanding the effects that reverberate throughout the regional economy from the initial stimulus, fortunately they are not necessary to determine the ultimate effects. The final changes in total sales (Gross Output) of each sector can be read directly from the third table of the I-O model, the table of direct plus indirect requirements.

c. Table of Total (Direct Plus Indirect) Requirements

Generally, with the aid of a computer, our third table may be constructed through inversion of a matrix, associated, as we shall see in the following section, with our second table. For our illustrative model the table is Table A-3.

Table A-3.
TABLE OF DIRECT PLUS INDIRECT REQUIREMENTS PER DOLLAR OF DELIVERY TO FINAL DEMANDS*

	Agriculture	Manufacturing	Services
Agriculture	1.2117	0.5677	0.1637
Manufacturing	0.1042	1.5542	0.1861
Services	0.1237	0.5956	1.2210

*Transposed

Table A-3 tells us that if there is a $1 increase in the final demand for Agriculture, the total output of Agriculture will, after all the interdependent transactions have worked themselves out, increase by $1.21. Manufacturing and Services in this case will rise $0.57 and $0.16 respectively. We can now easily read from the table the effects of a $10,000 increase in exports of Manufacturing. Total sales of Agriculture will rise $1,000 ($10,000 x 0.10), Manufacturing sales will increase $15,500 ($10,000 x 1.55) and the output of the Services sector will expand by $1,900 ($10,000 x 0.19).

APPENDIX
The Regional Interindustry Model in Mathematical Summary

As previously shown, the I-O model records each sale in the economy as "intermediate" or "final." Total sales or output of any sector of an n-sector model can thus be expressed as

$$\sum_{j=1}^{n} x_{ij} + y_i = x_i \qquad (i=1,\ldots,n) \qquad (1)$$

where x_{ij} = the value of the output of sector i purchased by sector j,
y_i = the final demand for the output of sector i, and
x_i = the value of the total output of sector i.

The regional economy is thus conceptualized by n linear equations, each equation expressing the transactions of a particular sector with the processing sectors, and with final demands. Equation 1 represents the major portion of our first table, the Transactions Table. As such, it is merely a set of balance equations or accounting identities. To complete the mathematical description of the Transactions Table we write

$$\sum_{i=1}^{n} x_{ij} + p_j = x_j \qquad (j=1,\ldots,n) \qquad (2)$$

where p_j = final payments (purchases of imports and other factors) by sector j.
x_j = total outlay (purchases) of sector j.

$$x_i = x_j \qquad \text{for all } i=j \qquad (3)$$

The second table of the I-O model, the Table of Direct Requirements can be expressed as the matrix (a_{ij}) where

$$a_{ij} = \frac{x_{ij}}{x_j} \qquad (i,\ j=1,\ldots,n) \qquad (4)$$

Substituting (4) into (1) yields

$$x_i = \sum_{j=1}^{n} a_{ij}x_j + y_i \qquad (i=1,\ldots,n) \qquad (5)$$

which may be expressed more compactly as

$$X = AX + Y \qquad (6)$$

where

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} \ldots & a_{1n} \\ a_{21} & a_{22} \ldots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} \ldots & a_{nn} \end{pmatrix}, \text{ and } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{7}$$

It may now be shown that gross output minus intermediate use equals the net output of the system or final use.

$$X - AX = (I\text{-}A)X = Y \tag{8}$$

where I is an n x n identity matrix. Given the exogenous or final demands on the economy, it is possible to solve the system for total outputs,

$$X = (I - A)^{-1}\ Y \tag{9}$$

where $(I - A)^{-1}_T$ is the third table of the I-O model, the Table of Direct Plus Indirect Requirements. The matrix is customarily written in transposed form to facilitate reading of the desired information along the rows rather than down the columns.

THE INTERREGIONAL INTERINDUSTRY ECONOMIC TRANSACTIONS MODEL

1. The Basic Interregional Model

Interregional input-output analysis differs from regional studies in that the former attempts to incorporate a spatial component into the analysis. The regional I-O model classifies economic activity into sectors or industries; interregional theory likewise breaks down activities into industries but also into geographic regions. The interregional model focuses upon the economic relationships not only within each region, but among the regions as well.

If, say, a lumber mill in Region #2 obtains its inputs from the forestry sector of region #1, a direct economic "tie" or interdependence exists between the two economic activities and thus between the two regions. If a furniture manufacturer in region #3 purchases output from the lumber mill in region #2, in addition to the direct link formed between regions #2 and #3, an indirect tie is established between regions #1 and #3. Interregional I-O theory is designed to reveal both the direct and indirect interdependencies among regions.

For purposes of illustration a two region three sector interregional transaction table might be constructed as shown below:

Table A-4. Interregional Input-Output Format.

		Region #1					Region #2				
		Ag.	Man.	Serv.	F.D.		Ag.	Man.	Serv.	F.D.	G.O.
Region #1	Ag.	$	$	$	$	Ag.	$	$	$	$	$
	Man.	$	$	$	$	Man.	$	$	$	$	$
	Serv.	$	$	$	$	Serv.	$	$	$	$	$
Region #2	Ag.	$	$	$	$	Ag.	$	$	$	$	$
	Man.	$	$	$	$	Man.	$	$	$	$	$
	Serv.	$	$	$	$	Serv.	$	$	$	$	$
Final Payments (Inputs & V.A.)		$	$	$	$		$	$	$	$	$
G.O.		$	$	$	$		$	$	$	$	$

Reading across, say, the Manufacturing row in region #1, we see how the sector distributes its output among the intermediate and final demand sectors of region #1 as well as those of region #2. The final demand sectors in region #1 are again, Household Consumption, Investment, Government Spending, and Exports. Exports in this case, however, exclude exports to region #2 to avoid double counting. Reading down the column of, say, the Services sector of region #2, we see the inputs this sector purchases from the processing sectors of both regions as well as from the final payments sector. Table A-4 thus presents the flows of goods and services, a) between sectors of region #1, b) from region #1 to region #2, c) from region #2 to region #1, and d) between sectors of region #2. The sub-matrices forming the main diagonal of such a table constitute the domestic processing matrices of the regions represented.

APPENDIX

The Model in Mathematical Summary

In an m-region, n-sector area, the Transactions Table of the interregional model described in the preceding section may be represented mathematically as

$$ {}_{r}x_{i} = \sum_{s=1}^{m} \sum_{j=1}^{n} {}_{rs}x_{ij} + {}_{r}y_{i} \qquad \begin{matrix}(i=1,\ldots,n)\\(r=1,\ldots,m)\end{matrix} \qquad (1) $$

where

${}_{r}x_{i}$ = the gross output of sector i in region r,
${}_{rs}x_{ij}$ = the output of sector i in region r sold to sector j in region s, and
${}_{r}y_{i}$ = the final demand for the output of sector i in region r.

The second table of the interregional model, the Table of Direct Requirements, can be described as

$$ {}_{rs}a_{ij} = \frac{{}_{rs}x_{ij}}{{}_{s}x_{j}} \qquad \begin{matrix}(i, j=1,\ldots,n)\\(r, s=1,\ldots,m)\end{matrix} \qquad (2) $$

Substitution of (2) into (1) yields

$$ {}_{r}x_{i} = \sum_{s=1}^{m} \sum_{j=1}^{n} {}_{rs}a_{ij}\, {}_{s}x_{j} + {}_{r}y_{i} \qquad \begin{matrix}(i=1,\ldots,n)\\(r=1,\ldots,m)\end{matrix} \qquad (3) $$

Letting $A^* = ({}_{rs}a_{ij})$, the interregional system is now treated in the same manner as is the regional system. The general solution is found through matrix inversion as

$$ {}_{r}X = (I-A^*)^{-1}\, {}_{r}Y \qquad (r=1,\ldots,m) \qquad (4) $$

Through employment of the $(I-A^*)^{-1}$ matrix or its transpose, $(I-A^*)_T^{-1}$, the Table of Direct Plus Indirect Requirements, one can determine under the assumption of unchanged coefficients of regional production and interregional trade the resulting effects of an increase in the final demand for the output of any sector in any region on the outputs (sales) of all sectors in all regions.

2. The Interregional Trade Flows: The Gravity Distribution Technique

a. Leontief-Strout Model

The Leontief-Strout gravity flow model[1] involving interregional commodity flows may be examined in terms of probabilities. As a first step let us assume the existence of n commodities traded among m regions. For any one of the n commodities let

X_{go} = the production of the commodity in region g
X_{oh} = the consumption of the commodity in region h
X_{oo} = the aggregate production of the commodity = the aggregate consumption of the commodity.

The probability that a unit of the commodity in the national market is produced by region g is the ratio of regional-to-national production, X_{go}/X_{oo}. Similarly, the likelihood that a unit is consumed in region h is the ratio of regional-to-national consumption, X_{oh}/X_{oo}.

Assuming momentarily that there exist no barriers whatever to commodity transport, the expectation P_{gh} that a unit that is produced in g is consumed in any other region h is merely the product of the ratios discussed above:

$$P_{gh} = \frac{X_{go}}{X_{oo}} \cdot \frac{X_{oh}}{X_{oo}} \qquad \begin{matrix}(g \neq h)\\(g,h = 1, \ldots, m)\end{matrix} \qquad (1)$$

Thus the total amount of the commodity shipped from g to h, X_{gh}, can be measured as the total production multiplied by the probability of any one unit being shipped from g to h:

$$X_{gh} = X_{oo}P_{gh} = \frac{X_{go}\,X_{oh}}{X_{oo}} \qquad \begin{matrix}(g \neq h)\\(g,h = 1, \ldots, m)\end{matrix} \qquad (2)$$

Introducing transport variables into the equation,

$$X_{gh} = \frac{X_{go}\,X_{oh}}{X_{oo}}\,d_{gh}\,\delta_{gh} \qquad \begin{matrix}(g \neq h)\\(g,h = 1, \ldots, m)\end{matrix} \qquad (3)$$

The variable d_{gh} is a measure of the inverse of the transport cost of moving a unit from g to h. δ_{gh} is a parameter designed to limit the number of variables in the system. If, for any of a variety of economic reasons, no export from g to h is expected, δ_{gh} = O; otherwise $\delta_{gh} = 1$.

---

[1] Wassily Leontief and Alan Strout. "Multiregional Input-Output Analysis," in Tibor Barna, ed., *Structural Interdependence and Economic Development*, New York: St. Martin's Press, 1963. pp. 119-51.

APPENDIX

The Gravity Distribution Technique in Mathematical Summary

Introducing into the last equation of the preceding section two additional parameters, C and K, which characterize in a summary way the relative position of region g vis-a-vis all other regions as a supplier, and of region h as a user, of a particular commodity

$$X_{gh} = Q_{gh} \frac{X_{go} X_{oh}}{X_{oo}} \qquad \begin{matrix} (g \neq h) \\ (g,h = 1, \ldots, m) \end{matrix} \qquad (1)$$

where

$$Q_{gh} = (C_g + K_h)\, d_{gh} \delta_{gh} . \qquad (2)$$

At this point Leontief and Strout develop four methods of estimating interregional trade flows: 1) Point Estimate, 2) Least Squares, 3) Simple Solution, and 4) Exact Solution. Methods (1) and (2) require knowledge of actual base-year flows and are of no interest to the present study since such information in unavailable for trade flows among the Western States. Of the two remaining procedures, (4) is to be preferred on the basis that while superiority of predictive power between (3) and (4) as to interregional shipments is not clearly established by the authors' empirical tests, method (4), unlike (3),[1] permits no discrepancy between actual and estimated total imports and exports. Henceforward we are concerned in the discussion only with method (4), in which base-year observations of actual interregional flows are not necessary and C and K can be determined indirectly.

In the national economy, internal trade can be summarized by Equations (3a) and (3b).

$$X_{go} = \sum_{h=1}^{m} X_{gh} \qquad (g = 1, \ldots, m) . \qquad (3a)$$

That is to say, the supply of a particular good in region g is equal to the amount of the good shipped from g to all other regions, including internal shipments X_{gg}.

$$X_{oh} = \sum_{g=1}^{m} X_{gh} \qquad (h = 1, \ldots, m) \qquad (3b)$$

[1] A method for rendering the Simple Solution internally consistent has been put forward by Jinkichi Tsukui and Karen Polenske, *Consistent Estimate for Leontief Gravity Model of Interregional Trade*, Harvard Economic Research Project, Cambridge, Massachusetts, April 2, 1964 (revised January 10, 1965). While the authors found a favorable comparison between the empirical results of their 'modified' model and those derived from application of the Leontief-Strout Point Estimate technique, the predictive power of the 'modified' Simple Solution as compared with that of the Exact Solution method was not explored.

The consumption of any commodity in region h is equal to the amount of the commodity shipped to h from all other regions, including internal consumption X_{hh}.

Substituting Equation (1) into (3a) and (3b) we obtain

$$X_{go} = X_{go} \frac{\sum_{r=1}^{m} X_{or} Q_{gr}}{X_{oo}} + X_{gg} \qquad \begin{matrix} (Q_{gg} = O) \\ (g = 1, \ldots, m) \end{matrix} \qquad (4a)$$

$$X_{oh} = X_{oh} \frac{\sum_{r=1}^{m} X_{ro} \; Q_{rh}}{X_{oo}} + X_{hh} \qquad \begin{matrix} (Q_{gg} = O) \\ (h = 1, \ldots, m) \end{matrix} \qquad (4b)$$

Rearranging terms and substituting Equation (2) into Equations (4a) and (4b) respectively,

$$X_{go} \sum_{r=1}^{m} \left[X_{or} \, (C_g + K_r) \, d_{gr} \delta_{gr} \right] = (X_{go} - X_{gg}) \, X_{oo} \qquad \begin{matrix} (\delta_{gg} = O) \\ (g = 1, \ldots, m) \end{matrix} \qquad (5a)$$

$$X_{oh} \sum_{r=1}^{m} \left[X_{ro} \, (C_r + K_h) \, d_{rh} \, \delta_{rh} \right] = (X_{oh} - X_{hh}) \, X_{oo} \qquad \begin{matrix} (\delta_{hh} = O) \\ (h = 1, \ldots, m) \end{matrix} \qquad (5b)$$

Together Equations (5a) and (5b) form a set of 2m equations with 2m unknowns, C_g and K_h. Since the observable base-year values for production and consumption will satisfy the overall equality between the aggregates,

$$\sum_g \sum_h X_{gh} = \sum_g X_{go} = \sum_h X_{oh} = X_{oo} \qquad (g,h = 1, \ldots, m). \qquad (6)$$

One of the 2 m Equations (5a) or (5b) is redundant and is to be eliminated. Further, it can be seen from Equation (1) that if any set of values of the variables C_g and K_h, say C_g* and K_h*, satisfy the equations, the values C_g*-c and K_h*+c (c=a constant) will also satisfy set (1). This is to say that solutions to only 2m-1 of the unknowns are needed to produce a unique set of interregional flows. Thus not only is one equation dropped, but one of the 2 m unknowns is arbitrarily fixed. The first equation (h = 1) is chosen to be eliminated and K_1 is arbitrarily set equal to zero.

A few steps remain to ease computational procedures. X_{go} C_g and X_{oh} K_h are treated as the unknowns rather than C_g and K_h, and are respectively factored from Equations (5a) and (5b), while all observable values are expressed in terms of X_{oo} (thus effectively setting X_{oo} = 1). The resulting equations are

$$X_{go}\, C_g \sum_r X_{or}\, d_{gr}\, \delta_{gr} + X_{go} \sum_r K_r\, X_{or}\, d_{gr}\, \delta_{gr} = X_{go} - X_{gg}$$

$$(r,g = 1, \ldots, m) \qquad (7a)$$

$$X_{oh}\, K_h \sum_r X_{ro}\, d_{rh}\, \delta_{rh} + X_{oh} \sum_r C_r\, X_{ro}\, d_{rh}\, \delta_{rh} = X_{oh} - X_{hh}$$

$$\begin{matrix} (r = 1, \ldots, m) \\ (h = 2, \ldots, m) \end{matrix} \qquad (7b)$$

Dividing Equations (7a) and (7b) by X_{go} and X_{oh}, respectively, we obtain

$$C_g \sum_r X_{or}\, d_{gr}\, \delta_{gr} + \sum_r K_r\, X_{or}\, d_{gr}\, \delta_{gr} = 1 - \frac{X_{gg}}{X_{go}}$$

$$(r,g = 1, \ldots, m) \qquad (8a)$$

$$K_h \sum_r X_{ro}\, d_{rh}\, \delta_{rh} + \sum_r C_r\, X_{ro}\, d_{rh}\, \delta_{rh} = 1 - \frac{X_{hh}}{X_{oh}}$$

$$\begin{matrix} (r = 1, \ldots, m) \\ (h = 2, \ldots, m) \end{matrix} \qquad (8b)$$

Equation sets (8a) and (8b) may now be expressed in matrix form[1] and solved via inversion and the resulting values for C_g and K_h inserted in Equation (1) to determine the interregional trade flows.

$$\left[\begin{array}{cccc|ccccc}
\sum_r \frac{X_{ro}\, d_{r2}\, \delta_{r2}}{X_{o2}} & 0 & \cdots & 0 & d_{12}\,\delta_{12} & 0 & d_{32}\,\delta_{32} & \cdots & d_{m2}\,\delta_{m2} \\
0 & \sum_r \frac{X_{ro}\, d_{r3}\, \delta_{r3}}{X_{o3}} & \cdots & 0 & d_{13}\,\delta_{13} & d_{23}\,\delta_{23} & 0 & \cdots & d_{m3}\,\delta_{m3} \\
0 & 0 & \cdots & \sum_r \frac{X_{ro}\, d_{rm}\, \delta_{rm}}{X_{om}} & d_{1m}\,\delta_{1m} & d_{2m}\,\delta_{2m} & d_{3m}\,\delta_{3m} & \cdots & 0 \\
\hline
d_{12}\,\delta_{12} & d_{13}\,\delta_{13} & \cdots & d_{1m}\,\delta_{1m} & \sum_r \frac{X_{or}\, d_{1r}\, \delta_{1r}}{X_{1o}} & 0 & 0 & \cdots & 0 \\
0 & d_{23}\,\delta_{23} & \cdots & d_{2m}\,\delta_{2m} & 0 & \sum_r \frac{X_{or}\, d_{2r}\, \delta_{2r}}{X_{2o}} & 0 & \cdots & 0 \\
d_{32}\,\delta_{32} & 0 & \cdots & d_{3m}\,\delta_{3m} & 0 & 0 & \sum_r \frac{X_{or}\, d_{3r}\, \delta_{3r}}{X_{3o}} & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
d_{m2}\,\delta_{m2} & d_{m3}\,\delta_{m3} & \cdots & 0 & 0 & 0 & 0 & \cdots & \sum_r \frac{X_{or}\, d_{mr}\, \delta_{mr}}{X_{mo}}
\end{array}\right]
\left[\begin{array}{c}
X_{o2}\, K_2 \\ X_{o3}\, K_3 \\ \vdots \\ X_{om}\, K_m \\ X_{1o}\, C_1 \\ X_{2o}\, C_2 \\ X_{3o}\, C_3 \\ \vdots \\ X_{mo}\, C_m
\end{array}\right]
=
\left[\begin{array}{c}
1 - \frac{X_{22}}{X_{o2}} \\ 1 - \frac{X_{33}}{X_{o3}} \\ \vdots \\ 1 - \frac{X_{mm}}{X_{om}} \\ 1 - \frac{X_{11}}{X_{1o}} \\ 1 - \frac{X_{22}}{X_{2o}} \\ 1 - \frac{X_{33}}{X_{3o}} \\ \vdots \\ 1 - \frac{X_{mm}}{X_{mo}}
\end{array}\right] \qquad (9)$$

[1]Leontief and Strout, *op. cit.*, p. 131. The zeros appear in the NE and SW quadrants of the matrix due to the assumption $\delta_{gg} = O$ which rules out consideration of intraregional shipments.

PART B:

THE ELEVEN WESTERN STATES INTERINDUSTRY STUDY FOR 1963

INTRODUCTION

The model developed and presented in this section stems directly from an initial interregional interindustry study of the Western States[1] which was based on eight distinct regions comprised of the three Pacific States, four of the eastern and southerly Mountain States, and the four remaining states grouped as a single geographic region. This first model displayed each regional economy at the fifteen sector level of disaggregation. Interest was manifested by state and federal agencies cooperating on the Pacific Southwest land and water planning studies in a full eleven state model showing a further disaggregation of the agricultural and mining sectors. Thus the original study has been expanded in the present report to fulfill these specifications.

The attention of these cooperating agencies mainly appeared to be centered on an assessment of sectoral trade impacts among the eleven western states and on the output levels of the individual states which comprise portions of the Water Resources Council Planning Regions.[2] The analysis has therefore been directed, almost exclusively to this end in the expanded model.

In settling upon the level of disaggregation for the new model, it was realized that with "m" regions and "n" sectors (industries) an ultimate inversion of a matrix order m x n would have to be carried out. With eleven regions and twenty-one sectors it was felt that the resulting two hundred and thirty-one order matrix would be of feasible dimensions given available inversion routines and limitations on computer time.

A review of the various forms of interregional input-output models that have been developed is presented in Appendix I. These include the Leontief "balanced regional"

[1]Davis, H.C., *Economic Evaluation of Water Part V: Multiregional Input-Output Techniques and Western Water Resources Development*, Water Resources Center Contribution No. 125, University of California, Berkeley, 1968.

[2]The data developed for each of the state economies permitted the construction of the Pacific Southwest River Basin model used in the *Analytical Summary Report* of the so-called "Framework" or Type I studies.

model,[1] the Isard model,[2] and models by Chenery,[3] Moses,[4] and Wonnacott.[5] The latter three formulations are generally termed "modified" Isard models in that they incorporate some simplifying assumptions necessary in order to implement the models empirically. As stressed by Davis[6]

> (the) common assumption states that imports of commodity i into region j will be distributed throughout the sectors of region j in the same manner as is the output of commodity i produced in region j. Thus if the electrical machinery industry in the East absorbs 10 percent of Eastern steel output, the same electrical machinery industry is assumed to absorb 10 percent of the imports of Western steel.

Given the paucity of intrastate trade flow data, Davis was forced to adopt the above assumption, with the additional constraint that all trade flows would necessarily be expressed in net terms. This further assumption was necessitated by the formats of the available individual state models adopted by the study.

Since analysis in the expanded model has been limited to sectoral trade impacts among regions, all tables shown in this section are presented without regard to final demand and value added quadrants. Tables shown pertain only to the productive sectors of the regional economies and associated imports and exports of these sectors.

In view of the foregoing, some summary data on the individual states and the 11 state region as a whole are presented to supplement the more detailed information provided in the Transactions Table of the interstate model.

Gross Regional Product for the eleven-state region constituting the Pacific and Mountain west is 103.7 billion dollars (1963) while the corresponding Gross National Product for 1963 was 583.9 billion dollars.[7] Thus the region accounted for 17.8 percent of the U.S. product.

Personal Income (1963) for each of the eleven Western States was as follows:[8]

---

[1] Wassily Leontief, *Studies in the Structure of the American Economy,* Oxford University Press, N.Y. 1951.

[2] Walter Isard, "Interregional and Regional Input-Output Analysis: A Model of Space-Economy," *Review of Economics and Statistics,* 33:318-39, November 1951.

[3] Hollis Chenery, "Interregional and International Input-Output Analysis," in T. Barna, ed. *The Structural Interdependence of the Economy,* Wiley, New York, 1956.

[4] Leon Moses, "A General Equilibrium Model of Production, Interregional Trade and Location of Industry," *Review of Economics and Statistics,* 42:373-97, Nov. 1960.

[5] R. J. Wonnacott, *Canadian-American Dependency, An Interindustry Analysis of Production and Prices.* North Holland Publishing Co., Amsterdam. 1961.

[6] Davis, H.C. *op. cit.*, p. 82.

[7] *Statistical Abstract of the United States,* U.S. Department of Commerce, Bureau of the Census, 1965.

[8] *Ibid.* p. 334.

Table B-1. Personal Incomes of Western States, 1963.

State	Personal Income — 1963 Millions of Dollars
Utah	2,083
Montana	1,553
Oregon	4,568
California	52,317
New Mexico	1,953
Nevada	1,246
Idaho	1,366
Colorado	4,831
Washington	7,575
Wyoming	834
Arizona	3,340

Gross State Product for the states of the region, in rank order, has been estimated as follows:

Table B-2. Gross State Products of Western States, 1963.

Rank	State	GSP 1963 Billions of Dollars
1	California	62.3
2	Washington	9.1
3	Colorado	7.8
4	Oregon	6.0
5	Arizona	5.7
6	New Mexico	3.3
7	Utah	2.5
8	Montana	1.9
9	Idaho	1.6
10	Nevada	1.5
11	Wyoming	1.1
	Total	103.7

The State of California accounts for some 60 percent of Gross Regional Product—an amount that is approximately 20 billion dollars larger than the total of the other ten states in the region.

SECTOR CLASSIFICATION

The sectors represented for each region of the original Western States study were structured to emphasize the water intensive industries.[1] These sectors, for manufacturing categories, were Food and Kindred Products, Chemicals and

[1]H.C. Davis, *op. cit.* p. 92-3.

Allied Products, Petroleum and Coal Products, Paper and Allied Products, and Primary Metal Industries. In extending the study to the full eleven states, the original structuring for the Manufacturing, Trade, and Service sectors was retained and the Agricultural and Mining sectors were each disaggregated to show the level of detail given by the 1958 National Interindustry Relations Study.[1] The 21 sectors in each of the state tables were structured as shown below in Table B-3.

Table B-3. Sector Classification.

SECTOR NUMBER AND TITLE	RELATED 1958 NATIONAL INPUT-OUTPUT SECTORS	RELATED 1957 SIC CODES
1. Livestock and Livestock Products	1	013, Pt. 014, Pt. 02
2. Other Agricultural Products	2	011, 012, pt. 014
3. Forestry and Fishery Products	3	074, 081, 082, 084, 086, 091
4. Agricultural, Forestry and Fishery Services	4 4	071, 072, 073, 085, 098, pt, 0729
5. Nonferrous Metal Ores Mining	6	102, 103, 104, 105, 108, 109
6. Crude Petroleum and Natural Gas	8	1311, 1321
7. Stone, Clay, and Quarrying	9	141, 142, 144, 145, 148, 149
8. All other mining	5, 7, 10	1011, 106, 11, 12 (excl. 138), 147
9. Construction	11, 12	15-17, 138, pt. 6561
10. Food and Kindred Products	14-15	20-21
11. Textiles, Leather, and Apparel	16-19, 33-34	22-23, 31, 3992
12. Paper and Allied Products	24-25	26
13. Chemicals and Allied Products	27-30	28 (excl. pt. 2819)
14. Petroleum Refining and Allied Industries	31	29
15. Primary Metals	37-38	33, pt. 2819
16. Machinery and Transport Equipment	43-61	35-37
17. Utilities	68	49
18. Wholesale and Retail Trade	69	50, 52-59, pt. 7399
19. Finance, Real Estate, and Services	70-77	60-67 (excl. pt. 6561), 70, 72-73, 75, 76, 78-79 (excl. pt. 7399), 80, 81, 82, 84, 86, 89.
20. All Other Manufacturing	13, 20-32, 26, 32, 35-36, 39-42, 62-64	19, 24;25, 27, 30, 32, 24, 38-39 (excl. 3992) 40-42, 44-48
21. Not Elsewhere Classified	65-67, 78-79, 81-87	40-42, 44-48

[1]*Survey of Current Business*, September, 1965.

CONSTRUCTION OF THE STATE INTERINDUSTRY MODELS

This section reviews the procedures for developing the individual state interindustry models. These tasks consist of estimation of the following general elements: a) Gross Output, b) Interindustry Flows, c) Final Demands, d) Payments Sector, e) Net Trade Flows. Models for six individual states were available at the outset of the original (Davis) 8-region, 15 sector model: Arizona (1958), California (1963), New Mexico (1960), Oregon (1963), Utah (1963), and Washington (1963). Separate interindustry models (1963) were constructed for the state of Colorado and the remaining four state block of Nevada, Idaho, Montana and Wyoming.[1]

1. Gross Outputs

The gross output vectors of the existing tables were originally aggregated into a fifteen-sector classification.[2] In cases where updating to 1963 was required, the following formula was applied:

$$x_{j\ (1963)} = x_{j\ (\text{table year})} \frac{va_{j\ (1967)}}{va_{j\ (\text{table year})}} \qquad (j=1, \ldots, 15) \qquad (1)$$

where x_j and va_j are the Gross Outlay (=Gross Output) and Value Added respectively of sector j.

The present task then was to construct on the basis of the original eight 15-sector gross output vectors eleven 21-sector vectors through expansion of the original agricultural and mining sectors as set forth in the classification scheme of the preceding section.

For the Agricultural Sectors of the 11 Western States gross outputs were estimated by Bollman and Woods.[3] The Mining Sectors were disaggregated on the basis of area production statistics given in the 1963 Census of Mineral Industries.[4]

The Gross Output vector of the Four State Region (block) was disaggregated as stated above and then subdivided into the four individual vectors for each of the states based on value added, employment, value of shipments, and other state data which were available according to the following formulation:

$$(x_j)_{\text{State}} = \frac{(va_j)_{\text{block}}}{(va_j)_{\text{state}}} \cdot (x_j)_{\text{block}} \qquad (j=1, \ldots, 21) \qquad (2)$$

---

[1]H.C. Davis, *op. cit.* pp. 68-75, 123-124.

[2]As several of the state models were independently constructed, the numbers of sectors and definitions of these sectors vary among tables. Where definitional problems arose in the aggregation process, the author relied primarily upon his own judgement. See Davis, *op. cit.* pp. 95-97.

[3]Lowell Wood and Frank Bollman, "An Evaluation of the Estimates of Gross Domestic Agricultural Output as used in Input-Output Studies with Particular Reference to the Western States," Giannini Foundation Paper, November 1967, Unpublished Working Paper, Table 7, p. 25.

[4]U.S. Department of Commerce, Bureau of the Census, *Census of Mineral Industries, 1963,* Washington, D.C. 1964.

2. Interindustry Transactions

The transactions or processing matrix of each of the eleven states was calculated in two basic operations. First, the 1958 national table was aggregated[1] to correspond to 21 sectors and then the gross output vector of each state was converted into a diagonal matrix and premultiplied by the U.S. national coefficient matrix.[2] The usage of national coefficients implies that the average production technique of each industry on the national level with regard to foreign and domestically produced goods is identical to the production techniques in each of the states.[3] The application of the national coefficients was made necessary by the fact that the Leontief/Strout model requires a commodity import (column) vector for each of the states under consideration; however, of the western state tables only the Oregon and California tables contain commodity import vectors, while the remaining tables present only sectoral (row) imports.[4]

Second, for each of the states, imports from other regions in the U.S. were subtracted from the transactions matrix to derive estimates of the domestically produced commodities. The regional import matrix was constructed from the commodity import vector by means of the assumption that regional imports were distributed among each of the state industries in the same proportions as their domestically produced counterparts (as emphasized in the introductory section, Part B, above). The regional import matrix was then subtracted from the original transactions matrix to obtain a matrix of domestic transactions.[5]

This procedure, used in conjunction with the 1958 National Input-Output technical coefficients, permitted the development of state transactions matrices in which the flows were shown "net" of required imports to meet estimated output and final demand levels for each sector. In the event that a state showed no productive facilities for a given sector, the transaction matrix showed row values as "non-competitive" imports and a corresponding null vector for the column.

---

[1] The aggregation of the 1958 national model into 21 sectors and the construction of the state transactions matrices was accomplished by the Computer Programs shown in Appendix V.

[2] U.S. Department of Commerce, Office of Business Economics, *Survey of Current Business*, Washington, D.C., September 1965, pp. 40-44.

[3] Since 1958 national coefficients have been used to develop 1963 state tables, either one of the following two assumptions is implied: that no technical progress occurred at the national level between 1958 and 1963 and the input pattern of each of the states is the same as that of the nation in 1963; or that with technical progress on the national level the production techniques in each of the western states in 1963 is about the same as that of the average national level in 1958.

[4] For the distinction between commodity and sectoral imports and the procedure for converting the latter to the former, see Appendix II.

[5] These calculations were performed through the use of the *Program for Constructing a Regional Model from National Interindustry Relationships,* Appendix V.

3. Final Demands

For each of the state tables there are five components of final demand: Private Consumption, Capital Formation, Federal Expenditures, State and Local Expenditures and Net Commodity Exports.

For most of the states under consideration, 1963 estimates of the four final demand vectors for the 15 sectors were available.[1] It therefore remained as a further task to disaggregate each of the final demand elements for agriculture and mining into four categories. The distribution procedure was made on the basis of the 1958 national demand pattern for these sectors.

Four of the five final demand vectors inclusive of regional imports[2] (i.e., before the import matrix was subtracted) were available for the 81 sector model for California and thus each of the vectors was aggregated to 21 sectors. Similarly, the final demands in the 1963 Oregon table (29 sectors) were inclusive of regional imports and thus each vector was aggregated to correspond to the 21 sectors.[3] The regional imports total for each of the final demand vectors of the 1963 input-output studies for Utah and Washington (27 and 39 sectors respectively) were first distributed proportionately among the sectors, then each vector was aggregated to correspond to the 21 sector classification.

The procedure for developing the 21 sector estimates for the four final demand vectors for each of the four states originally grouped into a single region (block) (Nevada, Idaho, Montana and Wyoming) consisted of four steps. First, imports of the aggregate final demand vector were distributed proportionately among the 15 sectors. Second, the sum of the elements of the original single final demand vector for the block was proportioned into four control totals of consumption, capital formation, federal expenditure and state and local expenditures on the following basis:

$$(\sum_i f_{ij})_{\text{block}} = (\sum_i \sum_j f_{ij})_{\text{block}} \cdot \frac{(\sum_i f_{ij})_{\text{nation}}}{(\sum_i \sum_j f_{ij})_{\text{nation}}} \quad \begin{matrix}(i=1, \ldots, 15) \\ (j=1, \ldots, 4)\end{matrix} \tag{3}$$

where f_{ij} = is the j th category of final demand for the output of sector i.

Third, the control totals for each of the four block final demand categories (Consumption, Investment, Federal Expenditure, and State and Local Expenditure) were subdivided into their corresponding state control totals as follows:

---

[1]For those final demand vectors of the available state tables based on years prior to 1963, updating procedures similar to those set forth in the preceding section on gross outputs were employed. For Colorado and the four state block of Nevada, Idaho, Montana and Wyoming, final demand vectors were originally constructed according to the methodology contained in Appendix III.

[2]Regional imports refer to all imports from other regions of the U.S. as opposed to foreign imports which refer to imports from outside the U.S.

[3]The authors of the 1963 Oregon tables did not undertake the development of a regional import matrix nor the necessary subtraction of that matrix from the transaction matrix to yield the domestically produced commodity flows. Thus, the Oregon flow matrix includes both domestically produced goods and regional imports.

$$(\sum_i f_{ij})_{state} = (\sum_i x_i)_{state} \cdot \frac{(\sum_i f_{ij})_{state}}{(\sum_i x_i)_{block}} \quad (4)$$

where X_i is the gross output of sector i.

Finally, each of the four final demand control totals for each state was distributed among the 21 sectors on the basis of national final demand proportions.

Once the abovementioned four 21-sector final demand vectors had been established for each of the four states, net commodity import and export vectors were constructed as follows:

$$(I-A)X - \sum_j f_{ij} = \begin{matrix} + \text{ exports} \\ - \text{ imports} \end{matrix} \qquad (i=1, \ldots, 21) \quad (5)$$

A positive difference indicates that domestic output was more than enough to satisfy estimated total state demands and the remainder must therefore have been exported. A negative result indicates that the state output of that particular commodity was insufficient to meet domestic demand and that an amount of the commodity equal in value to the difference between domestic demand and gross output was imported.

4. Payment Sectors

The sectoral import vector for each state table was derived from the regional import matrix as follows:

$$sm_j = \sum_{i=1}^{21} m_{ij} \quad (6)$$

where m_{ij} is the element of the i th row and j th column in the import matrix.

For each table the foreign sectoral import vector (from outside the U.S.) which was extimated on the basis of national coefficients as follows:

$$(fsm_j)_{state} = \frac{(x_j)_{state}}{(x_j)_{nation}} \cdot (fsm_j)_{nation} \qquad (j=1, \ldots, 21) \quad . \quad (7)$$

where fsm_j refers to foreign imports of sector j.

The total imports row was developed by summing sectoral and foreign imports.

The value added row was determined as follows:

$$(v_j)_{state} = (v_j)_{nation} \quad \frac{(x_j)_{state}}{(x_j)_{nation}} \quad (8)$$

ESTIMATION OF THE NET INTERSTATE TRADE FLOWS

The western interstate trade flows shown in Appendix IV were determined by the following procedure:

First, a model for the eleven states combined as a single region was constructed. The net commodity import and export vectors were then determined according to the procedure outlined in the construction of an individual state table..

If the net commodity trade vector indicated that the western states as a single region of the United States exported x amount of a particular commodity outside the region and if the sum of each state's net exports of that commodity is E (where $E>x$), then, E-x is the amount flowing between individual states of the western region. Similarly, if the western states as a group imported y amount of a particular commodity and the sum of the individual state's imports of that commodity is M (where $M>y$) then M-y must likewise be the amounts of the commodity flowing between the western state economies.

In either case the trade flows were distributed among the importing and exporting states in proportion to their level of trade.

To illustrate, suppose a western regional model composed of only four states yields a value of imports of commodity i into the west as a whole of 80. Suppose further that the state tables reveal the following information with regard to net trade in commodity i:

	IMPORTS	EXPORTS
Washington	90	
Oregon		140
California		160
Colorado	110	

Interstate imports must be 200-80=120 (total imports minus imports from outside the West) and the corresponding share of each state's trade with the others is determined as follows:

IMPORTS		EXPORTS	
Washington	90/200 . 120 = 54	Oregon	140/300 . 120 = 56
Colorado	110/200 . 120 = 66	California	160/300 . 120 = 64
	120		120

The resulting net trade flows upon which the multistate table is based are presented in Appendix IV. To determine the origins and destinations of the interstate trade flows, the computer program described in Appendix V[1] was used.

[1]This program is a computerized version of the "exact solution" variant of the Leontief/Strout gravity distribution technique.

Two additional sets of data are necessary as inputs to the computer program. First, a transport cost factor ($d_{i.gh}$ – the inverse of the transport cost of moving a unit of i from g to h) for each commodity under consideration. Ideally, this would entail a weighted average of air, water, rail and trucking transport costs between the states for each of the 21 commodities. Due to lack of data for some means of transportation and lack of comprehensive ones in others, the transportation cost variable was assumed to be the reciprocal of the distance in miles between the economic centers of gravity (i.e.: primary shipment points) of the western states, shown in Table B-4.

The second variable is $\delta_{i.gh}$ which is a parameter intended to indicate whether any shipments took place from one region to another. If, for any of a variety of economic reasons, no export of i from g to h is expected, $\delta_{i.gh}=0$; otherwise $\delta_{i.gh}=1$. Again, ideally, data regarding shipments of each of the commodities in question should have been the governing data for this variable; however, data deficiencies necessitated the adoption of a value of unity to be assigned to the δ variable for every potential commodity flow between each pair of states.

The method by which the Leontief/Strout model assigns the origins and destinations of the inter-state trade is best explained by example. Consider the flow of Paper and Allied Products from the State of Washington to California, $X_{Wash.\text{-}Calif.}$. The exports of Washington to California of the commodity was estimated through the following equation:

$$X_{\text{Wash.-Calif.}} = \frac{S_{\text{Wash.}}\ D_{\text{Calif.}}\ (C_{\text{Wash.}} - K_{\text{Calif.}})\ d_{\text{Wash.-Calif.}}\ \delta_{\text{Wash.-Calif.}}}{S_{\text{West}}\ \ (\text{where } S_{\text{West}} \equiv D_{\text{West}})}$$

Where $S_{Wash.}$ represents the supply of Paper and Allied Products by Washington to the West

$$(S_{\text{Wash.}} = ID_{\text{Wash.}} + FD_{\text{Wash.}} + EX^{\text{net}}_{\text{Wash.-West}})$$

and $D_{Calif.}$ equals the demand of California for the production of the Paper and Allied Products sector of the Western Regional Economy

$$(D_{\text{Calif.}} = ID_{\text{Calif.}} + FD_{\text{Calif.}} + M^{\text{net}}_{\text{West-Calif.}}).$$

The constants $C_{Wash.}$ and $K_{Calif.}$ characterize the relative position of Washington vis-a-vis all other states as suppliers, and of California as a relative consumer of the commodity. The C and K parameters are determined indirectly.[1]

Once the Leontief/Strout Model was utilized to determine the origins and destinations of the calculated interstate trade, an interstate trade matrix for each commodity was developed. The next step was to distribute these imports sectorally to develop an intra-regional import matrix from each of the intra-regional import

[1] See Part A, Appendix: The Gravity Distribution Technique in Mathematical Summary, page 14.

vectors. This was accomplished by adopting the same assumption made in the process of developing each state's import matrix, viz., that the imports of commodity x into a state follow the same sectoral distribution pattern as does the domestically produced commodity x. Since each state trades with the other ten in terms of 21 commodities, 110 (11 x 10) import matrices of order 21 were formed. These matrices were then combined with the eleven state models to produce a 231 (11 x 21) order interregional interindustry model of the type described in Part A (page 11). In other words, for each state we have now an intra-regional import matrix from each of the other states.

Table B-4.
RECIPROCALS OF WESTERN INTERCITY DISTANCES (HUNDREDS OF MILES)

	Albuquerque	Bakersfield	Boise	Denver	Phoenix	Portland	Salt Lake City	Seattle	Helena (Mont)	Cheyenne	Carson City
Albuquerque		0 1190	0 1016	0 2364	0 2198	0 0707	0 1645	0 0672	0 0909	0 1879	0 1293
Bakersfield	0 1190		0 1202	0 0866	0 1996	0 1134	0 1355	0 0936	0 1157	0 1165	0 3846
Boise	0 1016	0 1202		0 1190	0 1028	0 2247	0 2762	0 1934	0 1865	0 1312	0 2762
Denver	0 2364	0 0866	0 1190		0 1221	0 0778	0 1972	0 0737	0 1233	0 9880	0 0941
Phoenix	0 2198	0 1996	0 1028	0 1221		0 0785	0 1460	0 0671	0 0845	0 1086	0 1280
Portland	0 0707	0 1134	0 2247	0 0778	0 0785		0 1239	0 5682	0 1457	0 0833	0 1639
Salt Lake City	0 1645	0 1355	0 2762	0 1972	0 1460	0 1239		0 1137	0 2008	0 2183	0 1841
Seattle	0 0672	0 0936	0 1934	0 0737	0 0671	0 5682	0 1137		0 1639	0 0781	0 1256
Helena (Mont)	0 9090	0 0057	0 1865	0 1233	0 0845	0 1457	0 2008	0 1639		01404	0 1083
Cheyenne	0 1879	0 1165	0 1312	0 0880	0 1086	0 0833	0 2183	0 0781	0 1404		0 0988
Carson City	0 1293	0 2846	0 2762	0 0941	0 1280	0 1639	0 1841	0 1256	0 1083	0 0988	

THE ELEVEN STATE INTERINDUSTRY ECONOMIC TRANSACTIONS MODEL—1963

1. Table of Interindustry Transactions

In the Interstate Interindustry Transactions Table, the 21 order intrastate transactions matrices are shown along the principal diagonal. Column elements, other than the 21 order blocks along the principal diagonal, can be viewed as net imports, in millions of dollars, required by the sectors designated for each state. Row elements represent exports from the producing sectors of each state to the designated consuming sectors of each of the other states. The states of the region were entered into the table in the following order:

1. Arizona
2. California
3. Colorado
4. New Mexico
5. Oregon
6. Utah
7. Washington
8. Nevada
9. Idaho
10. Montana
11. Wyoming

A schematic representation of the multiregional transactions table is shown on the following page as figure B-1.

In several of the states not all productive sectors are represented. California and Colorado exhibit all of the sectors as defined. For the remaining states the following sectors have no productive capacity:[1]

State	Sectors
Arizona	3, 8
New Mexico	15
Oregon	6
Utah	3
Washington	8
Nevada	3, 11, 12, 14
Idaho	6, 14
Montana	11, 12
Wyoming	11, 12, 13, 15

[1]In the transactions matrix sectoral imports for the deleted rows have been treated endogenously. The column entries for these sectors are zeros.

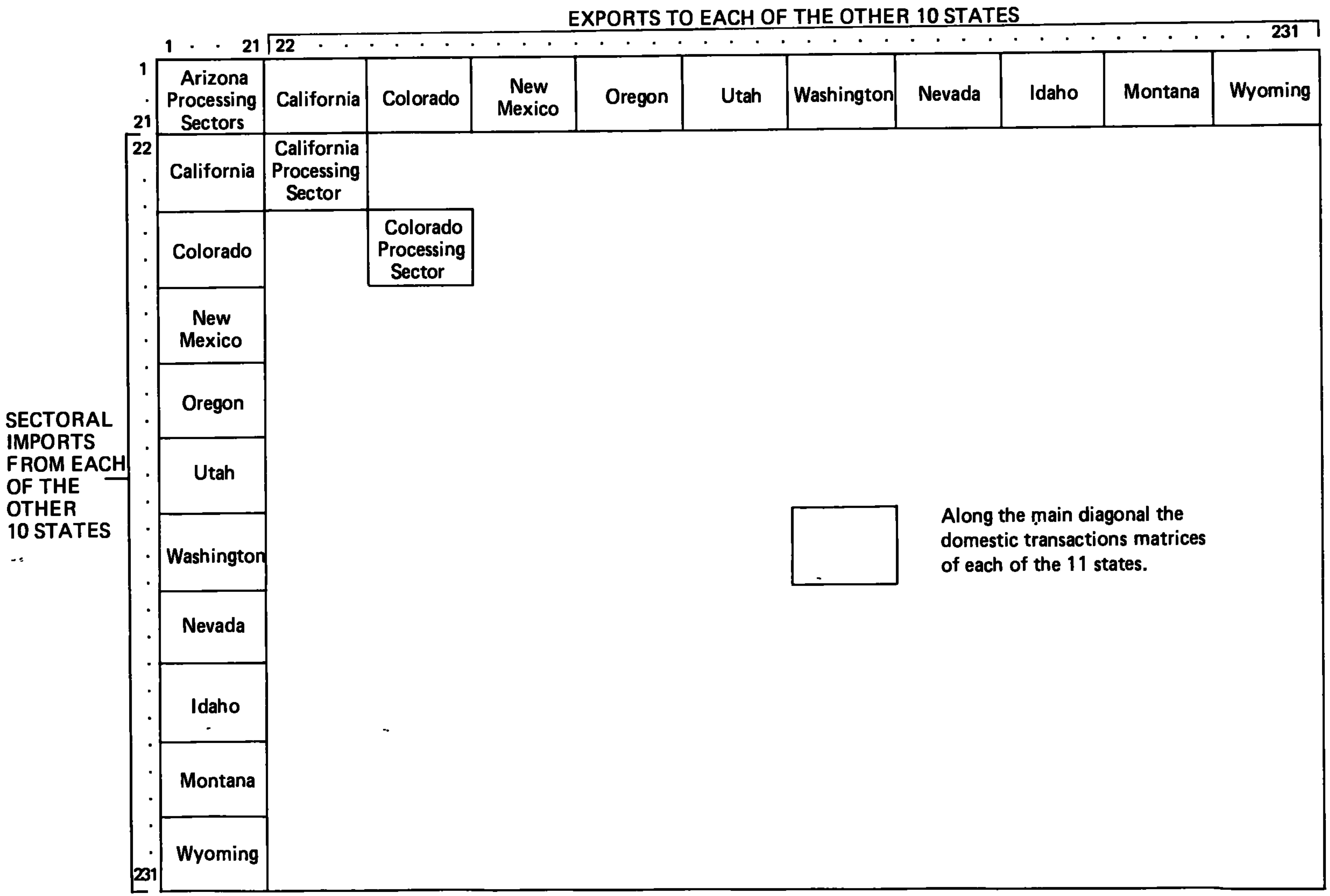


Figure B-1. Format of Western Interstate Interindustry Transactions Table.

2. The Economic Transactions Multipliers

As a first step in developing trade flow impacts the "Leontief Inverse" was derived from the interstate transactions matrix as discussed in Part A. The 231 order inverse for the Western states is shown as Table B-5. The elements of this matrix show the total requirements from each regional industry that are necessary to deliver one unit of output to the final demand categories. Reading along a given row of the transposed inverse every element shows the dollar amount required from the industry designated at the top, by industries similarly designated at left to deliver a dollar of output to final demand. The sectors which have no productive capacity in the various states as noted earlier have row elements where are all zeros except for the intersection of the row and column bearing the same designation, i.e., where i=j. The zeros appearing for each row element where i ≠ j indicates that a non-existent industry makes no purchases in or out of state as would be expected. Other industries within the state requiring the products of this sector, satisfy their needs by means of imports which by the nature of state production must be "non-competitive." These imports are shown as column entries, in each instance, where domestic productive capacity is lacking.

The Leontief Inverse provides detailed information on the relationship of each sector to every other sector of the entire region. For some purposes it may be sufficient to know the percentage impact that expenditures made for the products of one state sector will have on other states. This information can be summarized in a table which has 11 columns and 231 rows, the elements of which are given in percentage terms. To obtain these summary results two final steps are necessary. First, the row elements of the transposed Leontief inverse are summed over 21 columns at a time, and, row totals are also formed for the 11 state region as a whole (Table B-6). Second, each element of the 11 element rows is divided by the row total to yield a corresponding percentage, (Table B-7). The results shown in Table B-7 can be interpreted to mean that per unit change in final demand for each sector designated at left, the percentage of the total 11 state region impact that will occur in each of the states shown at the top. The columns 1 through 11 are identified by the list of states as shown on page 28. For example Row 1 shows that for a 1 dollar (or 1 million dollar) change in the final demand for Arizona Livestock and Livestock Products the following impacts will be felt in the states of the region:

State	Percent Impact
1. Arizona	85.06
2. California	8.69
3. Colorado	1.59
4. New Mexico	1.25
5. Oregon	.64
6. Utah	.55
7. Washington	.53
8. Nevada	.02
9. Idaho	.33
10. Montana	.62
11. Wyoming	.72
Total =	100.00

Similar demands for the production of Nevada Livestock Products (Row 148) result in the following impacts:

State	Percent Impact
1. Arizona	.50
2. California	21.12
3. Colorado	1.22
4. New Mexico	.75
5. Oregon	1.19
6. Utah	.31
7. Washington	.71
8. Nevada	71.33
9. Idaho	1.24
10. Montana	.97
11. Wyoming	.66
Total =	100.00

In the second example the gravity flow estimates indicate that approximately 21 percent of the impact of expenditures for Nevada Livestock Products are felt in the California economy. The distribution of impacts in the second example is substantially different from that of the first which traced the impacts for the counterpart Arizona sector.

A review of the impact percentages for the economy of a state directly, and via trade patterns to the other state economies, indicates that for most states the greatest impact is a direct one. Apart from the state sectors listed earlier which have no productive facilities, and are readily identified in the final table by the appropriate 1.0000, direct impacts range as high as 98 percent, and alternately, to a low of 19 percent. One reason for the rather high direct impact values that appear in many instances may be the level of industry aggregation which shows only 21 productive sectors for each state. Greater disaggregation might give lower values in some instances. The gravity model results are presented as a first approximation to the actual trade patterns prevailing in the West.

APPENDIX I

INTERREGIONAL INTERINDUSTRY MODELS

1. Leontief Model

A 'balanced regional' input-output model was proposed by Leontief in 1953.[1] In his model Leontief gave recognition to the fact that while some goods are consumed near their point of production, others are transported considerable distances prior to consumption.

Within the model the nation is conceived in terms of n regions. Each region produces m goods of which the first h are regional and the remaining m-h are national. It is assumed that the percentage division between regional and national of easily transportable goods is known and invariant with changes in output. The system thus contains two sets of constants. The first is the technological matrix or structural equations which are assumed identical for all regions:

$$a_{ik} = \frac{x_{ik}}{x_k} \qquad (i,k = 1, \ldots, m) \qquad (1)$$

where x_{ik} = the amount of production of commodity i required by industry k and x_k = the total output of industry k. The second set of constants is the national-regional proportionality matrix:

$$_j r_g = \frac{_j x_g}{x_g} \qquad \begin{matrix}(g = h + 1, \ldots, m)\\(j = 1, \ldots, n)\end{matrix} \qquad (2)$$

where $_j x_g$ = the amount of national good g produced in region j.

Substitution of the structural equations into the input-output balance equations yields

$$x_i - \sum_{k=1}^{m} a_{ik}\, x_k = y_i \qquad (i = 1, \ldots, m) \qquad (3a)$$

---

[1]Wassily Leontief. *Studies in the Structure of the American Economy.* New York: Oxford University Press, 1953, pp. 93-116.

where y_i = the final demand for commodity i. Since the final demand vectors are determined exogenously and the $[a_{ik}]$ matrix is known, there exist m equations and m unknowns. Hence the equations may be solved for the outputs of the 'national goods' x_i. These national goods are commodities whose production and consumption balance only for the nation as a whole. The so-called 'regional goods' are distinguished by the fact that their production-consumption balance is struck within the region.

Under the assumption of a constant national-regional proportionality matrix, once the total output of national goods is known, the system is easily solved for the regional outputs of the national goods,

$$
{}_jx_g = {}_jr_g x_g \qquad \begin{matrix}(g = h+1, \ldots, m). \\ (j = 1, \ldots, n)\end{matrix} \tag{3b}
$$

It remains now only to solve for the regional outputs of regional goods. This is done by solving as a set of simultaneous relationships the first h equations of (3a).

$$
{}_jx_\ell - \sum_{i=1}^{m} a_{\ell i}\, {}_jx_i = {}_jy_\ell \qquad \begin{matrix}(\ell = 1, \ldots, h) \\ (j = 1, \ldots, n)\end{matrix} \tag{3c}
$$

There are in (3c) h equations, one for every regionally balanced output. On the other hand, since each region produces m commodities, there are, including final demand, m + 1 variables. The final demand vectors have been previously constructed, however, and the regional outputs of the national goods m-h are determined by (3b). Hence there are but h unknowns and the h equations may be solved for the regional production of regional goods.

The net external trade balance (NETB) for each region may also be found within the system:

$$
\text{NETB} = {}_jx_g - \sum_{i=1}^{m} a_{ig}\, x_g - {}_jy_g \qquad \begin{matrix}(g = h+1, \ldots, m) \\ (j = 1, \ldots, n)\end{matrix} \tag{4}
$$

However, the model will not reveal for any region the geographic origins of its imports nor the destinations of its exports.

Although the Leontief model has been termed interregional because it incorporates within its structure two or more regions, the model is more accurately labelled intranational as it is generally designed for the disaggregation of a national model. Implementation of the model involves as the initial step the breakdown of the national or area final demand vectors into their regional components in such a way that the sum of the latter is consistent with the national or area totals. Using national input coefficients to characterize production in each region and constant ratios to distinguish between regional and national commodities, the system is solved according to the stages outlined above.[1]

[1] Such an implementation of the U.S. 1947 national model was undertaken by Walter Isard, "Some Empirical Results and Problems of Regional Input-Output Analysis," *op. cit.*, pp. 116-81.

2. Isard Model

A true interregional model, as opposed to Leontief's intranational system, was put forth by Isard in 1951.[1] Isard constructed his model on a different assumption regarding interregional trade. Interregional flows occur, Isard argues, due to two basic factors: 1) the existing geographic inequalities in population, income, and resources; and 2) production indivisibilities and the resulting economies of large-scale production and specialization. This heterogeneity on both sides of the market gives rise to a corresponding heterogeneity of physically similar commodities produced in different regions.

> As a necessary consequence, any given good or service produced in any region must be taken as a unique commodity, distinct from the same good or service produced in any other region. Thus, for example, if states are designated as regions, Pennsylvania brick becomes a commodity different from New York brick or California brick. In terms of a rigorous theory, this is as it should be, since Pennsylvania brick producers have a market area different in shape and extent from that of New York or California producers. In effect, then, the number of industrial categories in a territory is multiplied by the number of regions, if each region engages in each activity.[2]

Isard was the first to formulate an interregional model explicitly incorporating different technological matrices. In an n-region, m-commodity or -industry area, each regional technical matrix $[{}_{k\ell}a_{ij}]$ is expressed as

$$ {}_{k\ell}a_{ij} = \frac{{}_{k\ell}x_{ij}}{{}_{\ell}x_j} \qquad \begin{array}{l}(i, j = 1, \ldots, m)\\ (k = 1, \ldots, n)\end{array} \qquad (5)$$

where ${}_{k\ell}x_{ij}$ = the amount of output of industry i in region k sold to industry j in region ℓ, and ${}_{\ell}x_j$ = the total output of industry j in region ℓ. Substitution of the above structural equations into the balance equations yields

$$ {}_{k}x_i - \sum_{\ell=1}^{n} \sum_{j=1}^{m} {}_{k\ell}a_{ij}\, {}_{\ell}x_j = {}_{k}y_i \qquad \begin{array}{l}(i = 1, \ldots, m)\\ (k = 1, \ldots, n)\end{array} \qquad (6)$$

where ${}_{k}y_i$ = the final demand in region k for commodity i. The system is handled in the same manner as the simple input-output model discussed in Part A. The general solution to the above system is expressed as

$$ {}_{k}x_i = \left[I - \sum_{\ell=1}^{n} \sum_{j=1}^{m} {}_{k\ell}a_{ij}\, {}_{\ell}x_j \right]^{-1} {}_{k}y_i \qquad \begin{array}{l}(i = 1, \ldots, m)\\ (k = 1, \ldots, n)\end{array} . \qquad (7)$$

The format of the Isard model is as follows:

---

[1] Walter Isard. "Interregional and Regional Input-Output Analysis: A Model of a Space-Economy," *Review of Economics and Statistics*, 33:318-29, November 1951.

[2] Walter Isard, *op. cit.*, p. 320.

INTERREGIONAL INPUT-OUTPUT TABLE

		East			South			West					Subtotals		
		1. Agriculture	9. Chemicals	20. Households and Govt.	1. Agriculture	9. Chemicals	20. Households and Govt.	1. Agriculture	9. Chemicals	20. Households and Govt.	Exports (national)	Total Output	1. Agriculture	9. Chemicals	20. Households and Govt.
East	1. Agriculture														
	9. Chemicals														
	20. Households and Govt.														
South	1. Agriculture														
	9. Chemicals														
	20. Households and Govt.														
West	1. Agriculture														
	9. Chemicals														
	20. Households and Govt.														
	Imports (national)														
	Total Input														
Subtotals	1. Agriculture														
	9. Chemicals														
	20. Households and Govt.														

Source: Walter Isard, "Interregional and Regional Input-Output Analysis: A Model of a Space Economy," *Review of Economics and Statistics, 33*:321, November 1951.

Figure B-2. Format of the Isard Interregional Interindustry Transactions Table.

No attempt will be made to provide an interpretation of the table since only a modified version is presented in this report.

In summary, the Isard interregional model differs from the Leontief intranational model primarily in three ways. First, the Leontief model assumes one technical coefficient matrix, whereas the Isard model can incorporate as many different technical matrices as exist regions in the model. Secondly, the Leontief model uses output and consumption data to calculate trade balances and provides no information regarding the origins and destinations of the trade flows. The Isard model derives the trade flows and their origins and destinations from primary data. Thirdly, the Leontief model requires regional specification of final demand for regional goods only, the regional proportions of the production of national goods being predetermined. The Isard model requires specification of final demand for all goods. Thus an increase in the final demand for a national good in the Isard model will produce different results depending on the geographic distribution of the rise in the level of demand, whereas in the Leontief model the regional pattern of increased output is fixed.

Although the Isard model precludes the possibility of substitutability among physically similar but geographically differentiated commodities, it is of the two models the more general. The Leontief model is but a special case of the Isard model in which production functions are invariant among regions, trade flows between regions unknown, and the proportions of inputs from various geographic sources fixed for all sectors in all regions.

While the Isard model provides greater flexibility of use and yields considerably more information, the data-gathering procedures required for implementation of the model in its original form are extremely costly and time-consuming. Attempts to adopt the model in practice have led, primarily because of the scarcity of data pertaining to interregional trade flows, to modification of the model through the employment of simplifying assumptions.

3. Chenery-Moses-Wonnacott or Modified Isard Model

The Chenery study[1] of the North-South interdependence of the Italian economy, the Moses U.S. model,[2] and the Wonnacott study of the interrelationship between Canada and the U.S.[3] all represent individual efforts to fulfill the statistical requirements of the basic Isard model. While each study naturally possesses its unique characteristics, all are similar in that each employs essentially the same simplifying assumption necessary for the practical application of the Isard model.

This common assumption, as stated earlier, is that imports of commodity i into

[1] Hollis Chenery. "Interregional and Intranational Input-Output Analysis," in T. Barna, ed., *The Structural Interdependence of the Economy, op. cit.*, pp. 51-103.

[2] Leon Moses. "A General Equilibrium Model of Production, Interregional Trade and Location of Industry," *Review of Economics and Statistics, 42*:373-97, November 1960.

[3] R.J. Wonnacott. *Canadian-American Dependency, An Interindustry Analysis of Production and Prices.* Amsterdam: North-Holland Publishing Co., 1961.

region j will be distributed throughout the sectors of region j in the same manner as is the output of commodity i produced in region j. To state the assumption in terms of the Isard model,[1] let us first set forth the Isard coefficients in the form:

$$_{k\ell}a_{ij} = {}_{\ell}a_{ij}\ {}_{k\ell}t_{ij} \tag{8}$$

where $_{\ell}a_{ij}$ is the total amount of commodity i purchased by sector j in region ℓ and $_{k\ell}t_{ij}$ is the proportion of total purchases of commodity i made by sector j in region ℓ that comes from region k. These $_{k\ell}t_{ij}$ terms are the 'supply coefficients' of the Chenery model and the 'trade coefficients' of the Moses and Wonnacott models. All three studies make the assumption that

$$_{k\ell}t_{ij} = {}_{k\ell}t_{i} \quad \text{for all j.} \tag{9}$$

4. Present Model

Due to the similar limitations of available data, the present study is forced to adopt the same simplifying assumption discussed immediately above. Since the trade vectors of California and Oregon are presented in net terms only, an additional constraint of working solely with net trade flows among the states is imposed. This latter constraint necessitates for balancing purposes the existence of an adjustment column and an adjustment row in the table. An entry in the adjustment column represents the difference between the total output of sector i and the sum of internal consumption and total net exports of sector i's production. An entry j in the adjustment row represents the value of sector j's total inputs (total output) less the sum of sector j's consumption of internally produced inputs and the inputs imported from the other states in the model. The adjustment row includes foreign sectoral imports, as estimates of these flows cannot be as readily constructed within the model as were those of foreign commodity exports since the gravity flow model, which was utilized for the assigning of origins and destinations of the trade flows, deals only with commodity flows.

A two-sector, three-state version of the format of the table for the Pacific States is presented below with a description of representative cells within the table.

---

[1]The discussion here follows the argument of R.E. Kuenne, *The Theory of General Economic Equilibrium.* Princeton, N.J.: Princeton Univ. Press, 1963.

		California				Oregon				Washington							
		Agriculture	Industry	Final Demand	Total	Agriculture	Industry	Final Demand	Total	Agriculture	Industry	Final Demand	Total	Exports - Pacific	Exports - Foreign	Adjustment	Total Output
California	Agriculture																
	Industry																
	Total																
Oregon	Agriculture	a		b	c		d		e					f	g	h	i
	Industry																
	Total						j										
Washington	Agriculture																
	Industry																
	Total																
	Imports - Pacific						k							ℓ			
	Value Added						m		n								
	Adjustment						o										
	Total Input						p										q

a — the value of that portion of Oregon agricultural output purchased by California agriculture.
b — the value of that portion of Oregon's agricultural output sold to satisfy the final demand sector of California. The aggregated final demand vector for each state in the table excludes all exports.
c — the total value of Oregon agricultural production exported to California.
d — the value of that portion of Oregon agricultural output purchased by Oregon industry.
e — the total value of Oregon agricultural production consumed domestically.
f — the value of that portion of Oregon agriculture exported to California and Washington.
g — the value of that portion of Oregon agricultural output exported to states other than Pacific Coast states and to foreign countries.
h — export adjustment figure (= i - e - f - g).
i — the total value of Oregon agricultural production.
j — the value of domestically produced inputs purchased by Oregon industry.
k — the value of inputs purchased by Oregon industry from California and Washington.
ℓ — the total value of trade among the Pacific Coast states (imports - exports).
m — value added of Oregon industry.
n — gross state product of Oregon.
o — import adjustment figure (= o - j - k - m).
p — the total value of inputs purchased by Oregon industry.
q — the total gross output of the Pacific Coast states (gross output = gross input).

Figure B-3. Illustrative Interpretation of Pacific States Interstate Interindustry Transactions Table.

The major cells along the main diagonal (upper left to lower right) are the major portions of the input-output tables of the states. All other major cells represent two-state trade matrices. The subtotals major row and column were not carried over from the Isard table in their original format. For Isard, the

> . . . subtotals major row is of particular interest. It sums the data in each minor vertical column by industry, yielding inputs into each industry of each region from each of the 20 industries as a national whole. Thus a comparison of the appropriate columns in this subtotals major row brings out the differences in production practices and consumption habits among regions.[1]

The subtotals format is of less interest in the present study since we wish to focus attention more upon the interregional trade relationships than upon differences in regional production and consumption patterns.

It should be pointed out that while we have assumed as did Isard identical sector classifications for each region in the model, there is no theoretical reason why this need be so. In fact, a sectoral classification that varies from region to region is not only feasible, it is most often highly desirable. There are several reasons why the analyst might prefer such a scheme, the primary one being to bring emphasis upon the more important activities in each region in regard to output and/or linkage with other sectors within or without the region. However, the present study retains a uniform classification for each region. This is done less for theoretical reasons than for practical ones. If sufficient primary data on Western interstate trade flows were available, there would be no barriers to varying classification. In view of the present state of the data, however, the interstate trade coefficients must be generated by a statistical technique – the gravity flow method – that requires export and import figures to be of a regionally uniform classification.

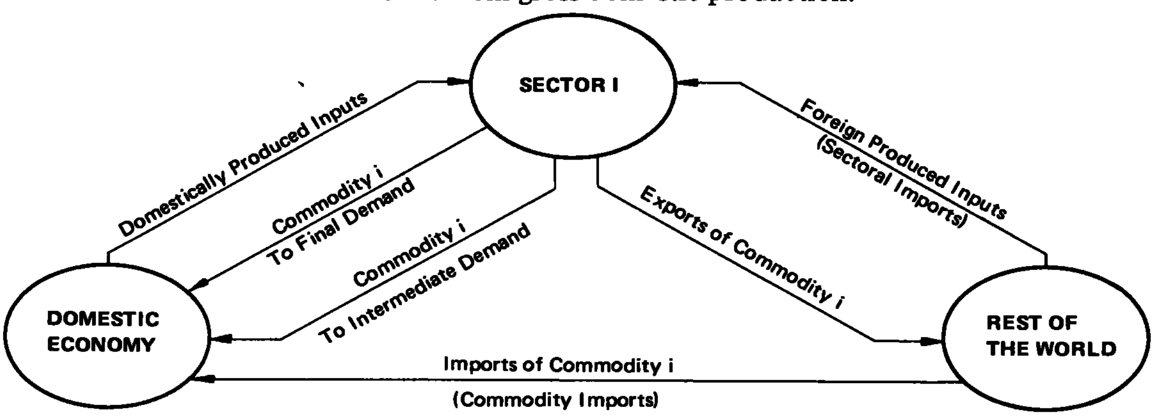
[1] W. Isard, *op. cit.*, p. 324.

APPENDIX II

COMMODITY AND SECTORAL IMPORTS

1. Definition – Imports: Commodity vs. Sectoral

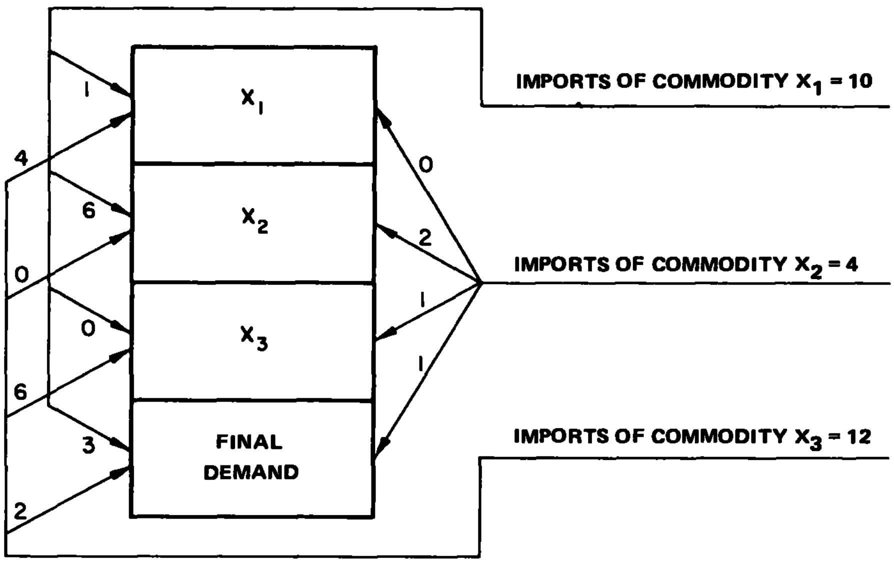
While it is sometimes useful in input-output analysis to distinguish between competing and noncompeting imports, it is quite necessary to understand fully the differences between commodity (row) and sectoral (column) imports. It will be recalled that one determines from the input-output table the distribution pattern of the production of a particular sector by reading along that sector's row. The row or commodity import (cm_i) is thus a measure of the quantity of commodity i which was imported into the domestic economy. Figures for the flows of this type of imports are derived in the same manner as are those for exports: gross commodity imports are estimated by sampling techniques based on primary or secondary data; net commodity imports are customarily calculated by subtracting domestic intermediate and final demands from gross domestic production.



What is here termed "sectoral" imports are those imports that appear not as a column vector as do commodity imports but are found in vector form as a row in the payments sector of the table. Reading down a column one determines the structure of inputs to, or outlays of, a particular sector. A column import (sm_j) is thus a measure of the amount that sector j expended upon foreign-produced inputs. Sectoral imports, in contrast to commodity imports, can be, and in all probability will be, composed of more than a single commodity. Entries for this type of import will appear not only in the columns of the processing matrix but in those of the final demand sector as well.

The differences between the two types of imports can be illustrated by a simple numerical example. Let us suppose the existence of a three-sector economy (X_1, X_2, X_3) which imports on a gross basis the product of each of the three sectors. It can be seen from the example that the total of the sectoral imports equals the total of commodity imports since the input-output method of recording the two flows constitutes a double-entry accounting system.

IMPORT FLOW
DIAGRAM



IMPORT FLOW MATRIX

	X_1	X_2	X_3	FD	Total Commodity Imports
X_1	1	6	0	3	10
X_2	0	2	1	1	4
X_3	4	0	6	2	12
Total Sectoral Imports	5	8	7	6	26

2. Conversion of Sectoral to Commodity Imports

Of the Western States' tables only Oregon and California present commodity import vectors; each of the remaining tables contains only a vector of column or sectoral imports. Commodity imports, which are necessary for the linkage of the state economies within the multistate model, cannot easily be calculated from these latter tables via the formula $[I-A]\ X - FD_{agg}$ since the A matrix in the formula is the coefficient matrix derived from a flow table consisting of both foreign and domestically produced goods. As the Utah, Washington, Arizona, and New Mexico transactions tables reveal domestic flows only, the above calculation cannot be immediately performed. Hence an attempt was made to convert the sectoral import vectors of these tables into commodity import vectors via the assumption employed in the discussion of format No. 2 to effect the reverse, that of commodity imports into the sectoral variety. The assumption again is: sector i is presumed to purchase x percent of the total imports of steel into the economy if sector x accounts for x percent of the domestic use of steel.

To illustrate the procedure based on this assumption, let

$[d_{ij}]$ = the domestically produced commodity flows represented in the processing matrix augmented by the final demand vectors, excluding exports

and m_{ij} = an element of the import matrix, representing the amount of commodity i imported by sector j.

To determine the row or commodity imports, it is initially necessary to distribute the column or sectoral imports throughout the processing matrix augmented by the final demand sector excluding the export column. If there are n sectors in the processing matrix and k final demand vectors, excluding exports, then on the basis of the above assumption,

$$m_{ij} = \text{total amount of commodity i imported by all sectors} \times \frac{\text{amount of commodity i purchased by sector j}}{\text{total amount of commodity i purchased by all sectors}}$$

$$= \Sigma\, m_{ij} \frac{d_{ij} + m_{ij}}{\sum\limits_j d_{ij} + \sum\limits_j m_{ij}} \qquad \begin{matrix}(i = 1, \ldots, n)\\ (j = 1, \ldots, n+k)\end{matrix}. \tag{1}$$

We may cross-multiply to obtain

$$m_{ij} \sum_j d_{ij} = d_{ij} \sum_j m_{ij} \qquad \begin{matrix}(i = 1, \ldots, n)\\ (j = 1, \ldots, n+k)\end{matrix}. \tag{2}$$

Dividing each side of the equation by $\sum_j d_{ij}$ and summing over the rows, we obtain

$$\sum_i m_{ij} = \sum_i \frac{x_{ij} \sum_j m_{ij}}{\sum_j d_{ij}} \qquad \begin{array}{l}(i = 1, \ldots, n)\\(j = 1, \ldots, n + k)\end{array} \tag{3}$$

It was then intended to solve the last set of equations for the unknown commodity import vector,

$$cm = \sum_j m_{ij} \qquad \begin{array}{l}(i = 1, \ldots, n)\\(j = 1, \ldots, n + k)\end{array}. \tag{4}$$

Since the final demand vectors for each of the regions within the multistate table have been aggregated into a single vector, $k = 1$ and the procedure thus calls for the solution of $n + 1$ equations for n unknowns.

An approximate solution was achieved by means of ordinary least squares regression but negative elements representing gross commodity imports appeared in the solution vector. Since the purpose of the program was only to distribute the sectoral imports of each purchasing industry among that industry's column of inputs, "negative commodity imports" could not be considered exports and such entities were therefore meaningless. The program was modified through the introduction of additional constraints and iterative procedures toward a nonnegative solution, but reasonable results were not attained.

The problem was then reformulated into a linear programming format in which the solution vector was required to possess only nonnegative elements. The general form of the problem then became that of finding column vector cm such that

$$[d_{ij}]\ cm = sm_t$$

where all elements of the unknown commodity import vector cm, as well as those of the transposed sectoral import vector sm_t, are required to be nonnegative. An efficient method for solving the problem consists of finding nonnegative vectors cm and y such that

$$vy \text{ is a minimum}$$

subject to

$$[d_{ij}]\ cm + y = sm_t \tag{5}$$

where v is a sum vector, $(1, 1, \ldots, 1)$, and y is the solution vector to the dual of the program.[1] The problem was formulated in this manner and data for the state of Washington were tested. No reasonable nonnegative solution vector evolved,

[1]For further discussion of nonnegative solutions to linear equations see David Gale, *Linear Economic Models.* New York: McGraw-Hill, Inc., 1960, pp. 119-21.

however, and the attempted conversion of sectoral imports to commodity imports via the linear programming formulation was also abandoned.

It was then decided to formulate the import conversion on the basis of the assumption that the technological input structure of each state industry in the 15-sector aggregation format bears some relation to that of the same industry defined on the national level. The first step was to postmultiply the U.S. national coefficient matrix by the state gross output vector. The resulting transaction matrix was presumed to resemble, in terms of the *relative* magnitudes of the elements of the column vectors, the transactions table of the state in terms of foreign and domestically produced commodities. From this matrix was subtracted the aggregated 15 x 15 state transactions table in terms of domestically produced commodity flows. The matrix of differences, or "import matrix," was then normalized by the vector of column totals and was presumed to approximate the distribution pattern of state imports. Sectoral imports for each industry j, sm_j, were distributed according to the calculated input coefficients of the "import matrix."

Adopting the symbol "^" to indicate the conversion of a vector into a diagonal matrix, i.e., a matrix whose principal diagonal consists of the elements of the vector and whose off-diagonal elements are zero, the basic calculation procedure was as follows:

$$\left[m_{ij}\right] = \left[A_{(U.S.)}\,\hat{X}_{(State)} - A_{(State)}\,\hat{X}_{(State)}\right] \cdot \left[\hat{IO}^{-1}\right]\left[\hat{SM}\right] \qquad (6)$$

$$(i, j = 1, \ldots, 15)$$

where $\left[m_{ij}\right]$ = the import matrix

A = technical coefficient matrix

$\hat{IO} = \sum_i \left(A_{(U.S.)}\,\hat{X}_{(State)} - A_{(State)}\,\hat{X}_{(State)}\right)$

Sectoral imports into the final demand sector were treated in a somewhat similar fashion. The state final demand totals including sectoral imports for each of the four vectors– Personal Consumption, Gross Private Capital Formation, Federal Government Purchases and State and Local Government Purchases–were calculated and distributed according to the national pattern aggregated to 15 sectors. From the resulting four commodity flow vectors were subtracted the corresponding final demand vectors pertaining to domestically produced flows to obtain the four vectors of imports into the final demand sector.

$$\left(\left(\sum_i (FD_{(State)}) + SM\right)\left(FD_{(U.S.)}\frac{1}{\sum_i FD_{(U.S.)}}\right) - FD_{(State)}\right)\frac{1}{IO^*}\,SM \qquad (7)$$

$$(i = 1, \ldots, 15)$$
$$(j = 16, \ldots, 19)$$

$$\text{where } IO^* = \sum_i \left(\left(\sum_i (FD_{(State)}) + SM \right) \left(FD_{(U.S.)} \frac{1}{\sum_i FD_{(U.S.)}} \right) \cdot FD_{(State)} \right) . \quad (8)$$

In each case the results were examined, comparisons with the results pertaining to other states were undertaken, and some adjustments made in accordance with the author's judgment.

Commodity imports were then calculated as

$$CM = \sum_j m_{ij} \qquad \begin{matrix} (i = 1, \ldots, 15) \\ (j = 1, \ldots, 19) \end{matrix} . \qquad (9)$$

APPENDIX III

CONSTRUCTION OF FINAL DEMAND VECTORS FOR COLORADO; NEVADA–IDAHO–MONTANA–WYOMING

Excluding the vector of exports, there are in each table four final demand vectors–1) Personal Consumption, 2) State and Local Government, 3) Federal Government, and 4) Gross Private Fixed Capital Formation–each of which was constructed on the basis of information derived from several sources.

1. The determination of the Personal Consumption vector was undertaken on the basis of scaling up the California data, rather than scaling down the national figures, since California personal income accounts for nearly two-thirds of that of the West.[1] The first step, therefore, was to calculate the total personal consumption (C) for California in 1963 based on California personal income (Y):

$$\text{Calif. } C_{1963} = \text{Calif. } Y_{1963} \frac{\text{Calif. } C_{1958}}{\text{Calif. } Y_{1958}}. \tag{1}$$

   Once this total was calculated, the average personal consumption figure for each state was then determined on the assumption that it was roughly the same as that of California:

$$\text{State } C_{1963} = \text{State } Y_{1963} \frac{\text{Calif. } C_{1963}}{\text{Calif. } Y_{1963}} \tag{2}$$

   The income and consumption figures for all states in the preceding calculations were taken from the U. S. Statistical Abstract.[2]

   The total personal consumption figure for each state was then distributed among the 15 sectors according to a set of personal consumption coefficients derived from a study of deviations from national consumption patterns by the Western 11-state region.[3]

[1]The choice between the average consumption figure for California and that for the U.S. was actually unimportant in this particular instance as the former was estimated to be in 1963 81.76% while the 1963 figure for the latter was 81.16%.

[2]U.S. Department of Commerce, Bureau of the Census. *U.S. Statistical Abstract, 1965*. Washington, 1966.

[3]Time, Inc. *Life Study of Consumer Expenditure.* New York: Vol. 1, 1957.

2. For the construction of the 1963 vector of purchases of goods and services by the State and Local Government, the control total for each of the five states was derived from data published by the Tax Foundation.[1] From these totals were subtracted all transfer payments to yield a figure for the total State and Local Government purchases. The corresponding 1963 figure for the U. S. was then calculated in the same fashion and compared to the estimate by the Office of Business Economics.[2] The latter figure was slightly lower, and the figure for each of the five Western States was therefore adjusted downward, as the OBE figure was considered to be the more reliable.

   From the 1958 U. S. State and Local Government Purchases vector aggregated to 15 sectors, a percentage distribution was computed and the control total distributed accordingly. An independent estimate on the basis of U. S. Department of Commerce data[3] was then made of the expenditure upon what is probably the most volatile of the 15 sectors, Construction, and a slight adjustment was made in the vector on the basis of this independent calculation.

3. The development of the Federal Government purchases vector draws heavily upon the research conducted jointly by the Upper Midwest Research and Development Council and the University of Minnesota regarding the estimated impact of federal expenditures by states for 1960.[4] The research group based their total expenditure figure primarily upon estimates of the following:

   Federal Grants to State and Local Governments
   Income Disbursements to Individuals
   Payments to Members of the National Guard
   Temporary Unemployment Insurance Programs
   Military Procurement Expenditures
   Federal Interest Payments to the Public.

   To obtain an estimate[5] of the 1960 federal purchases of goods and services from the West, transfer payments, grants in aid, and interest payments were subtracted from the figure determined by the Midwest study for total federal expenditures in the five Western States. Figures for the flows of federal interest payments to the states are not included in existing data and had to be estimated.

---

[1]Tax Foundation Inc. *Facts and Figures on Government Finance, 1964-5.* New York, 1966.

[2]U.S. Department of Commerce, Office of Business Economics. *Survey of Current Business.* Washington, August 1965.

[3]U.S. Department of Commerce, Bureau of the Census. *Compendium of Government Finance, 1962.* Washington, 1963.

[4]Upper Midwest Economic Survey. *The Geographical Impact of the Federal Budget.* Technical Paper No. 3, October, 1962.

[5]Tax Foundation Inc. *Facts and Figures on Government Finance, 1960-1.* New York, 1962.

Holders of U. S. bonds were divided into four categories: 1) private individuals, 2) state and local governments, 3) insurance companies, and 4) banks, corporations, and other investors. Data pertaining to bondholdings exist by states for each of these groups. It was assumed that the share of federal interest payments of each state was proportional to the state's relative position as a bondholder. Total federal interest payments to each state were then calculated as the sum of the state's shares of the flows to the above four categories. For example, federal interest payments to individuals in each of the states were determined by the formula:

$$\left(\begin{array}{c}\text{Total federal interest payments}\\ \text{to individual bondholders}\end{array}\right) \times \left(\frac{\begin{array}{c}\text{Individual U.S. bondholdings}\\ \text{in the state}\end{array}}{\begin{array}{c}\text{Total U.S. bondholdings outside}\\ \text{the federal government}\end{array}}\right)$$

Data pertaining to individual bondholdings is supplied by the U. S. Department of Commerce.[1] Federal interest payments to individuals and total U.S. bondholdings are published by Time, Life, Inc.[2] U.S. Department of Commerce figures were adopted for state and local government holdings[3] and the Western States' shares of interest payments to this group was calculated in similar fashion. The states' shares of payments to the third category, insurance companies, was based on the shares of life insurance in force, the relevant data being obtained from the Institute of Life Insurance study.[4] The distribution of interest to the commercial banking system was based on the latter's holdings of U. S. Government obligations as revealed by the Bureau of the Budget.[5] Interest payments to corporations and other investors were essentially unknown quantities and were assumed proportional to the shares received by the banks.

Total federal purchases from the five states for the base year 1963 were then calculated on the assumption that the states' shares were the same in that year as they were for 1960. Distribution of the control total was effected in accordance with the 15-sector aggregated Federal Purchases vector from the 1958 national table.

4. The control total for the Gross Private Capital Formation vector was treated in two parts: purchase of producers' durables and expenditures upon construction activity. The former component was further subdivided into Manufacturing,

[1] U. S. Department of Commerce, Office of Business Economics. *Survey of Current Business.* Washington, July 1965.

[2] Time, Inc., *op. cit.*

[3] U. S. Department of Commerce, Bureau of the Census. *Compendium of Government Finance, 1962.* Washington, 1963.

[4] Institute of Life Insurance. *Life Insurance Fact Book, 1964.* New York, 1965.

[5] Bureau of the Budget. *98th Annual Report of the Comptroller of the Currency, 1960.* Washington, 1961.

Mining, Utilities, and Other. The five Western States' expenditures in 1963 on producers' durables in the first of these subdivisions, the Manufacturing sector, were calculated in the following manner:

$$PDM_{State} = PDM_{U.S.} \frac{KM_{State}}{KM_{U.S.}} \qquad (3)$$

where PDM = producers' durables expenditures in Manufacturing and KM = capital expenditures in Manufacturing. The figures for $PDM_{U.S.}$[1] were adjusted in the manner discussed earlier with regard to Personal Consumption in order to conform more closely with the total figure published by the Department of Commerce.[2] Data pertaining to KM_{State} and $KM_{U.S.}$ were obtained from the 1963 Census of Manufactures.[3]

Expenditures on producers' durables in the Mining sector were calculated on the same basis as were those in the Manufacturing sector above. Producers' durable expenditures in Mining were taken from data published by the Tax Foundation, Inc.[4] and the figures for capital expenditures in Mining for both the five Western States and for the U. S. were obtained from the 1963 Census of Mineral Industries.[5]

Data pertaining to capital expenditures in the Utilities sector by states were not directly available and estimates were based on the capacity added between 1960 and 1963 according to the equation:

$$PDU = PDU_{U.S.} \frac{C_{State\ (1960\text{-}1963)}}{C_{U.S.\ (1960\text{-}1963)}} \qquad (4)$$

where: PDU = producers durable expenditures in the Utility sector, and C = capacity added. The total for $PDU_{U.S.}$ in 1963 was taken from the Tax Foundation, Inc. data[4] and the "C" figures were derived from data published in the U.S. Statistical Abstract.[6]

Expenditures for producers' durables in the category of Other, which includes the commercial and the transportation sectors, were distributed on the basis of the states' shares of producers' durables expenditures in the other three categories.

---

[1]Tax Foundation Inc. *Facts and Figures on Government Finance, 1964-5.* New York, 1966.

[2]U. S. Department of Commerce, Office of Business Economics. *Survey of Current Business.* Washington, August 1965.

[3]U. S. Department of Commerce, Bureau of the Census. *Census of Manufactures, 1963.* Washington, 1964.

[4]Tax Foundation Inc., *ibid.*

[5]U. S. Department of Commerce, Bureau of the Census. *Census of Mineral Industries, 1963.* Washington, 1964.

[6]________. *U.S. Statistical Abstract, 1965.* Washington, 1966.

A breakdown by states of the second component of Private Capital Formation—Total Private Construction—had been published by the Business and Defense Services Administration annually beginning in the year 1939.[1] Publication of the series was terminated in 1952, however, and was never resumed. Consequently, several approaches to the problem of estimating the 1963 construction figures were considered.

From data supplied by the 1963 Construction Review,[2] the first procedure attempted was to apportion the U. S. total according to each state's share of permit-authorized private construction. The resulting distribution reveals little more, however, than apparent evidence that government construction agencies in the West are more stringent in their controls over construction activity or that private construction values contained in the Western permits tend to be overstated relative to those provided by agencies located in other parts of the country. (California's share of private residential and nonresidential construction calculated on this basis, for example, amounts to almost one-quarter of such construction in the U. S.)

A second method attempted was the distribution of the U. S. total on the basis of the West's share of the national wage bill in contract construction according to data published by the Bureau of Employment Security[3] and by the F. W. Dodge Corp.[4] Similarly, distribution on the basis of employment in contract construction was also undertaken from U. S. Department of Labor statistics.[5] The results in either case were little better than those obtained by calculations based on permit authorizations.

Since data regarding private construction by states were published through 1952 by the government, Western State shares were calculated for the postwar years 1946–52 and average shares for the period determined. Correlations between these postwar averages and other variables such as state personal income were constructed. Each attempt yielded results which for some states were quite obviously unreasonable.

In the final analysis, recourse was made for the first time to the information provided by various state input-output tables, and private construction totals for the year 1963 were assigned to each Western State primarily on the basis of this information and the judgment of the senior author.

[1] U. S. Department of Commerce, Business and Defense Services Administration. *Construction and Building Materials, Statistical Supplement 1953.* Washington, 1954.

[2] U.S. Department of Commerce, Business and Defense Services Administration. *Construction Review, 1963.* Washington, 1964.

[3] U.S. Department of Labor, Bureau of Employment Security. *Employment and Wages.* Washington, (issues 1951–1963).

[4] F.W. Dodge Corp. *Construction Contract Statistics, 1963.* Los Angeles, 1964.

[5] U.S. Department of Labor, Bureau of Employment Security, *ibid.*

Once figures for both producers' durables and private construction were determined, the producers' durables total was distributed in accordance with the Gross Private Fixed Capital Formation vector of the 1958 national table, taking into account the difference in the share of private construction.

Coefficients. Essentially two types of coefficients were available for adoption to the Colorado and the four-state block: a) U.S. national coefficients and b) the state survey coefficients of New Mexico, Utah, and Washington. The selection of coefficients to be applied is a choice between assumptions. If one chooses to adopt national coefficients as did the authors of the California and Oregon studies, one necessarily assumes that the unadjusted[1] input pattern of any industry in the state with regard to foreign and domestically produced goods is identical to the input pattern of domestically produced goods of that same industry on the national level. If the survey coefficients representing domestically produced commodity flow ratios (format #2) of state A are chosen to be representative of the structure of the economy of state B, one has adopted the assumption that the pattern of domestically produced inputs of any industry in state B is identical to the pattern of domestically produced inputs of the same industry in state A.

It seems reasonable to assume that the structures of the economies of Colorado and the four-state block are more likely to approximate the structures of the relatively agriculturally-oriented economies of New Mexico, Utah, and Washington than the more industrially-oriented structure of the U.S. economy. A choice was made among the three sets of coefficients by comparing the production distributions of gross output. Each element of the gross output vector of Colorado was expressed as a percentage of the total gross output. The same calculations were made for the gross output vectors of the three state tables with survey coefficients and a comparison was made. The production pattern of Colorado was found to resemble most closely that of Utah and the coefficients of that state table were adopted to represent the technological structure of the Colorado economy. In the same manner, the coefficients of New Mexico were chosen to represent the economy of the four-state block.

[1]By 'unadjusted' national coefficients it is meant here that the U.S. coefficients are unaltered by the process of subtracting net imports from the processing matrix. The Oregon coefficients are 'unadjusted'; those of the California table are 'adjusted.'

APPENDIX IV

WESTERN INTERSTATE FLOWS NET TRADE

WESTERN INTERSTATE FLOWS NET TRADE

	COMMODITY #1		COMMODITY #2		COMMODITY #3		COMMODITY #4		COMMODITY #5		COMMODITY #6	
	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET
Arizona		8.06	15.10			22.20		7.33	3.13			1.49
California		487.13	81.57		39.03			42.94		89.89		67.90
Colorado	165.81			56.42		0.15	3.46		12.94		75.19	
New Mexico	111.06		0.05		0.64		0.97		3.71		197.58	
Oregon		25.22	12.83			0.01	44.31			12.08		232.73
Utah		1.28		73.60		11.95		5.65	10.59		63.68	
Washington		58.16	16.41		5.54			0.86	2.04			8.03
Nevada	21.07			13.73		10.77		0.11	15.36		8.98	
Idaho	54.21		15.63			0.07	3.41		26.64			102.06
Montana	116.54		15.70			0.04	2.36		20.02			1.82
Wyoming	111.16			13.54		0.02	2.38		7.54		68.60	
TOTAL	579.85	579.85	157.29	157.29	45.21	45.21	56.89	56.89	101.97	101.97	414.03	414.03

	COMMODITY #7		COMMODITY #8		COMMODITY #9		COMMODITY #10		COMMODITY #11		COMMODITY #12	
	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET
Arizona	0.72			0.61		13.11		114.19	–	–		21.54
California		43.40		13.68	523.49		252.60		–	–		517.11
Colorado	8.57		74.36		1.71			114.37	–	–		39.44
New Mexico	2.93		108.44			96.78		43.33	–	–		8.79
Oregon	1.98			4.69		68.60		30.01	–	–	352.96	
Utah	2.29		17.50		25.61		242.47		–	–		35.08
Washington	6.80			270.66		220.16	96.91		–	–	279.90	
Nevada	7.71		3.01			55.63		31.14	–	–		4.56
Idaho	3.24			0.15		34.64		137.93	–	–		16.79
Montana	4.70		19.62			36.79		81.46	–	–		5.53
Wyoming	4.46		66.86			24.90		39.55	–	–		2.02
TOTAL	43.40	43.40	289.79	289.79	550.81	550.81	591.98	591.98	–	–	650.86	650.86

	COMMODITY #13		COMMODITY #14		COMMODITY #15		COMMODITY #16		COMMODITY #17		COMMODITY #18	
	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET
Arizona		2.41		68.81	108.00		85.60			15.05		110.98
California		52.18	513.48			224.58	126.67		311.29		938.67	
Colorado		1.33		66.92		14.39		68.05	5.23		12.29	
New Mexico		1.16		15.41		194.88		56.40		24.18		118.53
Oregon		4.17		116.24	59.86			61.99	0.67			33.38
Utah		0.71		7.46	193.21		50.05			47.17	176.84	
Washington		0.10		258.75		2.91		16.97		224.34		661.79
Nevada		0.13		15.66		0.09		18.23		0.29		50.98
Idaho	63.67			4.67		1.81		14.35		7.91		62.03
Montana		1.14	9.64		50.33			16.66	1.18			57.29
Wyoming		0.34	30.80			2.74		9.58	0.57			32.82
TOTAL	63.67	63.67	553.92	553.92	441.40	441.40	262.33	262.33	318.94	318.94	1127.80	1127.80

	COMMODITY #19		COMMODITY # 20		COMMODITY #21	
	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET	EXPORT NET	IMPORT NET
Arizona		388.46		67.81		0.48
California	2812.10		481.64			348.04
Colorado		322.48		151.09		42.46
New Mexico		253.42		113.80		61.35
Oregon		268.57		91.50		9.37
Utah		275.10		14.45	84.57	
Washington		850.28	117.94		391.35	
Nevada		30.47		91.74		2.19
Idaho		154.70		14.27		4.76
Montana		155.83		14.27		4.52
Wyoming		112.79		40.65		2.95
TOTAL	2812.10	2812.10	599.58	599.58	475.92	475.92

APPENDIX V

COMPUTER PROGRAMS

PROGRAM FLOW 500

Purpose: To prepare an interindustry flow table for a region which will show transactions that are "net" in terms of regionally produced products, i.e., the flows are net of imports.

Inputs: The program requires as inputs a national (or other) transactions matrix (an option provides that the matrix can be tape input), a normalizing vector (vector of gross outputs which conforms to the national flow matrix), regional final demand vectors, regional gross output vector, value added vector, and a final demand value added vector. Where certain productive sectors are nonexistent for the region, but present in the transactions matrix used as basic input (above), a program option provides for deleting these sectors from the calculation and treating them exogenously.

Computers: The program was designed to run on CDC 6400 or 6600 machines in Fortran IV or Extended Fortran, using five scratch disks. One tape is required if the user desires the regional flow matrix prepared by the program to be available in tape input form for further operations; one additional tape is required if the user has his input matrix on tape.

Restrictions:

1. The order of the input matrix vs. the user's core may be a restriction. Read carefully the description of ISIZE under the control card description.
2. Maximum order of the national, or other flow matrix, used as input is 500.
3. Maximum number of regional final demand vectors which may be input is 10.
4. Each element of the normalizing vector must be nonzero.
5. The maximum number of deleted sectors must be less than the order of the basic input matrix, and sector numbers must be in ascending order.

All of these restrictions, except the first one, produce error messages if violated and the program run will terminate. NOTE: There is a further "restriction" as far as CP time and efficiencv. If the order on the input matrix is less than or equal to 120, the user should utilize the program FLOW. FLOW is the same as FLOW 500 but for matrices less than or equal to 120 and does not entail the large number of tape manipulations as does FLOW 500.

Method: The program reads two title cards, control card, and sector "delete" card(s) (if deletions are specified) along with the associated data inputs specified above. Machine calculations are undertaken as given in detail below with resulting regional table DF (showing domestic flows) being printed and punched, or placed on tape for further use.

Title Cards: The two title cards noted in the foregoing section can contain any alphanumeric information in columns 1-72 for user's convenience, i.e., designating printout sheets and input data used. The cards may be left blank but must be present as part of the overall input sequence.

Control Card: The control card provides for options through a choice of six variables in the order given below. The card is in a 6I3, I6 format. All variables must be punched and all must be right adjusted within their column fields.

1. N – order of national (or other) input matrix. (Col. 1-3)
2. K – number of regional final demand vectors used as inputs. (Col. 4-6)
3. M – number of productive sectors to be deleted from basic input matrix to make it conformable with the productive sectors as specified for the regions. (Col. 7-9)
4. ISEE a 1 punched in Col. 12 will result in a "debug" printout which can be used for analytical purposes if the program run fails. If a zero is entered no "debug" print will be given.
5. IDFOUT a 0, 1, 2, 3, or 4 punched in column 15 governs the form of output of the regional (domestic) flow matrix as follows:

   0 = No DF output is given

   1 = Punch DF on cards columnwise in 7F10.2 format

   2 = DF matrix, put on tape columnwise in 7F10.2 format

   3 = Punch uncondensed (with zero rows and columns for deleted sectors) DF columnwise in 7F10.2 format

   4 = Tape uncondensed (with zero rows and columns for deleted sectors) DF columnwise in 7F10.2 format

6. MATIN matrix input media. A 1, 2, 3, or 4 punched in column 18 will provide for the following mode of input:

   1 = Binary tape with vectors of matrix columnwise

   2 = BCD tape, columnwise

   3 = BCD tape, rowwise

   4 = Cards, columnwise

7. ISIZE, punched in columns 19-24, is a variable equal to the array size given in the Fortran statement DIMENSION A ( ) – enter actual number in parentheses. In the main program of FLOW500, A is given as a vector although it is carried as a variable-dimensioned matrix in the subprograms. This allows for a dynamic core arrangement as obviously less core (less $ cost to the user) is needed for a 200-order than a 400-order matrix. The arrangement of A is that ISIZE/N columns are carried in core at any given time. ISIZE is limited by the core space available (dimension of A).

   To calculate ISIZE: (using definitions of N, K, M, above)

   a. NK = N + K — maximum size of DF matrix

      NPC = N – M + K + 5 — maximum # of columns to print

      NPR = N + 7 — maximum # of rows to print

      During calculations, the maximum matrix held in core is N – K; however, the final matrix to be printed is assembled in core as a unit.

      NSR = maximum (NK, NPR)

      NSC = maximum (NK, NPC)

      Therefore, NSR + NSC = matrix size needed.

   b. The FLOW500 program minus the dimension of A, uses (rounded up) $60{,}000_8$ core. The core space available (CS) minus program space (PS) equals amount of core available for a (MS) or ISIZE = CS – PS. Now if ISIZE is greater than or equal to MS, then the control card and the dimension A in the main program = MS. If ISIZE is less than MS, then CS = actual core space and ISIZE, and dimension A in the main program = calculated ISIZE (= CS – PS).

c. Now that ISIZE and CS have been calculated, three things need to be done:

   (1) Set dimension A in the program equal to ISIZE.

   (2) Put calculated ISIZE on control card in columns 19-24.

   (3) Put CS on job card where core size is to be specified.

NOTE TO USER: The use of ISIZE appears complex but after using it a time or two, it will become straightforward. The advantages of a dynamic core arrangement more than compensate for the time to calculate ISIZE, amount of core available; i.e., in cost, faster turn-around, efficient use of core, and CS.

Sector Delete Cards: These card(s) must be punched in 24I3 Format and will be present only if columns 7-9 of the Control Card are not 0 as specified above. The sectors to be deleted must be punched in ascending numerical order and must be right adjusted.

Tape Detail: If MATIN control option is 1, 2, or 3, tape identification for tape request is tape 4 for inputing flow matrix on tape. If IDFOUT option is 2 or 4, tape identification for tape request is Tape 8 for puting punch output on tape.

Input Data Detail: All following input data are in 7F10.2 Format.

1. Regional Gross Output vector (X) of order N. If the Sector Deletion option is chosen, the corresponding elements of X vector must be equal to zero.
2. Normalizing Vector (X) of order N, i.e., vector of national gross outputs.
3. National interindustry transactions, or flow, matrix (A) of order N. (If the Sector Deletion option has been selected, the corresponding columns of the A matrix will be set to zero by the program.)
4. Regional Final Demand vectors (Y) of order N, number of vectors = K. Max. K = 10.
5. National, or other value added transactions vector (VA) of order N.
6. Regional Final Demand value added transactions (FDVA) of order K.
7. National or other, foreign import transactions (FDFI) of order K.

```
FLOW500,5,150,100000.800279,CARASSO COMPILE ONLY
RUN,S.
7
      PROGRAM FLOW500(INPUT,OUTPUT,PUNCH,TAPE1=1001,TAPE2=1001,TAPE3=100
     11,TAPE5=1001,TAPE4,TAPE7=1001,TAPE8)
C     WHICH WILL SHOW TRANSACTIONS THAT ARE NET IN TERMS OF REGIONALLY
C     PRODUCED PRODUCTS,I.E., THE FLOWS ARE NET OF IMPORTS
C     INPUTS
C     THE  PROGRAM REQUIRES AS INPUTS A NATIONAL (OR OTHER) TRANSACTIONS
C     MATRIX (AN OPTION PROVIDES THAT THE MATRIX CAN BE TAPE INPUT), A
C     PROGRAM FLOW PREPARES AN INTERINDUSTRY FLOW TABLE FOR A REGION
C     NORMALIZING VECTOR (VECTOR OF GROSS OUTPUTS WHICH CONFORMS TO THE
C     NATIONAL FLOW MATRIX), REGIONAL FINAL DEMAND VECTORS, REGIONAL
C     GROSS OUTPUT VECTOR, VALUE ADDED VECTOR, FINAL DEMAND VALUE ADDED
C     VECTOR, FOREIGN IMPORT VECTOR, FINAL DEMAND FOREIGN IMPORT VECTOR.
C     THE A MATRIX, VALUE ADDED VECTOR, AND FOREIGN IMPORT VECTOR MAY
C     BE IN A COEFFICIENT FORM, IN WHICH CASE THE NORMALIZING VECTOR
C     MUST BE INPUT AS A VECTOR OF 1.S. (SUM VECTOR) OF ORDER N.
C     WHERE CERTAIN PRODUCTIVE SECTORS ARE NONEXISTENT FOR THE REGION,
C     BUT PRESENT IN THE TRANSACTIONS MATRIX USED AS BASIC INPUT (ABOVE)
C     A PROGRAM OPTION PROVIDES FOR DELETING THESE SECTORS FROM THE
C     CALCULATION AND TREATING THEM EXOGENOUSLY.
      DIMENSION TITLE(36)
      DIMENSION A(15000)
      DIMENSION IM(510)
      COMMON ISEE,IP,M,K
      COMMON MATIN,IDFOUT,ISIZE,TITLE
      COMMON IM
C     TAPES---1=A,ABAR,DELETED ROWS COL-WISE   ------
C             2=A,(I-A)*VXR,RF                 --
C             3=Y VECTORS DELETED ROWS-ROWWISE,RIM,BIG PRINTOUT---
C             4=ORIGINAL                       --
C             5=ARF ROW-WISE,DF                ------
C             7=USED FOR COLUMN TO ROW SWITICHING WITHIN PROGRAM---SCRAT
C             8=DOMESTIC  FLOW MATRIX TAPE OUTPUT
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      REWIND 5
C     READ 2 TITLE CARDS
      READ 60, (TITLE(I),I=1,24)
      PRINT 70, (TITLE(I),I=1,24)
C     READ CONTROL CARD
      READ 90, N,K,M,ISEE,IDFOUT,MATIN,ISIZE
      NK=N+K
      NPC=N-M+K+5
      NSC=MAX0(NK,NPC)
      NPR=N+7
      NSR=MAX0(NK,NPR)
      II=ISIZE/NSC
      IP=(II/8)*8
      IF (II.LE.NSC) GO TO 10
      II=NSC
      IP=NSC
10    CONTINUE
      PRINT 80, N,K,M,ISEE,IDFOUT,MATIN,ISIZE,II,IP,NK,NSR,NSC,NSR,II
C     N=ORDER OF MATRIX
C     K=NUMBER OF REGIONAL FINAL DEMANDS
C     M=NUMBER OF ROWS TO BE DELETED
C     ISEE=0 NO DEBUG PRINTOUT, ISEE=1 DEBUG PRINTOUT
C     IDFOUT=0 BYPASS OPTION
C           =1 CARD PUNCH OUTPUT OF PRINTED FINAL DF MATRIX
C           =2 MAG TAPE OUTPUT OF PRINTED FINAL DF MATRIX
C           =3 CARD PUNCH OUTPUT OF UNCONDENSED DF MATRIX **
C           =4 MAG TAPE OUTPUT OF UNCONDENSED DF MATRIX **
```

```
C     ** UNCONDENSED MEAANS WITH ZERO ROWS AND COLUMNS STILL IN THE
C        MATRIX - ASSUMING THAT M GT 0,ELSE 1=3 AND 2=4 OPTIONS IF M=0
C      MATIN=TAPE (1,2,3) OR CARD (4) INPUT
C           =1=BIN COLUMNWISE
C           =2=BCD COLUMNWISE
C           =3=BCD ROWWISE
C           =4=CARDS COLUMNWISE
C          *REGARDLESS OF WHERE/HOW IT IS, THE PROGRAM WILL PUT
C           MATRIX ON TAPE1-BIN-COL
C      ISIZE=DIMENSION OF A - CURRENT LRL LIMIT FOR REGULAR OVER THE
C      COUNTER JOBS YIELDS A LIMIT OF 44100 (SEE WRITEUP)
C      II= ALLOWABLE COLUMN SIZE OF A (N+K COLUMNS AT THE PEAK)
C      ONLY MANIPULATE II COLUMNS AT A TIME (BLOCKS OF N,II)
C      IP=CLOSEST MULTIPLE OF II/8 FOR EASE OF PRINTING
       IF ((M.GE.N).OR.(N.GT.500).OR.(K.GT.10).OR.(MATIN.GT.4).OR.(MATIN.
     1LE.0).OR.(IDFOUT.GT.4).OR.(IDFOUT.LT.0).OR.(ISEE.GT.1).OR.(ISEE.LT
     2.0)) GO TO 20
       GO TO 30
20     PRINT 40
       STOP
30     CONTINUE
       CALL FLOW (A,N,II,NSR)
       PRINT 50
       STOP
C
40     FORMAT (*CONTROLCARDERROR*)
50     FORMAT (*EOJFLOW*)
60     FORMAT (12A6)
70     FORMAT (1H1,12A6/12A6)
80     FORMAT (*CONTROLCARD*,6I4,I6/*COMPUTEDVARS*,5I4,*A(*,I4,*,*,I3,*)*
     1)
90     FORMAT (6I3,I6)
       END

       SUBROUTINE FLOW (A,N,II,NSR)
       DIMENSION A(NSR,II)
       DIMENSION Y(510), V(510), W(510), IM(510), VX(510), VXR(510), CE(5
     110), CI(510), IO(510), ID(510), TF(510), TSI(510), SI(510), VA(510
     2), ADJ(510), FI(510), FOVA(10), FOFI(10), FDS(10), TITLE(36)
       REAL IO,ID
       COMMON ISEE,IP,M,K
       COMMON MATIN,IDFOUT,ISIZE,TITLE
       COMMON IM
C      HOUSEKEEPING BEGINS TO CLEAR CORE
       DO 10 I=1,510
       Y(I)=0.
       V(I)=0.
       W(I)=0.
       IM(I)=0.
       VX(I)=0.
       CE(I)=0.
       CI(I)=0.
       IO(I)=0.
       IO(I)=0.
       SI(I)=0.
       FI(I)=0.
       VA(I)=0.
       TF(I)=0.
       ADJ(I)=0.
       TSI(I)=0.
       VXR(I)=0.
10     CONTINUF
       DO 20 I=1,10
       FOVA(I)=0.
       FOFI(I)=0.
       FDS(I)=0.
```

```
20    CONTINUE
      NK=N+K
      N1=N+1
      NM=N-M
      NKM=NK-M
      NM1=NM+1
      DO 30 I=1,500
30    IM(I)=0
      IF (M.EQ.0) GO TO 60
      READ 850, (IM(I),I=1,M)
      IF (M.LE.1) GO TO 150
      DO 40 I=2,M
      IF (IM(I).LE.IM(I-1)) GO TO 50
40    CONTINUE
      GO TO 60
50    II=I-1
      PRINT 830, II,I
      CALL EXIT
      STOP
C     DELETED ROWS MUST BE IN SEQUENCE
C     IF NOT -- END OF RUN
60    CONTINUE
C     READ VXR
      READ 860, (VXR(I),I=1,N)
      CALL SV (N,VXR,20H VXR READ IN         )
C     READ IN VX,CHECK FOR ZERO ELEMENTS=ERROR AS VX IS A DIVISOR
      READ 860, (VX(I),I=1,N)
      CALL SV (N,VX,20H VX READ IN          )
      DO 70 I=1,N
      IF (VX(I).EQ.0.) GO TO 80
70    VX(I)=1/VX(I)
      GO TO 90
80    PRINT 840
C     VX INVERTED TO NORMALIZE
90    CALL SV (N,VX,20H VX INVERTED         )
C     BRING IN MATRIX A -- ASSEMBLE BIN-COL-TAPE1
      GO TO (100,120,120,140), MATIN
C     TRANSFER DATA TO TAPE1 -SAVE TAPE4
100   DO 110 J=1,N
      READ (4) (V(I),I=1,N)
      WRITE (1) (V(I),I=1,N)
110   CONTINUE
      GO TO 160
C     BCD TAPE COL OR ROW--IF ROW GO RTOC AFTER BCD TO BIN
120   DO 130 J=1,N
      READ (4,860) (V(I),I=1,N)
      WRITE (1) (V(I),I=1,N)
130   CONTINUE
      IF (MATIN.EQ.3) CALL CTOR (1,2,1,N,N,A,NSR,II)
      GO TO 160
C     CARDS IN
140   DO 150 J=1,N
      READ 860, (V(I),I=1,N)
      WRITE (1) (V(I),I=1,N)
150   CONTINUE
160   REWIND 1
C     BEGIN PROGRAM WITH NATIONAL FLOW MATRIX COL ON TAPE 2 (BINARY)
C     SET DELETED COLS TO ZERO OF A MATRIX
C     SETUP  DUMMY 0 COL IN W
      DO 170 I=1,N
170   W(I)=0.
      L=1
      DO 190 J=1,N
      READ (1) (V(I),I=1,N)
      IF (IM(L).NE.J) GO TO 180
```

```
      WRITE (2) (W(I),I=1,N)
      L=L+1
      GO TO 190
180   WRITE (2) (V(I),I=1,N)
190   CONTINUE
      CALL SM (N,N,2,20H A MATRIX READ IN   ,A,NSR,II)
      REWIND 1
      REWIND 2
C     NORMALIZE A  ABAR=A*VX
      CALL MULT (2,1,II,N,VX,VX,A,V,0,NSR)
C     NORMALIZED A ON TAPE 1
      CALL SM (N,N,1,20H DIRECT COEF MATRIX ,A,NSR,II)
C     FORM VNETB IN SEVERAL STEPS  -- N=(I-ABAR)*VXR - SUM(Y)
C     FIRST DO I-A*VXR
      DO 200 I=1,N
200   V(I)=0.
      ME=MINO(N,II)
      MB=1
      MCT=0
210   LL=ME-MB+1
      DO 220 J=1,LL
220   READ (1) (A(I,J),I=1,N)
      DO 260 J=1,LL
      MCT=MCT+1
      DO 250 I=1,N
      IF (I.EQ.MCT) GO TO 230
      H=-(A(I,J))
      GO TO 240
230   H=1-(A(I,J))
240   V(I)=V(I)+(H*VXR(MCT))
250   CONTINUE
260   CONTINUE
      IF (ME.EQ.N) GO TO 270
      MB=MB+II
      ME=MINO(N,ME+II)
      GO TO 210
270   CONTINUE
      CALL SV (N,V,20H  CALC I-A*VXR       )
      DO 280 I=1,N
280   W(I)=0.
C     INPUT Y VECTORS SUM AS GO INTO W  AND PRINT TOO
      DO 300 J=1,K
      READ 860, (Y(I),I=1,N)
      CALL SV (N,Y,20H Y VECTOR  READ IN  )
      DO 290 I=1,N
290   W(I)=W(I)+Y(I)
      WRITE (3) (Y(I),I=1,N)
300   CONTINUE
      CALL SV (N,W,20H VY AGG ROW SUM VY  )
C     COMPUTE VNETB=V-W=I-A*VXR-SUM Y VECTORS
      DO 310 I=1,N
      V(I)=V(I)-W(I)
310   CONTINUE
      CALL SV (N,V,20H VNETB              )
C     GET CE=ALL POS  CI=ALL NEG   ELEMENTS OF VNETB
      REWIND 3
      DO 340 I=1,N
      IF (V(I)) 320,330,330
320   CI(I)=V(I)
      CE(I)=0.
      GO TO 340
330   CI(I)=0.
      CE(I)=V(I)
340   CONTINUE
      CALL SV (N,CE,20H CE (ALL POS/0)     )
```

```
      CALL SV (N,CI,20H CI (ALL NEG/0)      )
C     FORM  RF  REGIONAL FLOW MATRIX=ABAR*VXR
      CALL MULT (1,2,II,N,VXR,VXR,A,V,0,NSR)
C     V=ROW SUM OF A=RF  W=SUM OF Y VEC --SUM=P
      DO 350 I=1,N
350   V(I)=V(I)+W(I)
      CALL SV (N,V,20H ROWSUM AUG R.F.M    )
      REWIND 2
      REWIND 3
C     APEND Y VECTORS TO RF=ARF
C     SPACED TO RIGHT SPOT ADD VECTORS
      DO 360 I=1,N
360   READ (2)
      DO 370 I=1,K
      READ (3) (Y(J),J=1,N)
      WRITE (2) (Y(J),J=1,N)
370   CONTINUE
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 7
      CALL SM (N,NK,2,20H AUG REG FLOW MATRIX,A,NSR,II)
C     FORM RIM ON T3 RIM= ARF/P  * CI --P IS IN V,ARF OF T2
C     COL SUM FORMED IN SI
      CALL MULT (2,3,II,N,CI,V,A,SI,1,NSR)
      CALL SM (N,NK,3,20H RIM AS FORMED BY M ,A,NSR,II)
      CALL SV (NK,SI,20H SI BEFORE SUBTRAC  )
C     CE,CI,VX,VXR ARE DELETED HERE AS PREVIOUS NEED WHOLE VECTOR
C     AND FOLLOWING NEED ONLY THE DELETED PORTION
      CALL DELETE (CE,N)
      CALL DELETE (VX,N)
      CALL DELETE (VXR,N)
      CALL DELETE (CI,N)
C     PULL DELETED ROWS FROM ARF-SUMMING INTO TSI
C                                     COLLECTED 2-PART ROW SUMS
C                                     DELETING ROWS
C                                     PUTON T1
      DO 380 I=1,NK
380   TSI(I)=0.
      IF (M.EQ.0) GO TO 430
      NM3=NM+3
C     ABOVE=STARTING LOC IN COL VECTOR OF DEL ROW SUMS
C     GO COL-TO-ROW ARF
      CALL CTOR (2,5,0,N,NK,A,NSR,II)
      L=1
      DO 420 J=1,N
      READ (5) (Y(I),I=1,NK)
      IF (IM(L).NE.J) GO TO 420
      L=L+1
      DO 390 I=1,NK
390   TSI(I)=TSI(I)+Y(I)
      SN=0.
      SNK=0.
      DO 400 I=1,N
400   SN=SN+Y(I)
      ID(NM3)=SN
      DO 410 I=N1,NK
410   SNK=SNK+Y(I)
      TF(NM3)=SNK
      VXR(NM3)=SN+SNK
      CI(NM3)=SN+SNK
      NM3=NM3+1
      CALL DELETE (Y,NK)
      WRITE (7) (Y(I),I=1,NKM)
```

```
420   CONTINUE
C     GO ROW-COL OF DEL ROWS--NEED THEM COLWISE TO PRINT
      CALL CTOR (7,1,0,NKM,M,A,NSR,II)
      REWIND 1
      REWIND 2
      REWIND 7
430   CONTINUE
      CALL SV (NK,TSI,20H COL SUM DELETED ROW)
C     FORM SI=ABS(SI)-TSI(COL SUM DEL ROWS)
      DO 440 I=1,NK
440   SI(I)=ABS(SI(I))-TSI(I)
      CALL SV (NK,SI,20H SECTORAL IMPORT VEC)
      CALL DELETE (SI,NK)
C     THE BIG THING--BAG UP DF
C     THE DELETED COL SHOULD=0 SO DELETE HERE
C                       AND DELETE ROWS AS GO
C     DF=ARF + RIM    T5= T3 + T2
      REWIND 2
      REWIND 3
      REWIND 5
      IF (IDFOUT.EQ.4) REWIND 8
      ISEQ=0
      IDGO=IDFOUT+1
C     THIS AREA IS DEALING WITH UNCONDENSED MATR DF
      LK=1
      DO 560 J=1,NK
      READ (2) (V(I),I=1,N)
      READ (3) (W(I),I=1,N)
      IF (IM(LK).NE.J) GO TO 490
      LK=LK+1
C     THIS WILL PUNCH ZERO COLUMN ON DELETIONS
C     ROWS ARE ALREADY ZEROED AT DELETIONS
      GO TO (560,560,560,450,470), IDGO
450   DO 460 I=1,N,7
      ISEQ=ISEQ+1
      PUNCH 880, (ADJ(I),ADJ(I+1),ADJ(I+2),ADJ(I+3),ADJ(I+4),ADJ(I+5),AD
     1J(I+6),J,ISEQ)
460   CONTINUE
      GO TO 560
470   DO 480 I=1,N,7
      WRITE (8,880) (ADJ(I),ADJ(I+1),ADJ(I+2),ADJ(I+3),ADJ(I+4),ADJ(I+5)
     1,ADJ(I+6),J,ISEQ)
480   CONTINUE
      GO TO 560
490   CONTINUE
      DO 500 L=1,N
      V(L)=V(L)+W(L)
500   CONTINUE
C     PUNCH OR TAPE FULL COLUMN
      GO TO (550,550,550,510,530), IDGO
510   DO 520 I=1,N,7
      ISEQ=ISEQ+1
      PUNCH 880, (V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6),J,ISEQ)
520   CONTINUE
      GO TO 550
530   DO 540 I=1,N,7
      ISEQ=ISEQ+1
      WRITE (8,880) (V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6),J,IS
     1EQ)
540   CONTINUE
550   CALL DELETE (V,N)
      WRITE (5) (V(I),I=1,NM)
560   CONTINUE
      REWIND 2
```

```
      REWIND 3
      REWIND 5
      GO TO (630,570,600,630,630), IDGO
  570 00 590 J=1,NKM
      READ (5) (V(I),I=1,NM)
      00 580 I=1,NM,7
      ISEQ=ISEQ+1
      PUNCH 880, (V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6),J,ISEQ)
  580 CONTINUE
  590 CONTINUE
      REWIND 5
      GO TO 630
  600 REWIND 8
      DO 620 J=1,NKM
      READ (5) (V(I),I=1,NM)
      DO 610 I=1,NM,7
      ISEQ=ISEQ+1
      WRITE (8,880) (V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6),J,IS
     1EQ)
  610 CONTINUE
  620 CONTINUE
      REWIND 8
      REWIND 5
  630 CONTINUE
      CALL SM (NM,NKM,5,20H DOMESTIC FLOW MAT   ,A,NSR,II)
      DO 640 I=1,NKM
  640 IO(I)=0.
      DO 650 I=1,NM
      IO(I)=0.
  650 TF(I)=0.
      DO 680 J=1,NKM
      READ (5) (V(I),I=1,NM)
      DO 660 I=1,NM
      IO(J)=IO(J)+V(I)
      IF (J.GT.NM) GO TO 660
      IO(I)=ID(I)+V(I)
  660 CONTINUE
      IF (J.LE.NM) GO TO 680
      00 670 I=1,NM
  670 TF(I)=TF(I)+V(I)
  680 CONTINUE
      DO 690 I=1,NM
      TF(I)=TF(I)+CE(I)
  690 CONTINUE
      CALL SV (NM,ID,20H IO                 )
      CALL SV (NKM,IO,20H IO                 )
      CALL SV (NM,TF,20H TF                 )
C     REAO IN VA, CALC VAPRIME
      READ 860, (VA(I),I=1,N)
      CALL SV (N,VA,20H V.A. VECTOR REAO IN)
      CALL DELETE (VA,N)
      00 700 I=1,NM
  700 VA(I)=(VA(I)*VX(I))*VXR(I)
      CALL SV (NM,VA,20H VA PRIME CALCULATEO)
C     REAO IN FOVA--USE AS IS
      REAO 860, (FOVA(I),I=1,K)
      CALL SV (K,FOVA,20H REG.FOVA REAO IN    )
C     REAO IN FI,CALC FIPRIME
      REAO 860, (FI(I),I=1,N)
      CALL SV (N,FI,20H F.I. REAO IN        )
      CALL OELETE (FI,N)
      00 710 I=1,NM
  710 FI(I)=(FI(I)*VX(I))*VXR(I)
      CALL SV (NM,FI,20H F.I.PRIME CALC.     )
C     REAO,STORE FOFI AS IS
```

```
      READ 860, (FDFI(I),I=1,K)
      CALL SV (K,FDFI,20H FDFI READ IN        )
C     COMPUTE TSI - FIRST DELETE TSI,CALC IN 2 PIECES
      CALL DELETE (TSI,NK)
      DO 720 I=1,NM
720   TSI(I)=TSI(I)+FI(I)+SI(I)
      J=0
      DO 730 I=NM1,NKM
      J=J+1
730   TSI(I)=TSI(I)+FDFI(J)+SI(I)
      CALL SV (NKM,TSI,20H TSI COMPUTED       )
C     CALC FDS
      J=NM1
      DO 740 I=1,K
      FDS(I)=TSI(J)+IO(J)+FDVA(I)
740   J=J+1
      CALL SV (K,FDS,20H FDS COMPUTED       )
C     CALC ADJUSTMENT
      DO 750 I=1,NM
750   ADJ(I)=VXR(I)-(IO(I)+TSI(I)+VA(I))
      CALL SV (NM,ADJ,20H ADJUSTMENT VECTOR  )
C     DO SUMS
      S1=0.
      S2=0.
      S3=0.
      S4=0.
      S5=0.
      S10=0.
      S11=0.
      S12=0.
      S13=0.
      S14=0.
      S15=0.
      S16=0.
      S17=0.
      SCE=0.
      SXR=0.
      SIO=0.
      SSI=0.
      SFI=0.
      STS=0.
      SVA=0.
C     SUMS OVER NM=N-M
      DO 760 I=1,NM
      S1=S1+TF(I)
      SCE=SCE+CE(I)
      S10=S10+FI(I)
      STS=STS+TSI(I)
      SVA=SVA+VA(I)
      SSI=SSI+SI(I)
      S2=S2+VXR(I)
      S3=S3+CI(I)
      SIO=SIO+IO(I)
760   CONTINUE
C     SUMS OVER K
      DO 770 I=1,K
      SFI=SFI+FDFI(I)
      S16=S16+FDS(I)
      S14=S14+FDVA(I)
      J=NM+I
      S4=S4+SI(J)
      S12=S12+TSI(J)
770   CONTINUE
C     SUMS OVER NK
      DO 780 I=1,NKM
```

```
      S13=S13+TSI(I)
      S5=S5+SI(I)
780   CONTINUE
      SXR=S2
      S11=SFI+S10
      S15=SVA+S14
      S16=S16+SCE
      S17=SXR+S16
C     PUT SUMS IN COL VECTOR SLOT
      TF(NM+1)=S1
      TF(NM+2)=S4
      TF(NM+M+3)=SFI
      TF(NM+M+4)=S12
      TF(NM+M+5)=S14
      TF(NM+M+6)=0.
      TF(NM+M+7)=S16
      VXR(NM+1)=S2
      VXR(NM+2)=S5
      VXR(NM+M+3)=S11
      VXR(NM+M+4)=S13
      VXR(NM+M+5)=S15
      VXR(NM+M+6)=0.
      VXR(NM+M+7)=S17
      CI(NM+1)=S3
      CI(NM+2)=0.
      CI(NM+M+3)=S11
      CI(NM+M+4)=S13
      CI(NM+M+5)=0.
      CI(NM+M+6)=0.
      CI(NM+M+7)=0.
      ID(NM+1)=SIO
      IO(NM+2)=SSI
      ID(NM+M+3)=S10
      ID(NM+M+4)=STS
      ID(NM+M+5)=SVA
      ID(NM+M+6)=0.
      ID(NM+M+7)=SXR
      CE(NM+1)=SCE
      CE(NM+2)=0.
      NMM=NM+2
      DO 790 I=1,M
      CE(NMM+I)=0.
790   CONTINUE
      CE(NM+M+3)=0.
      CE(NM+M+4)=0.
      CE(NM+M+5)=0.
      CE(NM+M+6)=0.
      CE(NM+M+7)=SCE
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 5
      NMM=NM+M+2
      NM3=NM+3
      NM3M=NM3+M-1
      NMT=NM+7+M
C     SETUP TO PRINT STUFF
C     PUT OUT ONE COL AT A TIME AS IT IS TO BE PRINTED -- TAPE3
C     DF IS ON TAPE5, DEL ROWS COL-WISE ON TAPE1,REST STUFF IN CORE
C     READ COL DF ONTO V,APEND STUFF AND OUT
      DO 800 J=1,NM
      READ (5) (V(I),I=1,NM)
      V(NM+1)=IO(J)
      V(NM+2)=SI(J)
      V(NMM+1)=FI(J)
```

```
      V(NMM+2)=TSI(J)
      V(NMM+3)=VA(J)
      V(NMM+4)=ADJ(J)
      V(NMM+5)=VXR(J)
      IF (M.NE.O) READ (1) (V(I),I=NM3,NM3M)
      WRITE (3) (V(I),I=1,NMT)
800   CONTINUE
C     ID COL OUT
      WRITE (3) (ID(I),I=1,NMT)
C     NOW K PART OF DF AND VECTORS
      L=1
      DO 810 J=NM1,NKM
      READ (5) (V(I),I=1,NM)
      V(NM+1)=ID(J)
      V(NM+2)=SI(J)
      V(NMM+1)=FDFI(L)
      V(NMM+2)=TSI(J)
      V(NMM+3)=FDVA(L)
      V(NMM+4)=0.
      V(NMM+5)=FDS(L)
      L=L+1
      IF (M.NE.O) READ (1) (V(I),I=NM3,NM3M)
      WRITE (3) (V(I),I=1,NMT)
810   CONTINUE
      WRITE (3) (CE(I),I=1,NMT)
      WRITE (3) (TF(I),I=1,NMT)
      WRITE (3) (VXR(I),I=1,NMT)
      WRITE (3) (CI(I),I=1,NMT)
      ISEE=2
C     PRINT FINAL OUTPUT MATRIX
      CALL SM (NMT,NKM+5,3,TITLE,A,NSR,II)
C     PUNCH CE SEQUENCED 9999 IN COL 77-80
      IY=9999
      PUNCH 870, (CE(I),CE(I+1),CE(I+2),CE(I+3),CE(I+4),CE(I+5),CE(I+6),
     1IY,I=1,NM,7)
C     PUNCH VXR SEQUENCED 8888 IN COL 77-80
      IY=8888
      PUNCH 870, (VXR(I),VXR(I+1),VXR(I+2),VXR(I+3),VXR(I+4),VXR(I+5),VX
     1R(I+6),IY,I=1,NM,7)
      PRINT 820
C
C
C
820   FORMAT (*1EOJFLOW500*)
830   FORMAT (41HO DELETIONS NOT IN SEQUENCE --- CHECK THE,I4,4H AND,I4,
     118H NUMBERS,FIX,RERUN)
840   FORMAT (*NORMALIZINGVECTORCONTAINSAZERO,RUNTERMINATED*)
850   FORMAT (24I3)
860   FORMAT (7F10.2)
870   FORMAT (7F10.2,4X,I6)
880   FORMAT (7F10.2,I3,1X,I6)
      END
      SUBROUTINE MULT (IN,IOUT,II,N,VM,VD,A,RCS,IDIV,NSR)
      DIMENSION A(NSR,II)
      DIMENSION VM(510), VD(510), RCS(510)
      COMMON ISEE,IP,M,K
C     MULTIPLY MAT ON TAPE IN * VM (IF IDIV=1 THEN MAT/VD * VM)
C     PUT ANSWER ON IOUT TAPE--ROW SUM PLAIN,COL SUM DIV INTO RCS
      REWIND IN
      REWIND IOUT
      NK=N+K
      LL=0
      IF (IDIV.EQ.1) NN=NK
      IF (IDIV.EQ.0) NN=N
      DO 10 I=1,NN
```

```
10    RCS(I)=0.
      ME=MINO(II,NN)
      MB=1
20    L=ME-MB+1
      DO 30 J=1,L
C     GET IN M COL ON N ROWS EACH--MAX=100
      READ (IN) (A(I,J),I=1,N)
30    CONTINUE
      IF (IDIV.EQ.1) GO TO 60
C     PLAIN MULTIPLY
      DO 50 J=1,L
      LL=LL+1
      DO 40 I=1,N
      A(I,J)=A(I,J)*VM(LL)
C     ROW SUM
40    RCS(I)=RCS(I)+A(I,J)
50    CONTINUE
      GO TO 90
C     DIVIDE AND MULTIPLY
60    DO 80 J=1,L
      LL=LL+1
      DO 70 I=1,N
      A(I,J)=(A(I,J)/VD(I))*VM(I)
C     COL SUM
70    RCS(LL)=RCS(LL)+A(I,J)
80    CONTINUE
C     WRITE OUT BATCH
90    DO 100 J=1,L
      WRITE (IOUT) (A(I,J),I=1,N)
100   CONTINUE
C     RESET AND CHECK LIMITS
      IF (ME.EQ.NN) GO TO 110
      MB=MB+II
      ME=MINO(NN,ME+II)
      GO TO 20
110   REWIND IN
      REWIND IOUT
      RETURN
      END
      SUBROUTINE SV (ILONG,VECTOR,TITLE)
      DIMENSION VECTOR(ILONG), TITLE(2)
C     PRINT VECTOR OF LENGTH ILONG HEADED BY TITLE
      COMMON ISEE
      IF (ISEE.EQ.0) RETURN
      PRINT 40
      PRINT 10, (TITLE(I),I=1,2)
      PRINT 20, (VECTOR(I),I=1,ILONG)
      PRINT 30
      RETURN
C
C
C
10    FORMAT (**,2A10)
20    FORMAT (**,10(E12.3,**))
30    FORMAT (**)
40    FORMAT (*-------------------*)
      END
      SUBROUTINE SM (NR,NC,NTAP,TITLE,A,NSR,II)
C     PRINTS(NR,NC) MATRIX ON TAPE NTAP HEADED BY TITLE
C     WITH 52 LINES (ROWS/PAGE)  IN IP BATCHES
      DIMENSION TITLE(2)
      DIMENSION A(NSR,II)
      COMMON ISEE,IP,M,K
      IF (ISEE.EQ.0) RETURN
      REWIND NTAP
      IPAGE=1
```

```
      JB=0
      LC=0
      KC=0
      MB=1
      ME=MIN0(IP,NC)
   10 CONTINUE
      MC=ME-MB+1
C     MC=IS COL BATCH AMT
      IF (MB.NE.1) KC=KC+IP
      LC=LC+MC
      DO 20 J=1,MC
      READ (NTAP) (A(I,J),I=1,NR)
   20 CONTINUE
C     PRINT MC COL, NR ROWS
   30 JA=JB+1
      IA=1
      IB=0
      IF (JB.GE.LC) GO TO 120
      JC=JA+7
      IF (NC.LE.JC) GO TO 40
      JB=JC
      GO TO 50
   40 JB=NC
   50 IF (IB.GE.NR) GO TO 30
      IC=IA+50
      IF (NR.LE.IC) GO TO 60
      IB=IC
      GO TO 70
   60 IB=NR
   70 JD=JA-KC
      JE=JB-KC
      IF (ISEE.NE.2) GO TO 90
      PRINT 140, (TITLE(I),I=1,24),IPAGE
      PRINT 170, (J,J=JA,JB)
      DO 80 I=IA,IB
   80 PRINT 180, I,(A(I,J),J=JD,JE)
      GO TO 110
   90 PRINT 160, (TITLE(I),I=1,2),IPAGE
      PRINT 170, (J,J=JA,JB)
C     PRINT E FORMAT FOR DEBUG
      DO 100 I=IA,IB
  100 PRINT 150, I,(A(I,J),J=JD,JE)
  110 IA=IB+1
      IPAGE=IPAGE+1
      GO TO 50
  120 IF (ME.EQ.NC) GO TO 130
      MB=MB+IP
      ME=MIN0(ME+IP,NC)
      GO TO 10
  130 REWIND NTAP
      RETURN
C
C
C
  140 FORMAT (*1*,25X,12A6/26X,12A6,23X,*PAGE*I3//)
  150 FORMAT (*ROW*,I4,**,8(E15.3))
  160 FORMAT (*1*,2A10,100X,*PAGE*,I3/)
  170 FORMAT (*COLUMN*,8(I14,**))
  180 FORMAT (*ROW*,I4,**,8(F15.2))
      END
```

```
      SUBROUTINE DELETE (VECTOR,ILONG)
C     DELETE IM ELEMENTS FROM VECTOR OF LENGTH ILONG
      COMMON ISEE,IP,M,K
      COMMON MATIN,IDFOUT,ISIZE,TITLE
      COMMON IM
      DIMENSION TITLE(36), IM(510)
      DIMENSION VECTOR(ILONG), VECNEW(510)
      IF (M.EQ.0) RETURN
      NR=ILONG-M
      J=1
      L=1
      DO 20 I=1,ILONG
      IH=IM(L)
      IF (IH.EQ.I) GO TO 10
      VECNEW(J)=VECTOR(I)
      J=J+1
      GO TO 20
10    L=L+1
20    CONTINUE
      DO 30 I=1,NR
      VECTOR(I)=VECNEW(I)
30    CONTINUE
      RETURN
      END
      SUBROUTINE CTOR (IT,JT,LL,NR,NC,A,NSR,II)
C     MOVE MATRIX ON TAPE IT COL TO TAPE JT ROW
      DIMENSION A(NSR,II)
      COMMON ISEE,IP,M,K
      DIMENSION HOLD(510)
      REWIND IT
      REWIND JT
      ME=MINO(II,NR)
      MB=1
10    L=ME-MB+1
      MD=0
      DO 30 J=1,NC
C     READ A COL INTO HOLD,SELECT NEEDED ELEMENTS
      READ (IT) (HOLD(I),I=1,NR)
      MD=MD+1
      L=0
      DO 20 I=MB,ME
      L=L+1
      A(MD,L)=HOLD(I)
20    CONTINUE
30    CONTINUE
      DO 40 J=1,L
      WRITE (JT) (A(I,J),I=1,NC)
40    CONTINUE
      IF (ME.EQ.NR) GO TO 50
      MB=MB+II
      ME=MINO(ME+II,NR)
      REWIND IT
      GO TO 10
50    REWIND IT
      REWIND JT
      IF (LL.NE.1) RETURN
      DO 60 J=1,NR
      READ (JT) (HOLD(I),I=1,NC)
      WRITE (IT) (HOLD(I),I=1,NC)
60    CONTINUE
      REWIND IT
      REWIND JT
      RETURN
      END
```

Computer Program for the Leontief-Strout Gravity Type Model

Program:	Leontief-Strout Model
Authors:	Meir Carasso and Mary Carasso, March 1968
Computer:	CDC 6400
Purpose:	Calculate and print the net flows and gross flows for a given commodity (or commodities)
Restrictions:	The number of commodities $\leqslant$ 50, number of regions $\leqslant$ 15
Program Flow:	The program flow essentially is the step by step solution of the equations shown on p. 78.

Input: The input is in the following order and format:

1. One title card (80 columns of alpha-numeric data)
2. N, M, M20PT, ISEE punched in 4I4 on one card

   N = number of commodities (Col. 1-4)

   M = number of regions (Col. 5-8)

   M20PT = manner in which distances, delta, will be input, see below. (Col. 9-12)

   ISEE = blank or 0, no debug printout; = 1 then will be debug printout. Debug printout consists of a printout of all the input plus all intermediate calculations (Col. 13-16)

There will be N sets of remaining input; i.e., if N = 4, data set = cards 1., 2., + 4 times (3. - 7.). All the remaining cards are punched in 7F10.2 Format.

3. XG - (M entries)
4. XH - (M entries)

NOTE: No elements of XG or XH may = 0 as they are used as divisors. If a zero occurs, an error message is printed and the run ended.

5. d – distance matrix, the description equals that of δ.

6. δ – delta matrix = (M^2-M entries) amount of elements. The principal diagonal of the matrix consists of zeros and the remaining elements are as follows: $\delta_{ij} = \delta_{ji}$. The user may read in all the elements by setting M2OPT = 1. The elements are read in either rowwise or columnwise minus the diagonal (result is the same whether read in row or columnwise due to definition of matrix). In this case he will supply M^2-M elements. For example the input might read: δ_{12} δ_{13} δ_{14} δ_{21} δ_{23} δ_{24}, etc. punched sequentially, (a new row/column does not start a new card).

   OR

   If the user sets M2OPT = 0, only 1/2 the above amount of elements are read in (upper or lower triangle of the matrix) $= [\frac{M^2-M}{2}]$ and the computer sets up the other half of the matrix. For example, if M = 4: δ_{12} δ_{13} δ_{14} δ_{23} δ_{24} δ_{34} would be input. As per above, these may be punched row or columnwise.

7. XHH = XGG - (M entries)

Output:

The output consists of the gross flow and net flow matrices. If ISEE = 1 there will also be debug output.

Deck Setup:

Job Card
Run Card
Execute Card
7–8–9
Fortran Deck
7–8–9
Data
7–8–9
6–7–8–9

EQUATIONS:

Given:

a. $X_{i \cdot go}$ (XG)
b. $X_{i \cdot oh}$ (XH)
c. $\delta_{i \cdot gh}$ always zero or 1; $\delta_{i \cdot gg} \equiv \delta_{i \cdot hh} \equiv 0$ (delta)
d. $d_{i \cdot gh}$ (distance)
e. $X_{i \cdot gg} \equiv X_{i \cdot hh}$ (XHH or XGG)

Calculations:

a. Calculate amount of gross flows matrix X:

$$X_{i\cdot gh} = (X_{i\cdot go} X_{i\cdot oh}) \cdot (C_{i\cdot g} + K_{ih}) (d_{igh}\, \delta_{igh})$$
$$i = 1, N$$
$$g, h = 1, M$$

b. $C_{i\cdot g}$ and K_{ih} are solved via matrix inversion, i.e., $AX = Y$, solve for $X = A^{-1}\ Y$.

The matrix is described on the following page.

c. Calculate net flows matrix X':

$$X'_{gh} = X_{gh} - X_{hg} \text{ if } X_{gh} > X_{hg}$$

or $$X'_{hg} = X_{hg} - X_{gh} \text{ if } X_{gh} < X_{hg}$$

i.e., $X_{12} = 5$, $X_{21} = 3$: $X'_{12} = 5 - (-3) = 8$

$X_{12} = 4$, $X_{21} = 6$: $X'_{21} = 6 - 4 = 2$

Check:

1. $\sum_h X_{i\cdot gh} = X_{i\cdot go} - X_{i\cdot gg}$

2. $\sum_g X_{i\cdot gh} = X_{i\cdot oh} - X_{i\cdot hh}$

$$
\begin{bmatrix}
\sum_r \frac{X_{ro}\, d_{r2}\, \delta_{r2}}{X_{02}} & 0 & \cdots & 0 & d_{12}\,\delta_{12} & 0 & d_{32}\,\delta_{32} & \cdots & d_{m2}\,\delta_{m2} \\
0 & \sum_r \frac{X_{ro}\, d_{r3}\, \delta_{r3}}{X_{03}} & \cdots & 0 & d_{13}\,\delta_{13} & d_{23}\,\delta_{23} & 0 & \cdots & d_{m3}\,\delta_{m3} \\
0 & 0 & \cdots & \sum_r \frac{X_{ro}\, d_{rm}\, \delta_{rm}}{X_{om}} & d_{1m}\,\delta_{1m} & d_{2m}\,\delta_{2m} & d_{3m}\,\delta_{3m} & \cdots & 0 \\
d_{12}\,\delta_{12} & d_{13}\,\delta_{13} & \cdots & d_{1m}\,\delta_{1m} & \sum_r \frac{X_{or}\, d_{1r}\, \delta_{1r}}{X_{10}} & 0 & 0 & \cdots & 0 \\
0 & d_{23}\,\delta_{23} & \cdots & d_{2m}\,\delta_{2m} & 0 & \sum_r \frac{X_{or}\, d_{2r}\, \delta_{2r}}{X_{20}} & 0 & \cdots & 0 \\
d_{32}\,\delta_{32} & 0 & \cdots & d_{3m}\,\delta_{3m} & 0 & 0 & \sum_r \frac{X_{or}\, d_{3r}\, \delta_{3r}}{X_{30}} & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
d_{m2}\,\delta_{m2} & d_{m3}\,\delta_{m3} & \cdots & 0 & 0 & 0 & 0 & \cdots & \sum_r \frac{X_{or}\, d_{mr}\, \delta_{mr}}{X_{mo}}
\end{bmatrix}
\begin{bmatrix}
X_{02}\, K_2 \\ X_{03}\, K_3 \\ \vdots \\ X_{om}\, K_m \\ X_{10}\, C_1 \\ X_{20}\, C_2 \\ X_{30}\, C_3 \\ \vdots \\ X_{mo}\, C_m
\end{bmatrix}
=
\begin{bmatrix}
1 - \frac{X_{22}}{X_{02}} \\ 1 - \frac{X_{33}}{X_{03}} \\ \vdots \\ 1 - \frac{X_{mm}}{X_{om}} \\ 1 - \frac{X_{11}}{X_{10}} \\ 1 - \frac{X_{22}}{X_{20}} \\ 1 - \frac{X_{33}}{X_{30}} \\ \vdots \\ 1 - \frac{X_{mm}}{X_{mo}}
\end{bmatrix}
$$

$$[A] \qquad\qquad [X] \qquad [Y]$$

To solve for C,K: $X = A^{-1}\, Y$

```
C      LECNTIEF-STROUT GRAVITY FLOW MODEL PROGRAM
C
C      NET TRADE BALANCES AMCNG REGICNS MAKING UP AN INTERREGIONAL MODEL
C      MAY BE CALCULATED BY USING THIS PROGRAM
C
C
C
       PROGRAM LS(INPUT,OUTPUT,PUNCH)
C        CORE SIZE CALC    6C+C**2 + 2(C*R**2) + 5R + 3R**2
C      WHERE C = COMMOCITEIS AND R=REGIONS AND CISTANCES (ASSUME EQUAL.
       DIMENSION X(50,15,15),XPP(50,15,15),XG(15),XH(15),CIS(15,15),
      1DEL(15,15),A(50,50),C(15),K(15),Y(50),XX(50),XGH(50),XHH(50),
      2TITLE(8),HOLD(110),XP(110),IR(110),IC(11C)
         COMMON ISEE
       COMMCN TITLE,N,M
       DATA NN/0/
       REAL K
C      M2OPT=1=READ IN ALL MCOMB DIS,DEL, =0 REAC IN 1/2 SET UP REST
C      N=NUMBER OF COMMODITIES=I LIMIT, M=NUMBER TO-FROM SETS=G,H LIMIT
       READ 12, (TITLE(I),I=1,8)
12     FORMAT(8A10)
       PRINT 1,TITLE
1      FORMAT(*1*,8A10)
       READ 5,N,M,M2OPT,ISEE
5      FORMAT(4I4)
.      PRINT 2,N,M,M2OPT,ISEE
2      FCRMAT(* CONTROL CARD *,4I6)
C      LIMIT CHECK
       IF((M.GT.15).OR.(M.LT.1).OR.(N.GT.50).OR.(N.LT.1).CR.
      1     (M2OPT.GT.1).OR.(M2OPT.LT.0)) GC TO 6
       DC 4 I=1,N
       DO 4 J=1,M
       DO 4 L=1,M
       X(I,J,L)=0.
4      XPP(I,J,L)=0.
       GC TC 10
     6 PRINT 7
       STOP
     7 FCRMAT(29H CONTROL CARD ERROR*STOP RUN*)
10     IF (N.EQ.NN) GO TO 300
       NN=NN+1
       PRINT 3,NN,N
3      FCRMAT(* NN=*,I4,*N =*,I4)
       DC 8 I=1,M
       DC 8 J=1,M
       DIS(I,J)=0.
8      DEL(I,J)=0.
       M1=M-1
       M2=M1+M
       M1P=M1+1
       DC 9 J=1,M2
       DC 9 I=1,M2
9      A(I,J)=0.
C      COUNTER FOR PROCESSING
       READ 15, (XG(I),I=1,M)
       CALL SEEV(M,XG, 20H XG READ IN           )
       READ 15, (XH(I),I=1,M)
       CALL SEEV(M,XH, 20H XH READ IN           )
15     FCRMAT(7F10.2)
C      XH NCR XG CANNOT BE ZERO AS THEY = DIVISCRS
C      CHECK FOR ZEROS
       DC  20 I=1,M
    20 IF (XG(I).EQ.0.) GO TO 22
       GC TO 25
    22 PRINT 23,I
    23 FCRMAT( 4H XG(,I2,33H) = 0,CANNOT CONTINUE-SEE WRITEUP)
       STOP
```

```
   25 DC 30 I=1,M
      IF (XH(I).EQ.0.) GO TC 32
  30  CCNTINUE
      GC TC 35
   32 PRINT 33,I
   33 FCRMAT( 4H XH(,I2,33H) = 0,CANNOT CCNTINUE-SEE WRITEUP)
C     CALC COMBINATIONS CF M-DIAG
 35   MCOMB= M**2 - M
      MCOMB2=MCOMB/2
   50 IF(M2OPT.EQ.1) GO TC 75
C     M2OPT = 0 SO READ IN HALF SET UP HALF DIS,DEL
      READ 15,(HOLD(I),I=1,MCOMB2)
C     READ INTO VECTOR SHUFFLE TO MATRIX
      CALL SEEV(MCOMB2,HOLD,20H DIS-READ IN-HALF    )
C     DISTANCES FIRST
      L=0
      DC 60 I=1,M
      K=I-1
      DC 60 J=1,M
      IF (I.EC.J) GO TO 60
      IF (K.EC.0) GO TO 55
      DIS(I,J)=DIS(J,I)
      K=K-1
      GC TC 60
 55   L=L+1
      DIS(I,J)=HOLD(L)
   60 CCNTINUE
      CALL SEEM (M,M,CIS,20H DIS MATRIX    60    ,15,15)
C     DELTAS NEXT-COMMENTS---SEE DIS
      READ 15,(HOLD(I),I=1,MCOMB2)
      99'' 7997P=98=95PR8'9P57R 99'8H9999I28R9'9   R
      L=0
      DC 70 I=1,M
      K=I-1
      DC 70 J=1,M
      IF(I.EQ.J) GO TC 70
      IF(K.EC.0) GO TC 65
      DEL(I,J)=DEL(J,I)
      K=K-1
      GC TC 70
   65 L=L+1
      DEL(I,J)=HOLD(L)
   70 CCNTINUE
      CALL SEEM (M,M,CEL,20H DEL MATRIX    70    ,15,15)
      GC TC 100
C     ALL DIS, DEL GIVEN SET UP AS MATRIX - DIAGONAL
C     READ INTO HOLD AND SHUFFLE INTC MATRIX
C     DISTANCE FIRST
   75 READ 15,(HOLD(I),I=1,MCOMB)
      CALL SEEV(MCOMB,HOLD,20H DIS-READ IN-ALL     )
      L=0
      DO 80 I=1,M
      DC 80 J=1,M
      IF (I.EQ.J) GO TO 80
      L=L+1
      DIS(I,J)=HOLD(L)
   80 CCNTINUE
      CALL SEEM(M,M,DIS,20H DIS MATRIX         ,15,15)
C     DELTAS
      READ 15,(HOLD(I),I=1,MCOMB)
      CALL SEEV(MCOMB,HOLD,20H DEL-READ IN-ALL 80 )
      L=0
      DC 85 I=1,M
      DC 85 J=1,M
      IF(I.EQ.J) GO TC 85
      L=L+1
```

```
      DEL(I,J)=HOLD(L)
   85 CONTINUE
      CALL SEEM(M,M,DEL,20H DEL MATRIX    85   ,15,15)
C     FORM MATRIX A
C     FORM UPPER LEFT  QUADRANT DIAGONAL
C     = S(X(R,O)*DIS(R,M)*DEL(R,M))/X(O,M)---M MINUS 1 AMT  AS K(1)=0
C          R=1,M  M=2,M
C     REMAINDER  QUADRANT = 0
  100 DO 112 J=2,M
      I=J-1
      A(I,I)=0.
      DO 110 L=1,M
      IF (DEL(L,J).EQ.0.) GO TO 110
C     DEL IS EITHER 0 OR 1 SO IF 0 NO USE DOING EQ
C     DEL ALWAYS 0 ON DIAGONAL
      A(I,I)=A(I,I) + ((XG(L)*DIS(L,J)*DEL(L,J))/XH(J))
  110 CONTINUE
  112 CONTINUE
C     FORM LOWER RIGHT QUADRANT DIAGONAL
C     = S(XH(R,O)*DIS(M,R)*DEL(M,R))/XG(R)     M AMT
C         R=1,M M=2,M
C     REMAINDER QUADRANT = 0  SEE COMMENT ON UPPER LEFT QUADRANT ABOUT D
      DO  122 J=1,M
      I=J+M1
      A(I,I)=0.
      DO  120 L=1,M
      IF (DEL(J,L).EQ.0.) GO  TO 120
      A(I,I)=A(I,I)+ ((XH(L)*DIS(J,L)*DEL(J,L))/XG(J))
  120 CONTINUE
  122 CONTINUE
C     UPPER RT QUADRANT
      DO 128 J=2,M
      II=J-1
      DO 128 I=1,M
      KI=I+M1
      IF (DEL(I,J).EQ.0.)GO TO 128
      A(II,KI)= DIS(I,J)*DEL(I,J)
  128 CONTINUE
C     LOWER LEFT QUADRANT
      DO 135 J=2,M
      KI=J-1
      DO 135 I=1,M
      II=I+M1
      IF (DEL(I,J).EQ.0.)GO TO 135
      A(II,KI)=DIS(I,J)*DEL(I,J)
  135 CONTINUE
      CALL SEEM(M2,M2,A,20H ALL OF A      135    ,50,50)
C     READ IN X(MM)
      READ 15, (XHH(I),I=1,M)
      CALL SEEV(M,XHH,20H XGG=XHH READ IN     )
C     CALCULATE Y FOR A.X=Y OR X=A-1.Y
C     Y=1-X(MM)/XOM  THEM =X(MM)/X(MO)
      DO 138 I=1,M2
  138 Y(I)=0.
      DO 140 J=1,M1
      I=J+1
      IF(XHH(I).EQ.XH(I)) GO TO 140
      Y(J)= 1.- (XHH(I)/XH(I))
  140  CONTINUE
      DO  145 I=1,M
      II=I+M1
      IF(XHH(I).EQ.XG(I)) GO TO 145
      Y(II)=1.- (XHH(I)/XG(I))
  145 CONTINUE
      CALL SEEV(M2,Y,20H Y FOR FINDING X 145)
C     NOW INVERT A  VIA F1*MLFHINT--CROUT INVERSION
      CALL INVERT(A,M2,ISING)
```

```
      IF (ISING.EQ.0) GO TO 150
      PRINT 148
  148 FCRMAT(30H MATRIX IS SINGULAR-END OF RUN)
      STOP
C     NOW MULTIPLY A-1*Y
  150 CALL SEEM(M2,M2,A,20H   A INVERSE     150 ,50,50)
      DO 160 I=1,M2
      XX(I)=0.
      DO 160 J=1,M2
  160 XX(I)= XX(I) + (A(I,J)*Y(J))
      CALL SEEV (M2,XX,20H A-1*Y=X       160    )
C     NOW GET REAL X=M1 CS AND M KS
      K(1)=0.
      DO 170 I=1,M2
      IF(I.GT.M1) GO TO 162
      J=I+1
      K(J)=XX(I)/XH(J)
      GO TO 170
  162 J=I-M1
      C(J)=XX(I)/XG(J)
  170 CONTINUE
      CALL SEEV( M,K,20H K VECTOR      170    )
      CALL SEEV(M, C,20H C VECTOR             )
C     CALCULATE (ALAS) MAIN EQUATION --- 3DIMENSIONALLY FOR JAZZY PRINT
      DO 200 I=1,M
      DO 200 J=1,M
      IF(DEL(I,J).EQ.0) GO TO 190
      X(NN,I,J)=  (XG(I)*XH(J))      * (C(I)+K(J)) * (DIS(I,J)*DEL(I,J))
 190     A(I,J)=X(NN,I,J)
C     ABOVE JUST FOR DEBUG OUT
  200 CONTINUE
      CALL SEEM(M,M,   A,20H X FOR AN I       200,50,50)
C     XP WILL CONTAIN ALGEBRAIC VALUE  II,JJ INDEX VALUE ASSOCIATED
      L=0
      DO  210 I=1,M1
      DO  210 J=1,M
      IF (J.LE.I) GO TO 210
      L=L+1
      IF  (X(NN,I,J).GE.X(NN,J,I)) GO TO 204
C     LESS THAN INDEX
      IR(L)=J
      IRR=J
      IC(L)=I
      ICC=I
      GO TO 206
  204 IR(L)=I
      IRR=I
      IC(L)=J
      ICC=J
206   XPP(NN,IRR,ICC)=ABS( X(NN,I,J)-X(NN,J,I))
207   XP(L)=XPP(NN,IRR,ICC)
  210 CONTINUE
      CALL SEEV(MCOMB2,XP, 20H XPRIME       210     )
      PRINT 255,(IR(I),I=1,MCOMB2)
      PRINT 256,(IC(I),I=1,MCOMB2)
 255  FORMAT(10H ROW INDEX,20I5)
 256  FORMAT(10H COL INDEX,20I5)
      PRINT 272
      DO 273 J=1,M
      SR=0.
      SC=0.
      DO 265 I=1,M
      SR = SR + XPP(NN,J,I)
  265 SC = SC + XPP(NN,I,J)
      XD = XH(J) - XG(J)
      XY=ABS(SR-SC)
      PRINT 266,XD,XY
```

```
  266 FORMAT(//* ---------CHECK RESULTS*,3(E12.3,2X))
  270 CONTINUE
      PRINT 272
  272 FORMAT(/////)
      GO TO 10
C     PRINT X,XPP ARRAYS
300   CALL PRINT(X,0)
      CALL PRINT(XPP,1)
      PRINT 250
  250 FORMAT( 4H EOJ)
      CALL EXIT
      END
      SUBROUTINE SEEM(IROW,ICOL,AMATRX,TITLE,IROWD,ICOLD)
        COMMON ISEE
C     PRINTS IROW AND ICOL OF A MATRIX (AMATRX)    *DEBUG   *
C     DIMENSIONED IROWD * ICOLD,WITH A TITLE       *ROUTINE*
      DIMENSION AMATRX(IROWD,ICOLD),TITLE(2)
      IF(ISEE.EQ.0) RETURN
      PRINT 3
      PRINT 1,(TITLE(I),I=1,2)
      DO 10 I=1,IROW
      PRINT 5
   10 PRINT 2, (AMATRX(I,J),J=1,ICOL)
      RETURN
    1 FORMAT(1H ,2A10)
    2 FORMAT(1H ,10(E12.3),1H ))
    3 FORMAT(1H ,20H--------------------)
    5 FORMAT(1H )
      END
      SUBROUTINE SEEV(ILONG,VECTOR,TITLE)
        COMMON ISEE
C     PRINTS VECTOR OF LENGTH ILONG WITH TITLE     *DEBUG   *
C     VECTOR PRINTS AS A ROW                       *ROUTINE*
      DIMENSION VECTOR(50),TITLE(2)
      IF(ISEE.EQ.0) RETURN
      PRINT 3
      PRINT 1,(TITLE(I),I=1,2)
      PRINT 2, (VECTOR(I),I=1,ILONG)
      PRINT 4
      RETURN
    1 FORMAT(1H ,2A10)
    2 FORMAT(1H ,10(E12.3,1H ))
    3 FORMAT(1H ,20H--------------------)
    4 FORMAT(1H )
      END
      SUBROUTINE PRINT(A,II)
      DIMENSION A(50,15,15),TITLE(8)
C     PRINT 3DDM MATRIX -VARYING DIM 1-- A(N,M,M)--PRINT N (M,M) MATRICE
        COMMON ISEE
      COMMON TITLE,N,M
      DO 10 L=1,N
      DO 1  J=1,M,7
      JB=J
      JE=JB+6
      IF(JE-M)2,3,3
3     JE=M
2     PRINT 200,(TITLE(I),I=1,8)
      IF(II.GT.0) GO TO 4
      PRINT 201,L
      GO TO 5
4     PRINT 202,L
5     PRINT 204, (K,K=JB,JE)
      DO 6 I=1,M
6     PRINT203,(I,(A(L,I,JS),JS=JB,JE))
1     CONTINUE
10    CONTINUE
      IF(II.LE.0) RETURN
```

```
      PRINT 205
      DO 50 J=1,M
      PUNCH 215,J
      PRINT 216,J
      DO 45 K=1,M
      PUNCH 215,K
      PRINT 216,K
      IF (J.NE.K) PUNCH 210, (A(I,J,K),I=1,N)
      PRINT 220, (A(I,J,K),I=1,N)
45    CCNTINUE
50    CCNTINUE
      RETURN
2C5   FCRMAT(*1*)
215   FCRMAT(I4)
216   FCRMAT(//I6)
210   FCRMAT(7F10.2)
220   FCRMAT(* *,7(F1C.2,3X))
200   FCRMAT(1H1,40X,8A10  )                                          L/S 3070
2C1   FCRMAT(1H ,40X,23HGROSS FLOWS--COMMODITY ,I2///)                L/S 3080
2C2   FCRMAT(1H ,40X 23H  NET FLCWS--COMMCDITY ,I2///)                L/S 3090
2C3   FCKMAT(I8,7F16.7)                                               L/S 3100
2C4   FCRMAT(13X,I3,6I16)                                             L/S 3110
      RETURN                                                          L/S 3120
      END                                                             L/S 3130
CHINT   MATRIX INVERSICN SUBROUTINE    UPDATEC JULY 1964              L/S 3140
      SUBRCUTINE INVERT(A,IMAX,ISING)                                 L/S 3150
      DIMENSION A(50,50),IN(5C),TEMP(50)                              L/S 3160
      ISING=0                                                         L/S 3170
      N=IMAX                                                          L/S 3180
      IMAXC=N-1                                                       L/S 3190
      I1=1                                                            L/S 3200
    1 I3=I1                                                           L/S 3210
      IN(I1)=0                                                        L/S 3220
      SLM=ABS (A(I1,I1))                                              L/S 3230
      DC 3 I=I1,N                                                     L/S 3240
      IF (SUM-ABS (A(I,I1))) 2,3,3                                    L/S 3250
    2 I3=I                                                            L/S 3260
      IN(I1)=I                                                        L/S 3270
      SLM=ABS (A(I,I1))                                               L/S 3280
    3 CCNTINUE                                                        L/S 3290
      IF (I3-I1) 4,6,4                                                L/S 3300
    4 DC 5 J=1,N                                                      L/S 3310
      SUM=A(I1,J)                                                     L/S 3320
      A(I1,J)=A(I3,J)                                                 L/S 3330
    5 A(I3,J)=SUM                                                     L/S 3340
    6 I3=I1+1                                                         L/S 3350
      IF (A(I1,I1)) 7,31,7                                            L/S 3360
    7 DC 8 I=I3,N                                                     L/S 3370
    8 A(I,I1)=A(I,I1)/A(I1,I1)                                        L/S 3380
      J2=I1-1                                                         L/S 3390
      IF (J2) 9,11,9                                                  L/S 3400
    9 DC 10 J=I3,N                                                    L/S 3410
      DC 10 I=1,J2                                                    L/S 3420
   10 A(I1,J)=A(I1,J)-A(I1,I)*A(I,J)                                  L/S 3430
   11 J2=I1                                                           L/S 3440
      I1=I1+1                                                         L/S 3450
      DC 12 I=I1,N                                                    L/S 3460
      DC 12 J=1,J2                                                    L/S 3470
   12 A(I,I1)=A(I,I1)-A(I,J)*A(J,I1)                                  L/S 3480
      IF (I1-N) 1,13,1                                                L/S 3490
   13 IF (A(N,N)) 14,31,14                                            L/S 3500
   14 DC 17 JP=1,N                                                    L/S 3510
      J=N+1-JP                                                        L/S 3520
      A(J,J)=1.C/A(J,J)                                               L/S 3530
      IF (J-1) 15,18,15                                               L/S 3540
```

```
15 DO 17 IP=2,J
   I=J+1-IP
   IPO=I+1
   SUM=0.
   DO 16 L=IPO,J
16 SUM=SUM-A(I,L)*A(L,J)
17 A(I,J)=SUM/A(I,I)
18 DO 22 J=1,IMAXO
   JPO=J+1
   DO 22 I=JPO,N
   SUM=0.
   IMO=I-1
   DO 21 L=J,IMO
   IF (L-J) 19,20,19
19 SUM=SUM-A(I,L)*A(L,J)
   GO TO 21
20 SUM=SUM-A(I,L)
21 CONTINUE
22 A(I,J)=SUM
   DO 27 I=1,N
   DO 26 J=1,N
   TEMP(J)=0.0
   DO 25 K=I,N
   IF (K-J) 25,24,23
23 TEMP(J)=TEMP(J)+A(I,K)*A(K,J)
   GO TO 25
24 TEMP(J)=TEMP(J)+A(I,K)
25 CONTINUE
26 CONTINUE
   DO 27 J=1,N
27 A(I,J)=TEMP(J)
   DO 30 I=2,N
   M=N+1-I
   IF (IN(M)) 28,30,28
28 ISS=IN(M)
   DO 29 L=1,N
   SUM=A(L,ISS)
   A(L,ISS)=A(L,M)
29 A(L,M)=SUM
30 CONTINUE
   GO TO 32
31 ISING=1
32 RETURN
   END
```

Ruth2 Computer Program for Manipulating Leontief Type Models

Purpose: At the option of the user, this program transforms, aggregates, and/or performs a variety of matrix manipulations for routine work with Leontief type models.

Language: Extended Fortran or Fortran IV

Computers: The program was designed to run on CDC 6400 or 6600 machines in Fortran IV or Extended Fortran, using four scratch disks. Up to four tapes may be required, depending on the user's option described later: the input flow matrix may be on tape, the final matrix may be on tape, all intermediate printed matrices may be on tape, and the composite flow matrix may be created on tape.

Restrictions: There are two restrictions with RUTH2.

1. The order of the input matrix is limited by the core size available. The formula for determining the core size for a given run is as follows, where N = order of the input matrix:

$$\text{CORE SIZE}_8 = 55000_8 + N_8^2$$

This is given in octal as the core capacity of a computer is usually handled in octal. Also, the formula is based on a loader size of 4000_8. The dynamic core size is set up by the user by one dimension statement in the program plus one of the control card parameters. This will be discussed in detail in Section under control card option ND.

2. If the user chooses to invert his matrix, the restriction on the inversion order is about 200. This restriction is due to timing. Actually a matrix as large as the core allows (see above) could be inverted if time was not of the essence. For a matrix larger than 200 the HERP Inversion should be used; this is described at the end of this appendix.

Method: The program requires five basic control cards and associated data. At user's discretion, the main control card governs the four options for the form of the "final matrix" the matrix used for further matrix operations. Additionally, with these four options, further matrix manipulations and print and punch options are available. The basic "transactions" or "flow" matrix, described below, can be card or tape input.

PROGRAM DESCRIPTION AND OPERATION:

A. Read Control Cards. (All five control cards must be present and in proper sequence.)
   1. Two title cards containing any alphanumeric information in colums 1-72 for user's convenience, i.e., labeling or otherwise designating printout sheets. The cards may be left blank, but as stressed above, must be present as part of the control card sequence.
   2. Main Control Card Punched In 13I4 FORMAT: The 13 fields in order of occurrence provide by means of numeric punches for the various choices and options of the program. These choices and options are as follows:
      a. Order of the input flow matrix (N)
      b. Choice of matrix to be formed from basic transactions matrix, e.g., direct coefficients matrix, Leontief inverse, etc. (KHOICE).
      c. Aggregation of transactions matrix to any specified number of new sectors or deletion of rows and columns (NP).
      d. Number of column vectors by which desired matrix is to be post-multiplied (MX).
      e. Print of input matrix (for verification) (IPRINT).
      f. Multiply desired matrix, or its transpose by a diagonal matrix(s) (MULT).
      g. Punch option 1. Provides for option to punch, or not to punch (or put on tape for punching by peripheral equipment) the final matrix formed by the program. (The final matrix is described under this section, C.2.b.). The punch format can be of the form for reinput to the program or can be specified to be punched on a separate data card. (A TRANSLATE program exists for translating M3LP input to MPS input for use with the IBM 360 MPS linear programming system.) (IPUNCH1)
      h. Provides for transpose of final matrix before multiplying by column vectors (NTRANV).
      i. Punch option 2. This option provides for punching everything that is printed (except the input matrix) in a format suitable for reinput to the program. This output may also be placed on tape and punched by peripheral equipment. This option is not normally used and cannot be used as tape input to the program for further runs, unless the first matrix is desired input matrix. Card output may be used for input if it is selected discriminately (IPUNCH2).
      j. Tape input option (for large transactions matrices) (ITAPE).
      k. Parameter which declares the size of the dimension statement of ABAR matrix in the user's source deck (ND).
      l. Tape option to create a binary rowwise tape for input to HERP matrix inversion routine (INV).
      m. Option to put sets of original input matrix sets of MULT matrices on a tape in a specified composite form for further use (MULT2).

3. 1st Format Card describes the format of the matrix if on card or BCD tape. This card must always be present.
4. 2nd Format Card describes the format in which all punched matrices, except M3LP matrix are punched.

B. Read Corresponding Data According to Program Flow
1. If aggregation routine is called for (NP $\neq$ blank or zero) the program requires a set of grouping cards punched in 24I3 format to follow control cards. These are described in detail below and must be followed by a blank card.
2. Read matrix.
3. If option calls for any variant requiring the matrix to be normalized, a normalizing vector (usually a vector of gross outputs) must follow the matrix immediately. For option not requiring matrix to be normalized, this vector must be absent from input deck.
4. If matrix multiply option (MULT) is selected, a vector of length N will follow in the input stream.
5. If multiplying by column vectors (MX), there will be MX vectors of length N with one title card preceding each vector next in the input data stream.
6. Data terminates with two blank cards followed by a card with 9999 punched in columns 1-4. Multiple sets of data may be run by resuming input format beginning in 1. above, i.e., title cards, control cards, etc. If multiple sets of data are run, there is no end of file card nor 3 terminator cards between sets, only three terminator cards at the end of the entire deck, followed by an EOF card (6-7-8-9).

C. Detail Input Information of 5 Mandatory Cards: The deck setup must always begin with the following 5 cards: 2 title cards, 1 control card, 2 format cards.
1. The two title cards can contain any alphanumeric information in columns 1-72 for user's convenience, i.e., designating printout sheets and input data used. They may be left blank.
2. The control card provides for options through a choice of 13 variables in the order given below. The card is in 13I4 Format. All variables are right justified within the four-column fields.
   a. Col 1-4: N: order of input matrix. The order of the input matrix must be $\leqslant$ ND (see k. below), the core determiner.
   b. Col 7: + or − (plus or minus): a plus sign (or blank) deletes print-out of required matrix, a minus sign yields print-out.
   Col 8: KHOICE: Determines final matrix to be computed (before multiply options):

KHOICE	MATRIX
1	input matrix A
2	normalized input matrix $\overline{A}$
3	$(I - \overline{A})$
4	$(I - \overline{A})^{-1}$

c. Col 9-12: **NP**: if NP 0 or blank, matrix is not aggregated. If NP greater than 0, matrix will be aggregated and NP = order of aggregated matrix. NP must be less than N.
d. Col 13-16: **MX**: The number of vectors the final matrix is to be multiplied by.
e. Col 20: **IPRINT**: If IPRINT = 1 input matrix will be printed, otherwise not.
f. Col 24: **MULT**: If MULT = 0 or blank the option is bypassed. The MULT option allows the user to multiply the final matrix by a vector as though the vector were a diagonal matrix, therefore yielding a matrix. There are two features of MULT. If MULT is a positive number, the final matrix is transposed before multiplying. If MULT is a negative number, the final matrix is **not** transposed before multiplying. Secondly, **number** equals the number of vectors to be multiplied by the final matrix – each vector is multiplied by the final matrix and printed (or utilized on MULT2 option). (There is no accumulation.)
g. Col 28: **IPUNCH1**: If IPUNCH1 = 0 or blank the option is bypassed. If IPUNCH1 = 1 the final matrix is punched in M3LP format. If IPUNCH1 = 2 the final matrix will be punched columnwise. If IPUNCH1 = 3 the final matrix is put in a BCD tape columnwise in format 1 (see j. below).
h. Col 32: **NTRANV**: If NTRANV = 1 the column vector described in d. above (MX option) multiply the TRANSPOSED final matrix. If NTRANV = 0 the final matrix is used.
i. Col 36: **IPUNCH2**: If IPUNCH2 = 0 or blank all matrices that are printed are punched except the inverted matrix of KHOICE = 4 (to punch $(I - \bar{A})^{-1}$ IPUNCH1 must be = 2). If IPUNCH2 = 1 the printed matrices are not punched. If IPUNCH 2 = 2 the punch-out will be put on tape – BCD and columnwise in format 1 (see j. below).
j. Col 40: **ITAPE**: If ITAPE = 0 or blank the input flow matrix is on cards columnwise. If ITAPE = 1 the input flow matrix is on tape in BCD columnwise format with the first **record** of tape = N (order of matrix) in 14 format (format 1). If ITAPE = 2 the input flow matrix is on tape in BCD columnwise format with the first record = start of matrix (format 2). If ITAPE = 3 the input matrix is in binary, columnwise. If the MULT2 option (see m. below) was used to create a composite matrix from multiple sets of matrices, **and** that matrix is now the input matrix, this option of ITAPE = 3 **must** be chosen.
k. Col 42-44: ND: The parameter ND is equal to the order of the matrix ABAR given in the DIMENSION statement of the main Fortran program of RUTH2; i.e., if the Fortran Card (sequenced A32) reads DIMENSION ABAR (110, 110), then ND would have the value of 110 on the control card. The remainder of the Fortran subprograms use variable dimension statements based on the value of ND. Therefore ND **must** correspond to the aforementioned ABAR dimension or else either the program will blow

up or produce erroneous results; No error message can be given, therefore this is totally the user's responsibility.

l. Col 48: INV: If INV = 0 or blank, this option is bypassed. If INV = 1, a binary rowwise tape of the final matrix is created using Tape 4. The primary use of this option is to create a tape which can be input into the HERP inversion program.

m. Col 52: MULT2: If MULT2 = 0 or blank this option is bypassed. If MULT2 - 1 a composite matrix is formed from several flow matrices to yield a regional flow matrix. The following description uses the California (CA), Great Basin (GB), Lower Colorado (LC), and Upper Colorado (UC), flow matrixes as an example of assembling 4 area flow matrices into a larger regional matrix. The following is repeated four times (or for the number of matrices to be assembled) by using multiple run feature. [The input flow matrix from the FLOW program is "uncondensed," i.e., with zero rows and columns. It is normalized by Row (observing the no zero divide). The normalizing vector is typically in this case gross output minus exports. The resulting normalized matrix is used with the multiple MULT option (in this example MULT = +3). The MULT vectors usually consist of punched output vectors from the Leontief-Strout (L/S) program. These vectors must be of the same order as the "uncondensed" input matrix. The resulting multiple matrix multiplication yields a set of (3) matrices.]

NOTE: The L/S program yields a zero vector for each matrix set depending on its order in the setup = i.e., for CA (first data set) the zero vector is 1, for GS (second data set) the zero vector is 2, etc. This vector is obviously not utilized in the multiplications but for clarification the matrices below have been labeled from 1-4 although in each set one of the numbers is absent! In our example we have 4 input matrices and each of these has 3 MULT matrices yielding a total of 16 matrices. The order of each is 40 and they will be assembled into a 160-order matrix as follows:

1 40	41 80	81 120	121 160
CA INPUT	GB MULT Vector 1	UC MULT Vector 1	LC MULT Vector 1
CA MULT Vector 2	GB INPUT	UC MULT Vector 2	LC MULT Vector 2
CA MULT Vector 3	GB MULT Vector 3	UC INPUT	LC MULT Vector 3
CA MULT Vector 4	GB MULT Vector 4	UC MULT Vector 4	LC INPUT

This matrix is put on tape 8 in binary columnwise; i.e., column 1 would be column 1 of the CA input matrix in elements 1-40, CA-MULT option matrix times vector 2 – matrix column 1 in elements 41-80, etc. This is called the composite matrix which is mentioned under ITAPE above and if input for further use, ITAPE must = 3. If this composite matrix is to be utilized further, it must be done in a separate run: it does not at any time equal the final matrix. This whole arrangement is accomplished by utilizing the multiple run setup described earlier. **This is mandatory.** In this example there would be 4 data sets followed by terminator cards and EOF after the Fortran deck.

3. **Matrix Input Format Card**: Describes the format of input flow matrix including left and right parentheses but not the statement label nor the word FORMAT. This may be punched anywhere on the card. For example: (7F10.2).
4. **Matrix Output Format Card**: Describes the format in which all punched matrices, except M3LP format matrix, are punched. It is the same as input format card, described above. The format includes left and right parentheses but **not** the statement label **nor** word FORMAT. This may be punched anywhere on card. This format is restrictive in that the number of fields per card must equal 7: 7E10.2 or 7F10.6 are acceptable, 5E12.6 will cause incorrect punch output except where N is $\leqslant 5$. Due to the sequencing of the cards (see output description-section E.2.c.), the matrix elements can only use 70 columns per card. NOTE: Current restrictions (9/68) on CDC 6400 compiler: Only **this** format card must be followed by, (commas) to end of card. If (7F10.2) is punched in Col 1-8, then Col 9-80 = ,, (commas).

D. **Detail Input Information of Remaining Data.** (Following is a description of the format of the cards other than the 5 mandatory cards.)
   1. Grouping Cards Are Punched In 24I3 Format. All variables are right justified within the 3 column fields. The first entry (Col 1-3) shows the new group (or sector) number. The remaining entries on the card shows the sector numbers of the original matrix A that are to be aggregated into the new sector (at most 23 on a card). If there are more than 23 sectors in a group, a second card with the same number in Col 1-3 is used. There should be NP groups. Rows and columns may be deleted by omission. **A blank card must terminate the set of grouping cards.**
   2. Vectors: All vectors are to be input in 7F10.2 format.
   3. Title Card: Precedes **each** column vector that multiplies the final matrix (MX option) and can have any alphanumeric information punched in Columns 1-72.
   4. Input Matrix: If the matrix is on card or tape, the matrix input card outlined above describes the data. If the matrix is on tape, it may be in one of two arrangements as described under control card – ITAPE area.

E. **Program Output**
   1. Printed Output:
      a. The control card is always printed for verification of run results (with options labeled).
      b. The grouping cards are printed equal to the card image of the input.
      c. The normalizing vector is printed in 7F10.2 Format.
      d. The input matrix is printed if IPRINT = 1. NOTE: All matrices are printed in 7F16.7 columns per page.

e. If KHOICE is negative the final matrix is printed. If the final matrix is the inverse (KHOICE = 4) and KHOICE is negative, the Transpose is printed.
f. The MULT option always prints the resultant matrix.
g. If the final matrix is multiplied by column vector(s) (MX option), the resulting column vector(s) is printed in 7F10.2 Format.
h. If IPUNCH1 option is for M3LP output, each element is printed (elements less than 10^{-7} are not printed).
i. **If the program terminates correctly, the last line of output will read: EOJ RUTH2.**

2. Punch Output:
   a. The normalizing vector is always punched in 7F10.2 Format.
   b. If the final matrix is multiplied by column vector(s) (MX option), the resulting column vector(s) is always punched in 7F10.2 with the input vector number in columns 73-80.

      If the user desires to change the format of the punch out, the Fortran source card sequenced A242 should be changed appropriately, as from (7F10.2, 5X, 12) to (7F10.6, 5X, 12). NOTE: Only the underlined portion (previous sentence) may be changed and the new field width cannot be greater than 10.
   c. The matrices that are punched depend on the users options IPUNCH1 (if = 2) and IPUNCH2 (if = 3). All matrices (except M3LP matrix) are punched according to the matrix output card described above. The matrices are punched columnwise with the column number in columns 71-73 and the sequence number (card count from the start of the matrix) in Columns 75-80.
   d. If IPUNCH1 option is for M3LP output, each element is punched in the format described in M3LP writeup. This format can be modified by adjusting the appropriate card in the Fortran source deck as follows: The cards sequenced F5, F05 describe the labeling: F5 is the row label, and F05 the column label. If the columns were to be labeled B, the FORTRAN card sequenced F05 would read A1-1HB; and similarly for a row labeling change. If a label greater than 1 column is desired, it can be added; e.g., column label XXX would be a1 = 3 HXXX. However, the first alphanumeric field width (A1) in all format statements from 125-165 (sequenced F51-F89) would have to be changed to A3. If the row label is increased greater than 1, column (card F5) the second alphanumeric field would need to be changed accordingly. The beginning sequence number of the row, column, of the M3LP output is easier to adjust. Card F6 will equal the row adjustment and F06 the column adjustment. These two cards now read KSTART = 0(F6), LSTART = 0(F06). If the rows are to begin sequencing at 30 and columns at 40 (taking into account the shift due to the objective function), KSTART = 29, LSTART = 38. They are set one less than the desired starting number. The above two adjustments

may seem awkward but since the user seldom changes either of the above, it is deemed better to let him make the adjustment this way for the few times it is done.

3. Tape Output Format, Tape Identification (input/output):
   a. If IPUNCH1 = 3 or IPUNCH2 = 2, the matrix(s) which are taped are written on tape in the format described under punch output (3) with the first record = N. The tapes are BCD, columnwise, If IPUNCH1 = 3, the tape identification is TAPE 4. If IPUNCH2 = 2, the tape identification is TAPE 5.
   b. If ITAPE = 1, 2, 3, the input matrix is on tape as described above and the tape identification is TAPE 3.
   c. If INV = 1, the final matrix is put out for HERP on TAPE 4.
   d. If MULT2 = 1, the composite matrix is put out on TAPE 8.
   e. Tapes 1, 2, 6, 7, 9 are scratch tapes (or disk, depending on user's system).

APPENDIX (to RUTH2 WRITEUP)

Program: The RUTH600 program has the same basic structure as RUTH2; however, it can handle larger matrices because of the programming method used. RUTH2 keeps the entire processing matrix in core. Therefore, the size of the core imposes a restriction. RUTH600 keeps only a part of the processing matrix in at any given time; plus dynamic storage allocation is used. This writeup will only point out the difference in the two programs, so the user will need to be familiar with the RUTH2 writeup to use RUTH600.

Program Differences: The job setup is the same; 2 title cards, control card, data, terminator cards. The format of the control card in RUTH2 is 13I4. The format of the control card in RUTH600 is 10I4, I8, 3I4, the I8 being ISIZE, explained below.

Storage-Control Card: ISIZE replaces ND: ISIZE is the dimension size of a storage vector A. A is the core area utilized by the program to store a part of the processing matrix. For efficiency. A should be as large as feasible. The area A is utilized as follows: The total size, ISIZE, is divided by N (the number of columns, which equals the number of rows) to determine the number of columns which may be kept in core at any one time. For example, if N = 600 and the user sets ISIZE at 30000, only 50 columns could be processed at one time so each matrix operation would require 12 cycles. Obviously ISIZE is dependent on the users core size. The source program requires about 44000_8 core (+ 4K loader, normally).

```
RUTH600,12,40,70000.981010,SHCROEDER
FLOOR(3)
COPYCF( INPUT, PUNCH )
7
*DECK  RUTH
      PROGRAM RUTH600(INPUT,OUTPUT,TAPE3,TAPE4,TAPE11, TAPE90=101 )
C                                                                       A  52
C      INTERINDUSTRY FLOW MAIN PROGRAM                                  A   3
C      N=MATRIX ORDER --- MUST NOT BE GREATER 600 THIS DECK             A
C      ISIZE=SIZE OF DIMENSION STATEMENT OF A IN MAIN SOURCE PROGRAM
C      NP=SIZE OF AGGREGATED MATRIX   IF BLANK OR 0 NO AGGREGATION      A   6
C      KHOICE=CHOICE ---SEE MASTER WRITEUP                              A   7
C      ITAPE=0  MATRIX ON CARDS                                         A   8
C      ITAPE=1  MATRIX ON TAPE IN BINARY
C      IPRINT=1 IF PRINT ORIG MATRIX,=B OTHERWISE                       A  14
C      IPUNCH1=0 NO OPTION
C      IPUNCH1=1 PUT FINAL MATRIX ON TAPE4, BINARY.
C      MX=NO VECTORS A * BY IF AX IS COMPUTED                           A  22
C      NTRANV=1=COL VEC*TRANSPOSE ELSE * REG MATRIX                     A  23
C      MULT=POS NO FINAL MATRIX TRANSPOSED TIMES DIAGONAL               A  24
C      MULT = NEG NO FINAL MATRIX TIMES DAGONAL AMATRIX                 A  25
C      WHERE NO = NUMBER OF VECTORS TO BE MULTIPLIED TIMES FINAL MATRIX A  26
C      EACH VECTOR IS * FINAL MATRIX8 PRINTED8 SEQUENCED                A  27
C      IT IS ASSUMED BUT NOT TESTED FOR THAT NO IS LESS THAN 100        A  28
C      MULT = BLK OR 0 BYPASS OPTION                                    A  29
C      MULTP OPTIONS ONLY DEAL WITH THE OUTPUT OF MULT OPTION           A  30
C      MULTP=1 PRINT REGULAR MATRIX + ROWSUM
C      MULTP=2 PUT MATRIX AND ROW SUMS ON TAPE11, CONSEQUETIVELY.
C      MULTP=3 PRINT RESULTS AND PUT ON TAPE11.
C      INPUT CARDS FOLLOWING THE MAIN CONTROL CARD GIVIVING OPTIONS ETC  A  40
C      INPUT FORMAT ON FIRST CARD AFTER CONTROL CARD,OUTPUT ON SECOND \ A  41
C      TAPES 3 FLOW MATRIX INPUT                                        A  44
C            4 FINAL MATRIX OUTPUT (IPUNCH1 OPTION)                     A  45
C            11 USED FOR MULTP OPTION
C                                                                       A  56
      COMMON V1(600),V2(600),V3(600),TITLE(18),II7
      COMMON A(800)
      COMMON /3/ HOLD(600,7)
      COMMON /RNDMIO/ FILENAM, JGO
      DIMENSION IREAD(8)
      DIMENSION NSORT(600)
      EQUIVALENCE (NSORT,V3)                                            A  63
      DATA FILENAM / 6LTAPE90 /
10    READ 320, (TITLE(I),I=1,16)
      READ 330, N,ISIZE,NP,KHOICE,ITAPE,IPRINT,IPUNCH1,MX, NTRANV, MULT,
     1 MULTP
      IF (N.EQ.9999) GO TO 260                                          A  69
       II=ISIZE/N
      IF(II.GT.N) II=N
      II7=(II/7) * 7
      PRINT 350                                                         A  70
      PRINT 360, N,ISIZE,NP,KHOICE,ITAPE,IPRINT,IPUNCH1,MX, NTRANV, MULT
     1, MULTP
      IF (((N.GT.0).AND.(II.GT.0)).AND.((IABS(KHOICE).GE.1).AND.(IABS(KH A  73
     1OICE).LE.5)).AND.((NP.GE.0).AND.(NP.LE.N)).AND.((IPRINT.EQ.0).OR.( A  74
     2IPRINT.EQ.1)).AND.((IPUNCH1.EQ.0) .OR.(IPUNCH1.EQ.1)).AND.((NTRANV A
     3.EQ.0).OR.(NTRANV.EQ.1)).AND.((ITAPE.EQ.0).OR.(ITAPE.EQ.1)).AND.(I
     4SIZE.GT.0).AND.((MULTP.GE.0).AND.(MULTP.LE.3))) GO TO 20
      PRINT 400                                                         A  80
      CALL EXIT                                                         A  81
20    CONTINUE                                                          F  22
      DO 50 I=1,N                                                       A 104
      V1(I)=0.                                                          A 105
      V2(I)=0.                                                          A 106
50    V3(I)=0.                                                          A 107
      DO 60 I=1,ISIZE                                                   A 108
```

```
60    A(I)=0.                                                          A 109
      IF(  KHOICE .LT. 5 ) GO TO 85

C         KHOICE = 5   READ IN MATRIX PRODUCED IN AN EARLIER RUN
C                      AND MULTIPLY BY MX COLUMN VECTORS. (INPUT TAPE IS
C                      TAPE3. )

      READ(3) TITLE
      PRINT 65, TITLE
 65   FORMAT( 1H1, 20X, 8A10, /20X, 8A10, /20X, 2A10 )
      READ(3) (V1(I), I=1,N)
      PRINT 67, (I, V1(I), I=1,N)
 67   FORMAT(  //, 20X, *INPUT MATRIX WAS FORMED BY MULTIPLYING AN EARLI
     1ER FINAL MATRIX BY THE FOLLOWING VECTOR*, /20X, *USED AS A DIAGONA
     2L MATRIX*, /(5X, 7(I5, F12.7)))

C        IF MX .GT. 4, PUT INPUT MATRIX ON FILENAM AT LOC.

      KCH = 5
      IF( MX .LE. 4 ) GO TO 80
      JGO = 0
      DO70 I=1,N
      READ(3) (V1(I), I=1,N)
      CALL RANDIO( V1, V1(N), LLOC )
      IF( I .EQ. 1 ) LOC = LLOC
70    CONTINUE                                                         C  26
      CALL RETURNS(3)
      KCH = 4
      KSECT = N/64 + 1
 80   CONTINUE
      CALL MULTV( MX, NTRANV, A, N, II, LOC, KSECT, KCH )
      GO TO 10
 85   CONTINUE
      IF( IABS(KHOICE) .NE. 4 ) GO TO 31
      NH=NP                                                            A  83
      IF (NH.EQ.0) NH=N                                                A  84
      IF (NH.GE.150) GO TO 30                                          A  85
      PRINT 270                                                        A  86
30    CONTINUE                                                         C  16
      ND = NH / 3 + 1
      IF( ND*ND .LE. ISIZE ) GO TO 154
      PRINT 152, ISIZE, N
 152  FORMAT( ///10X, *THE DIMENSION OF A,*, I6,*, IS NOT LARGE ENOUGH F
     1OR HERP SINCE A IS OF RANK*, I4 )
      CALL EXIT                                                        A 114
 154  CONTINUE
 31   CONTINUE
      IF( ITAPE .GT. 0 ) GO TO 33
      READ 320, (IREAD(I),I=1,8)
 33   CONTINUE

C        READ IN INPUT MATRIX COLUMNWISE, PRINT IF DESIRED, AND STORE IN
C        FILENAM.

      CALL MATIN (A,N,II,LOC,IPRINT,IREAD,ITAPE)
      KSECT = N/64 + 1
      LN=N                                                             A 124
      IF (NP.EQ.0) GO TO 90                                            A 125
      LN=NP                                                            A 126
      II=ISIZE/NP                                                      A 127
      IF (II.GT.NP) II=NP                                              A 128
      II7=(II/7)*7                                                     A 129
      CALL AGGREG (NSORT, NP, A, N, II, LN, KSECT, LOC )
90    IF (IABS(KHOICE)-1) 110,100,110                                  A 134
100   TITLE(17)=6HA MATR
      TITLE(17) = 8HA MATRIX
      TITLE(18) = 10H
      GO TO 180                                                        A 137
```

```
C         /KHOICE/ GE 2   NORMALIZE INPUT MATRIX - GET ABAR.

  110  CONTINUE
       CALL NORMIZ (NSORT, NP, A, N, II, LN, KSECT, LOC )
       IF (NP.GT.0) N=NP
       IF (IABS(KHOICE)-2) 130,120,130
  120  TITLE(17) = 10HA NORMALIZ
       TITLE(18) = 10HED
       GO TO 180

C         /KHOICE/ GE 3   CALCULATE I-ABAR.

 130   CALL IMA (A, N, II, KSFCT, LOC )
       IF (IABS(KHOICE)-3) 150,140,150
  140  TITLE(17) = 10HI-A MATRIX
       TITLE(18) = 10H
       GO TO 180
 150   CONTINUE

C         /KHOICE/ EQ 4   CALCULATE (I-ABAR) INVERSE.

       IPRINT = 0
       IF(  KHOICE .LT. 0 ) IPRINT = 1
       IF( IPUNCH1 .EQ. 1 ) IPRINT = IPRINT + 2
       CALL HERP( A, N, ND, IPRINT, LOC, KSECT )
       GO TO 200
 180   IF (KHOICE) 190,195,195
  190  CONTINUE
       JGO = -1
       LLOC = LOC
       I1 = 1
       I2 = 7
       K1 = 1
       K2 = N
       KEND = 7*N
       IPP = 7
       NC = MINO( 55, N )
       IPAGE = 1
       DO194 J=1,N
       CALL RANDIO( A(K1), A(K2), LLOC )
       LLOC = LLOC + KSECT
       IF( IPUNCH1 .EQ. 1 ) WRITE(4) (A(I), I=K1,K2)
       IF( K2 .LT. KEND ) GO TO 192
       CALL PRINT( N, IPP, NC, A, N, IPAGE, I1, I2, TITLE )
       K1 = 1
       K2 = N
       I1 = I2 + 1
       I2 = I2 + 7
       I2 = MINO(N,I2)
       GO TO 194
  192  CONTINUE
       K1 = K2 + 1
       K2 = K2 + N
  194  CONTINUE
       IF( K1 .EQ. 1 ) GO TO 200
       IPP = I2 - I1 + 1
       CALL PRINT( N, IPP, NC, A, N, IPAGE, I1, I2, TITLE )
       GO TO 200
  195  CONTINUE
       IF( IPUNCH1 .EQ. 0 ) GO TO 200
       JGO = -1
       LLOC = LOC
       DO196 J=1,N
       CALL RANDIO( A, A(N), LLOC )
       LLOC = LLOC + KSECT
       WRITE(4) (A(I), I=1,N)
  196  CONTINUE
 200   IF (MULT.NE.0) CALL MULTPY (KT, MULT, MULTP, A, N, II, KSECT, LOC)
       IF( MX ) 260, 260, 240
```

```
 240  CONTINUE
      CALL MULTV( MX, NTRANV, A, N, II, LOC, KSECT,  KHOICE )
 260  PRINT 310                                                        A 183
      CALL EXIT                                                        A 184
C                                                                      A 185
C                                                                      A 186
C                                                                      A 187
 270  FORMAT(*1 MATRIX TO BE INVERTED IS LESS 150 USE RUTH2*)
 310  FORMAT (12H1EOJ RUTH600)                                         A 192
 320  FORMAT (8A10)                                                    A 193
 330  FORMAT (I4,I8,12I4)                                              A 194
 350  FORMAT(*1  N     ISIZE       NP    KHOICE     ITAPE    IPRINT   IPUNCH1
     1       MX    NTRANV      MULT     MULTP * )
 360  FORMAT(1H ,I3,13I9)
 400  FORMAT (35H CONTROL CARD ERROR END OF RUN ****)                  A 204
      END                                                              A 205-
*DECK  AGGREG
      SUBROUTINE AGGREG (NSORT, M, A, N, II, LN, KSECT, LOC )
      COMMON V1(600),V2(600),V3(600),TITLE(18),II7
      COMMON /RNDMIO/ FILENAM, JGO
      DIMENSION A(LN,II)                                               H   3
      DIMENSION NS(600), NSORT(600), HOLD(600)                         H   4
      EQUIVALENCE (NS,V1), (HOLD,V2)                                   H   5
C     AGGREGATE MATRIX ON FILENAM (COLWISE) ONTO FILENAM (COLWISE)     H
C     M=AGGREGATION ORDER                                              H   7
      PRINT 110                                                        H   8
C     GROUPINGS TERMINATE WITH A BLANK                                 H   9
C     READ IN GROUPING CARDS--SET UP VECTOR OF PLACEMENT               H  10
 10   READ 120, NK,(NS(I),I=1,23)                                      H  11
      PRINT 130, NK,(NS(I),I=1,23)                                     H  12
      I=1                                                              H  13
      IF (NK.EQ.0) GO TO 30                                            H  14
 20   IF (NS(I).EQ.0) GO TO 10                                         H  15
      NSS=NS(I)                                                        H  16
      NSORT(NSS)=NK                                                    H  17
      I=I+1                                                            H  18
      GO TO 20                                                         H  19
C     HAVE VECTOR OF PLACEMENTS SET UP--NOW AGGREGATE                  H  20
C     IF M.GT.II, SEVERAL PASSES MAY HAVE TO BE MADE. ONE TO SUM COLS 1-II,
C     ONE TO SUM COLS (II+1)-(2II), ETC.
 30   CONTINUE                                                         C  16
      MB=1                                                             H  25
      ME=MINO(M,II)                                                    H  26
C                                                                      L  39
C     CLEAR MATRIX AREA AS SUMMING INTO IT                             H  27

 40   DO 50 I=1,M                                                      H  28
      DO 50 J=1,II                                                     H  29
 50   A(I,J)=0.                                                        H  30
      LD=ME-MB+1                                                       H  31
      JGO = -1
      LLOC = LOC
      DO 80 J=1,N                                                      H  32
C                                                                      A  53
C     READ IN A COL OF ORIG MATRIX INTO HOLD AND SUM ITS ELEMENTS      H  33
C     INTO APPROPRIATE BUCKETS OF A                                    H  34
C                                                                      A  55
      CALL RANDIO( HOLD, HOLD(N), LLOC )
      LLOC = LLOC + KSECT
      K=NSORT(J)                                                       H  36
      IF ((K.GE.MB).AND.(K.LE.ME)) GO TO 60                            H  37
C     ABOVE ELIMINATES K=0 (DELETIONS) + ALL THOSE FALL OUTSIDE RANGE  H  38
      GO TO 80                                                         H  39
 60   CONTINUE                                                         C  21
      KK = K - MB + 1
C     ABOVE ADJUSTS STORAGE SLOT OF COL ON 2ND PASS                    H  44
      DO 70 I=1,N
      L=NSORT(I)                                                       H  41
      IF (L.EQ.0) GO TO 70                                             H  42
      A(L,KK) = A(L,KK) + HOLD(I)
```

```
C     SUM INTO AGREGATED MATRIX*PIECE* FROM ORIG COLUMN
70    CONTINUE
80    CONTINUE
      JGO = 0
      DO 90 J=1,LD
      CALL RANDIO( A(1,J), A(M,J), KLOC )
      IF( J .EQ. 1 .AND. MB .EQ. 1 ) LOCA   = KLOC
90    CONTINUE
      IF( ME .EQ. M ) GO TO 100
      ME=MINO(M,ME+II)
      MB=MB+II
      GO TO 40
 100  CONTINUE
      KSECT = M/64 + 1
      LOC = LOCA
C     SET N=M AS IN REST OF PROGRAM M IS ORDER USED
      RETURN
C
C
110   FORMAT (1H1,50H    GROUPING OF VARIABLES IN THE AGGREGATE MATRIX /
     1//1H0,40X,9HVARIABLES//)
120   FORMAT (24I3)
130   FORMAT ( 5X,6HGROUP ,I4,6X,23I4)
      END
*DECK  HERP
      SUBROUTINE HERP( A, N, ND, IPRINT, LOCA, KSECTA )

C        MATRIX OF RANK N IS STORED COLUMNWISE ON FILENAM AT LOCA. WILL PUT
C        INVERSE BACK AT THIS LOCATION.

      COMMON V1(600),V2(600),V3(600),TITLE(18),II7
      COMMON /1/ SET1(510), SET2(510), SET3(510), HOLO(510)
      COMMON /2/ NUMROS, NUMROW, ROW(510,2)
      COMMON /RNDMIO/ FILENAM, JGO
      DIMENSION A(ND,ND)
      DIMENSION INT(3), IWHERE(3), KON(3,24), M(2,24), MINUSQ(24), ADDQ(
     124)
      DIMENSION LOC(20)
      DIMENSION INDKON(72)
      EQUIVALENCE (KON(1,1),INDKON(1))
      LOGICAL MINUSQ,ADDQ,REPEAT
C
C     THIS SUBROUTINE INVERTS THE MATRIX ORIGINALLY ON DISK 1 ACCORDING
C     THE ALGORITHM GIVEN IN THE MAMMOTH MANUAL, APPENDIX 3.
C     HERE 'A' IS SUB-MATRIX 1,1. 'B' IS SUB-MATRIX 1,2. AND SO ON.
C     THE 'KON' ARRAY SHOWS THE ORDER OF THE 24 SUB-MATRIX MULTIPLICATIO
C     FOR EXAMPLE, IN THE FIRST MULTIPLICATION, 'C'--3,1--IS MULTIPLIED
C     A--1,1.
C     'M' TELLS WHICH DISK EACH OF THE TWO MATRICES IS ON. THE PRODUCT S
C     MATRIX MUST GO ONTO THE DISK THAT THE FIRST OF THE 2 SUB-MATRICES
C     NOT ON. IF THE PRODUCT IS MADE NEGATIVE, 'MINUSQ' FOR THAT MULTIPL
C     CATION IS 'T'. IF THE PRODUCT IS ADDED TO ANOTHER MATRIX, ADDQ IS
      DATA LOC / 20*0 /
      DATA KON/3,1,1,2,1,1,1,1,3,3,1,3,2,1,3,1,1,2,2,1,2,3,1,2,3,2,2,3,2
     1,3,2,3,3,2,2,3,3,3,2,2,3,2,1,2,3,1,3,3,1,2,2,1,3,2,2,2,1,3,2,1,1,2
     2,1,2,3,1,3,3,1,1,3,1/
      DATA M/2,2,2,2,2,2,1,2,1,2,2,2,1,2,1,2,2,2,1,2,2,2,2,1,1,1,1,1,1,2
     1,1,2,1,2,1,2,2,1,2,1,2,1,2,2,2,2,2,2/
      DATA MINUSQ/.TRUE.,.TRUE.,.TRUE.,.FALSE.,.FALSE.,.TRUE.,.FALSE.,.F
     1ALSE.,.FALSE.,.TRUE.,.TRUE.,.FALSE.,.TRUE.,.TRUE.,.FALSE.,.FALSE.,
     2.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./
      DATA ADDQ/.FALSE.,.FALSE.,.FALSE.,.TRUE.,.TRUE.,.FALSE.,.TRUE.,.TR
     1UE.,.FALSE.,.TRUE.,.FALSE.,.FALSE.,.FALSE.,.TRUE.,.FALSE.,.TRUE.,.
     2FALSE.,.TRUE.,.FALSE.,.FALSE.,.TRUE.,.TRUE.,.TRUE.,.TRUE./
      LIT(IJ,IK)=(((IJ-1)/NUMROS)+1)+((IK-1)*9)
      LOK(KA,KB)=((KA-1)*3+(KB-1))*NUMROS+1
```

```
C          IPRINT = 0   NO PRINT-OUT.
C          IPRINT = 1   PRINT THE CALCULATED INVERSE.
C          IPRINT = 2   PUT THE INVERSE ON TAPE4, BINARY.
C          IPRINT = 3   PRINT THE INVERSE AND PUT IT ON TAPE4, BINARY.

      IPR = 0
      IF( IPRINT .EQ. 1 .OR. IPRINT .EQ. 3 ) IPR = 1
      CALL SECOND (TIMEE)                                               A  69
      PRINT 510, TIMEE                                                  A  70
      INT(1)=(  N  -1)/3+1
      INT(3)=(  N  -INT(1))/2
      INT(2)=  N  -INT(1)-INT(3)
      NMPERO = MINO( 510/INT(1), INT(1) )
      NUMROS=(INT(1)-1)/NMPERO+1                                        A  75

C     NUMROW=LENGTH OF SEGMENT IN ROW                                   A  76

      NMAX=MAXO(INT(1),INT(2),INT(3))                                   A  77
      NUMROW=NMAX*NMPERO                                                A  78
      KSECT = NUMROW / 64  +  1
C                                                                       A  82
C     PUT MATRIX ONTO DISK 2, BY NINTHS                                 A  83
C                                                                       D   5

      IBEG=0                                                            A  84
 40   CONTINUE
      DO 110 KC=1,3                                                     A  85
      INTKC=INT(KC)                                                     A  88
      LLOC = LOCA
      DO 100 KA=1,3                                                     A  89
      IAROW=1                                                           A  90
      IGO = 0
      IPROW=LOK(KA,KC)
      IT=LIT(IPROW,2)                                                   A  92
 50   JBEG=0                                                            A  94
      JGO = -1
      DO 70 I=1,NMPERO                                                  A  95

C          GET INPUT FROM FILENAM STARTING AT LOCA.

      CALL RANDIO( SET1, SET1(N), LLOC )
      LLOC = LLOC + KSECTA
      DO 60 J=1,INTKC                                                   A  98
      IX=J+IBEG                                                         A  99
      JX=J+JBEG                                                         A 100
 60   ROW(JX,1) = SET1(IX)
      IAROW=IAROW+1                                                     A 102
      IF (IAROW-INT(KA)) 70,70,80                                       A 103
 70   JBEG=JBEG+INTKC                                                   A 104
   80 CONTINUE
      JGO = 0
      CALL RANDIO( ROW(1,1), ROW(NUMROW,1), JLOC )
      IF( IGO .EQ. 0 ) LOC( IT ) = JLOC
      IGO = 1
      IF( IAROW-INT(KA) ) 50, 50, 97
  97  CONTINUE
 100  CONTINUE                                                          A 110
 110  IBEG=IBEG+INT(KC)                                                 A 111
      REPEAT=.FALSE.                                                    A 112
 120  CONTINUE                                                          A 113
C                                                                       A 115
C     DO 24 MULTIPLICATIONS, ACCORDING TO 'KON','M','MINUSQ',AND 'ADDQ' A 116
C                                                                       A 114
      DO 380 KOUNT=1,24                                                 A 117
      KA=KON(1,KOUNT)                                                   A 118
      KB=KON(2,KOUNT)                                                   A 119
      KC=KON(3,KOUNT)                                                   A 120
```

```
      INTKA=INT(KA)                                                     A 121
      INTKB=INT(KB)                                                     A 122
      INTKC=INT(KC)                                                     A 123
      IWHERE(1)=M(1,KOUNT)                                              A 124
      IWHERE(2)=M(2,KOUNT)                                              A 125
      IWHERE(3)=3-IWHERE(1)                                             A 126
      SIGN = 1.
      IF (MINUSQ(KOUNT)) SIGN = -1.
      ASSIGN 290 TO IADDQ                                               A 129
      IF (ADDQ(KOUNT)) ASSIGN 310 TO IADDQ                              A 130
C                                                                       A 131
C     PUT MULTIPLIER MATRIX (KB BY KC) INTO CORE                        A 132
C                                                                       B   5
      IGROW=LOK(KB,KC)
      IT=LIT(IGROW,IWHERE(2))                                           A 134
      JGO = -1
      JLOC = LOC(IT)
      IF (KOUNT.EQ.1) GO TO 130                                         A 136
      IF ((KB.EQ.KON(2,KOUNT-1)).AND.(KC.EQ.KON(3,KOUNT-1))) GO TO 220  A 137

C        GET CURRENT NINTH THAT IS NEEDED OFF FILENAM. IF ABOVE
C        IF STATEMENT IS TRUE, CURRENT NINTH IS ALREADY IN CORE.

 130  CONTINUE
      DO 140 IAROW=1,INTKB,NMPERO
      CALL RANDIO( ROW, ROW(NUMROW), JLOC )
      JLOC = JLOC + KSECT
      K=0                                                               A 141
      IARWPL=IAROW+NMPERO-1                                             A 142
      IARWPL = MIN0( IARWPL,INTKB)
      DO 140 J=IAROW,IARWPL                                             A 143
      DO 140 I=1,INTKC                                                  A 144
      K=K+1                                                             A 145
140   A(J,I)=ROW(K,1)                                                   A 146
C                                                                       A 147
C     INVERT MATRIX IN CORE, IF NECESSARY, AND PUT INVERSE BACK ON DISK(  A 148

      IF ((KOUNT-1)*(KOUNT-9)*(KOUNT-11)) 220,150,220                   A 149
 150  CONTINUE
      CALL TINYIN (INTKC,ISING,ND,A)
      IF (ISING.EQ.0) GO TO 170                                         A 151
      IF ((KOUNT.NE.1).OR.(REPEAT)) GO TO 450                           A 152
      REPEAT=.TRUE.                                                     A 153
      DO 160 I=1,72                                                     A 154
160   INDKON(I)=4-INDKON(I)                                             A 155
      GO TO 120                                                         A 156
 170  CONTINUE

C        PUT INVERSE BACK ON FILENAM. WILL WRITE IT OUT TWICE IF KOUNT = 1 OR 11
C        AND ONLY ONCE IF KOUNT = 9.

      ASSIGN 200 TO IBOTHQ
      IPROW=LOK(KB,KC)
      LT=LIT(IPROW,2)                                                   A 160
      JLOC = LOC(LT)
      IF( KOUNT .EQ. 9 ) GO TO 178
      ASSIGN 190 TO IBOTHQ
      IT=LIT(IPROW,1)                                                   A 159
      IF( LOC(IT) .EQ. 0 ) GO TO 175
      IGO = 1
      KLOC = LOC(IT)
      KGO = 1
      GO TO 178
 175  CONTINUE
      IGO = 2
      KGO = 0
```

```
 178  CONTINUE
      DO 210 IAROW=1,INTKB,NMPERO                                        A 164
      K=0                                                                A 165
      IARWPL=IAROW+NMPERO-1                                              A 166
      IARWPL = MINO( IARWPL, INTKB )
      DO 180 J=IAROW,IARWPL                                              A 167
      DO 180 I=1,INTKC                                                   A 168
      K=K+1                                                              A 169
180   ROW(K,1)=A(J,I)                                                    A 170
      GO TO IBOTHQ, (190,200)                                            A 171
 190  CONTINUE
      JGO = KGO
      CALL RANDIO( ROW, ROW(NUMROW), KLOC )
      GO TO ( 205, 195, 208 ), IGO
 195  CONTINUE
      LOC(IT) = KLOC
      IGO = 3
      GO TO 208
 205  CONTINUE
      KLOC = KLOC + KSECT
 208  CONTINUE
 200  CONTINUE
      JGO = 1
      CALL RANDIO( ROW, ROW(NUMROW), JLOC )
      JLOC = JLOC + KSECT
210   IPROW=IPROW+1                                                      A 174
C                                                                        A 175
C     MULTIPLY THRU BY THE MULTIPLICAND (KA BY KB) AND PUT INTO PRODUCT  A 176

220   IPROW=LOK(KA,KC)
      IGROW=LOK(KA,KB)
      LT=LIT(IGROW,IWHERE(1))                                            A 181
      JLOC = LOC(LT)
      KGO = 0
      IT=LIT(IPROW,IWHERE(3))                                            A 180
      IF( LOC(IT) .EQ. 0 ) GO TO 345
      KGO = 1
      KLOC = LOC(IT)
  345 CONTINUE                                                         MSLIB
C                                                                        A 223
C     IN 5 CASES, PRODUCT MUST GO ON DISK 2, AS WELL AS DISK 1           A 224

      IWRITE = 0
      IF ((KOUNT-1)*(KOUNT-12)*(KOUNT/22-1)) 370,360,370
 360  CONTINUE
      IWRITE = 1
      KT=LIT(IPROW,3-IWHERE(3))
      LLOC = LOC(KT)
 370  CONTINUE
      DO 340 IAROW=1,INTKA,NMPERO                                        A 184
      JGO = -1
      CALL RANDIO( ROW, ROW(NUMROW), JLOC )
      JLOC = JLOC + KSECT
      JBEG=0                                                             A 187
      IBEG=0                                                             A 188
      MPERO = NMPERO
      MPERO = MINO( INTKA-IAROW+1, NMPERO )
      DO 280 I=1,MPERO
      DO 270 L=1,INTKC                                                   A 190
      LX=L+IBEG                                                          A 191
      SUM=0.0                                                            A 194
      DO230 J=1,INTKB
      JX=J+JBEG                                                          A 196
230   SUM=SUM+ROW(JX,1)*A(J,L)                                           A 197
      ROW(LX,2) = SUM * SIGN
270   CONTINUE                                                           A 203
      JBEG=JBEG+INTKB                                                    A 204
```

```
280    IBEG=IBEG+INTKC                                                   A 205
       NMROW = NUMROW
       IF( MPERO .LE. NMPERO ) NMROW = MPERO*INTKC
       GO TO IAOOQ, (290,310)                                            A 206
290    DO 300 L=1,NMROW
300    ROW(L,1)=ROW(L,2)                                                 A 208
       GO TO 330                                                         A 209
 310   CONTINUE
       CALL RANDIO( ROW, ROW(NUMROW), KLOC )
       DO 320 L=1,NMROW
320    ROW(L,1)=ROW(L,1)+ROW(L,2)                                        A 212
 330   CONTINUE
       JGO = KGO
       CALL RANDIO( ROW, ROW(NUMROW), KLOC )
       IF( JGO .EQ. 0 .AND. IAROW .EQ. 1 ) LOC(IT) = KLOC
       KLOC = KLOC + KSECT
       IF( IWRITE .EQ. 0 ) GO TO 340
       JGO = 1
       CALL RANOIO( ROW, ROW(NUMROW), LLOC )
       LLOC = LLOC + KSECT
  340  CONTINUE
380    CONTINUE                                                          A 236
C                                                                        A 237
C         AFTER 24 MULTIPLICATIONS, PUT INVERSE BACK ON FILENAM AT LOCA,
C         ROWWISE.

       IF( IPR    .EQ. 0 ) GO TO 385
       IPAGE = 1
C
C         JPA AND JPB ARE THE RANGE OF THE ROWS CURRENTLY BEING STOREO.
C         JPC IS THE INDEX OF THE CURRENT ELEMENT BEING STORED IN A.
C         IPP IS THE NUMBER OF ROWS PER PAGE.
C         NC IS THE NUMBER OF COLUMNS PER PAGE.

       JPA = 1
       JPB = 9
       JPC = 1
       IPP = 9
       NC = MINO( 55,N )
       N9 = 9*N
       TITLE(17) = 10HINVERSE OF
       TITLE(18) = 10H MATRIX
 385   CONTINUE
       00387 I=1,N
       ROW(I) = 0.
 387   CONTINUE
       IR = 1
       I1=INT(1)                                                         A 241
       I2=I1+INT(2)                                                      A 242
       I3=I2+INT(3)                                                      A 243
       I1P=I1+1                                                          A 244
       I2P=I2+1                                                          A 245
       LLOC = LOCA
       00 440 KC=1,3                                                     A 246
       IGROW1=LOK(KC,1)
       IGROW2=LOK(KC,2)
       IGROW3=LOK(KC,3)
       IT1=LIT(IGROW1,2)                                                 A 250
       IT2=LIT(IGROW2,2)                                                 A 251
       IT3=LIT(IGROW3,2)                                                 A 252
       LOC1 = LOC(IT1)
       LOC2 = LOC(IT2)
       LOC3 = LOC(IT3)
       LOOP=INT(KC)                                                      A 256
       ICT=0                                                             A 257
       00 430 II=1,NUMROS                                                A 258
       JGO = -1
```

```
      CALL RANDIO( SET1, SET1(NUMROW), LOC1 )
      CALL RANDIO( SET2, SET2(NUMROW), LOC2 )
      CALL RANDIO( SET3, SET3(NUMROW), LOC3 )
      LOC1 = LOC1 + KSECT
      LOC2 = LOC2 + KSECT
      LOC3 = LOC3 + KSECT
      J1=0
      J2=0
      J3=0
      DO 420 K=1,NMPERO
      IF (ICT.GE.LOOP) GO TO 440
      ICT=ICT+1
      DO 390 J=1,I1
      J1=J1+1
  390 HOLD(J)=SET1(J1)
      DO 400 J=I1P,I2
      J2=J2+1
  400 HOLD(J)=SET2(J2)
      DO 410 J=I2P,I3
      J3=J3+1
  410 HOLD(J)=SET3(J3)
      JGO = 1

C        HAVE ONE COLUMN OF THE INVERSE IN HOLD.

      CALL RANDIO( HOLD, HOLD(N), LLOC )
      LLOC = LLOC + KSECTA
      DO411 I=1,N
      ROW(IR) = ROW(IR) + HOLD(I)
  411 CONTINUE
      IR = IR + 1
      IF( IPRINT .GT. 1 ) WRITE(4) (HOLD(I), I=1,N)
      IF( IPR    .EQ. 0 ) GO TO 416
      J = JPC
      DO412 I=1,N
      A(J) = HOLD(I)
      J = J + 1
  412 CONTINUE
      JPC = J
      IF( JPC .LT. N9 ) GO TO 416
      CALL PRINT( N, IPP, NC, A, N, IPAGE, JPA, JPB, TITLE )
      JPC = 1
      JPA = JPB + 1
      JPB = JPB + IPP
      JPB = MIN0(JPB,N)
  416 CONTINUE
  420 CONTINUE
  430 CONTINUE
  440 CONTINUE
      IF( IPR .EQ. 0 .OR. JPC .EQ. 1 ) GO TO 445
      IPP = JPB - JPA + 1
      CALL PRINT( N, IPP, NC, A, N, IPAGE, JPA, JPB, TITLE )
  445 CONTINUE
      PRINT 448, (I, ROW(I), I=1,N)
  448 FORMAT( 1H1, 10X, *THE COLUMN SUMS ARE AS FOLLOWS*, / 12X, *COL*,
     1 14X, *SUM*, /(10X, I5, 3X, F17.7) )
      IF( IPRINT .GT. 1 ) WRITE(4) (ROW(I), I=1,N)
      CALL SECOND (TIMEE)
      PRINT 510, TIMEE
C     THIS ROUTINE NEEDS THE BIG FIX BUT FOR THE MOMENT
C     DIMENSIN DOES NOT FIT WITH CODE
      RETURN
  450 PRINT 550
      CALL EXIT
C
  510 FORMAT (*1TIME=*,E15.4)
  550 FORMAT (*SINGULAREND*)
      END
```

```
*DECK  IMA
      SUBROUTINE IMA (A, N, II, KSECT, LOC )

C         SUBROUTINE COMPUTES I-A.

      COMMON V1(600),V2(600),V3(600),TITLE(18),II7
      COMMON /RNDMIO/ FILENAM, JGO
      DIMENSION A(N,II)                                                 J   3
      ME=MINO(N,II)                                                     J   8
      MB=1                                                              J   9
      MD=0                                                              J  10
      LLOC = LOC
C     READ IN LD COL                                                    J  11
10    LD=ME-MB+1                                                        J  12
      JGO = -1
      KLOC = LLOC
      DO 20 J=1,LD                                                      J  13
      CALL RANDIO( A(1,J), A(N,J), LLOC )
      LLOC = LLOC + KSECT
20    CONTINUE                                                          J  15
C     A(I,J)=-A(I,J) FOR I NOT.EQ.J ---A(I,J)=1-A(I,J) FOR I=J          J  16
      DO 50 J=1,LD                                                      J  17
      MD=MD+1                                                           J  18
      DO 40 I=1,N                                                       J  19
      IF (MD.EQ.I) GO TO 30                                             J  20
      A(I,J)=-A(I,J)                                                    J  21
      GO TO 40                                                          J  22
30    A(I,J)=1.-A(I,J)                                                  J  23
40    CONTINUE                                                          J  24
50    CONTINUE                                                          J  25
C     WRITE OUT COLUMNS COMPLETED ON FILENAM
      JGO = 1
      DO 60 J=1,LD                                                      J  27
      CALL RANDIO( A(1,J), A(N,J), KLOC )
      KLOC = KLOC + KSECT
60    CONTINUE                                                          J  29
      IF (ME.EQ.N) GO TO 70                                             J  30
      ME=MINO(N,ME+II)                                                  J  31
      MB=MB+II                                                          J  32
      GO TO 10                                                          J  33
70    CONTINUE                                                          C  26
      RETURN                                                            J  36
      END                                                               J  37-
*DECK  MATIN
      SUBROUTINE MATIN (A,N,II,LOC,IPRINT,IREAD,ITAPE)
      COMMON V1(600),V2(600),V3(600),TITLE(18),II7
      COMMON /RNDMIO/ FILENAM, JGO
      DIMENSION A(N,II)
      DIMENSION IREAD(8)
      JGO = 0
      IF( IPRINT ) 3, 3, 2
 2    CONTINUE
      TITLE(17) = 10HINPUT MATR
      TITLE(18) = 10HIX
      I1 = 1
      I2 = 7
      K = 1
      KEND = 7*N
      IPP = 7
      IPAGE = 1
      NC = MINO( 55, N )
 3    CONTINUE
      DO 40 J=1,N                                                       B   5
      IF (ITAPE.GE.1) GO TO 10                                          B   6
      READ IREAD, (V1(I),I=1,N)                                         B   7
      GO TO 30                                                          B   8
 10   CONTINUE
      READ (3)(V1(I),I=1,N)
```

```
   30 CONTINUE
      CALL RANDIO( V1, V1(N), LLOC )
      IF( J .EQ. 1 ) LOC    = LLOC
      IF( IPRINT ) 40, 40, 32
 32   CONTINUE
      IF( K .LT. KEND ) GO TO 34
      CALL PRINT( N, IPP, NC, A, N, IPAGE, I1, I2, TITLE )
      I1 = I2 + 1
      I2 = I2 + 7
      I2 = MINO(N,I2)
      K = 1
 34   CONTINUE
      DO36 I=1,N
      A(K) = V1(I)
      K = K + 1
 36   CONTINUE
40    CONTINUE                                                  B  14
      IF( ITAPE .GE. 1 ) CALL RETURNS(3)
      IF( IPRINT ) 60, 60, 50
50    CONTINUE                                                  C  20
      IPP = I2 - I1 + I
      CALL PRINT( N, IPP, NC, A, N, IPAGE, I1, I2, TITLE )
60    RETURN                                                    B  19
      END                                                       B  20-
*DECK  MULTPY
      SUBROUTINE MULTPY (KT, MULT, MULTP, A, N, II, KSECT, LOC )
      COMMON V1(600),V2(600),V3(600),TITLE(18),II7
      COMMON /3/ HOLD(600,7)
      COMMON /RNOMIO/ FILENAM, JGO
      DIMENSION A(N,II)                                         K   3
      DIMENSION X(600), SUM(600)
      EQUIVALENCE (X,V1), (SUM,V3)

C        MULT LT 0  -  MULTIPLY MATRIX BY DIAGONAL MATRIX.
C        MULT GT 0  -  MULTIPLY TRANSPOSE OF MATRIX BY DIAGONAL MATRIX.
C        /MULT/  -  NUMBER OF DIAGONAL MATRICES TO MULTIPLY INPUT MATRIX.
C                   WILL BE STORED CONSEQUETIVELY ON TAPE11.

C        MULTP = 1  -  PRINT VECTOR, RESULTS, AND ROW SUMS.
C        MULTP = 2  -  PUT VECTOR, RESULTS, AND ROW SUMS ON TAPE11.
C        MULTP = 3  -  PRINT RESULTS AND PUT ON TAPE11.

      IF( MULT .GT. 0 ) GO TO 2
      TITLE(17) = 10HFINAL * DI
      TITLE(18) = 10HAGONAL MAT
      GO TO 4
 2    CONTINUE
      TITLE(17) = 10HTRANS FIN
      TITLE(18) = 10H* DIAG MAT
      CALL TRANS( LOC, LOCT, KSECT, N, II, A )
 4    CONTINUE
      IMULT = IABS( MULT )

C        GO THROUGH THIS LOOP ONCE FOR EACH PRODUCT DESIRED. WILL STORE
C        THE FIRST PRODUCT ON FILENAM AT LOC.

      DO69 K=1,IMULT
C     READ DIAGONAL INTO X                                      K  13
      READ 70, (X(I),I=1,N)                                     K  14
70    FORMAT (7F10.2)                                           K  51
      PRINT 75, K, (I, X(I), I=1,N)
 75   FORMAT( 1H1, 10X, *DIAGONAL MATRIX NO.*, I3, * IS*, //(20X, I10,
     1 F15.4) )
      IF( MULTP .EQ. 1 ) GO TO 6
      WRITE(11) TITLE
      WRITE(11) (X(I), I=1,N)
 6    CONTINUE
      ME=MINO(N,II7)
      MB=1                                                      K  16
```

```
C
C     CLEAR FOR ROW SUMS
C
      DO 10 I=1,N
10    SUM(I)=0.
      IF( MULTP .EQ. 2 ) GO TO 15
      I1 = 1
      I2 = 7
      IPP = 7
      IPAGE = 1
      NC = MINO(55,N)
      ND = 600
 15   CONTINUE
      II = 1
      LLOC = LOC
      IF( MULT .GT. 0 ) LLOC = LOCT
      KLOC = LLOC
      LL = 0
20    LD=ME-MB+1
      JGO = -1
      DO 30 J=1,LD
      CALL RANOIO( A(1,J), A(N,J), LLOC )
      LLOC = LLOC + KSECT
30    CONTINUE
C     DO A COL AT A TIME,KEEPING ROW SUMS
      JGO = 0
      DO 50 J=1,LD
      LL=LL+1
      DO 40 I=1,N
      HOLD(I,II) = A(I,J) * X(LL)
      SUM(I) = SUM(I) + HOLD(I,II)
40    CONTINUE
      CALL RANOIO( HOLD(1,II), HOLD(N,II), KLOC )
      IF( K .EQ. 1 .AND. LL .EQ. 1 ) LSTORE = KLOC
      GO TO (47, 45, 45), MULTP
 45   CONTINUE
      WRITE(11) (HOLD(I,II), I=1,N)
      IF( MULTP .EQ. 2 ) GO TO 48
 47   CONTINUE
      II = II + 1
      IF( II .LT. 8 ) GO TO 48
      CALL PRINT( N, IPP, NC, HOLD, ND, IPAGE, I1, I2, TITLE )
      I1 = I2 + 1
      I2 = I2 + 7
      I2 = MINO(N,I2)
      II = 1
   48 CONTINUE
50    CONTINUE
      IF (ME.EQ.N) GO TO 60
      ME = MINO( N,ME+II7)
      MB = MB + II7
      GO TO 20
 60   CONTINUE
      IF( II .EQ. 1 ) GO TO 65
      IPP = I2 - I1 + 1
      CALL PRINT( N, IPP, NC, HOLD, ND, IPAGE, I1, I2, TITLE )
 65   CONTINUE
      GO TO (66, 68, 66), MULTP
 66   CONTINUE
      PRINT 67, (I, SUM(I), I=1,N)
 67   FORMAT( 1H1, 10X, *THE ROW SUMS ARE-*, 2X, *ROW*, 8X, *SUM*, //
     1 (29X, I3, 2X, F15.4) )
      IF( MULTP .EQ. 1 ) GO TO 69
 68   CONTINUE
      WRITE(11) (SUM(I), I=1,N)
      ENDFILE 11
```

```
 69    CONTINUE
       LOC = LSTORE
       RETURN                                                          K  48
       END                                                             K  52-
*DECK  MULTV
       SUBROUTINE MULTV( MX, NTRANV, A, N, II, LOC, KSECT, KCHOICE )

C      MULTIPLIES COL VECTOR* MATRIX ON FILENAM AT LOC.
C      PRINTS ANSWER=VECTOR --NTRANV=0=REG MAT,=1=TRANSPOSE           L   7

       COMMON V1(600),V2(600),V3(600),TITLE(18),II7
       COMMON /3/ HOLD(600,7)
       COMMON /RNDMIO/ FILENAM, JGO
       DIMENSION A(N,II), ID(3,4), X(600), Y(600)
       EQUIVALENCE (X,V1), (Y,V2)                                      L   5

C         WILL NOW READ IN MIN(MX,4) VECTORS BY WHICH THE INPUT MATRIX
C         WILL BE MULTIPLIED.

       ICNT = 0
       IF( NTRANV .EQ. 1 ) GO TO 2
       TITLE(17) = 10HFINAL MAT.
       TITLE(18) = 10H * VECTOR
       GO TO 5
  2    CONTINUE
       TITLE(17) = 10HTRANSP FIN
       TITLE(18) = 10H MAT * VEC
  5    CONTINUE
       IEND = MIN0(MX,4)
  4    CONTINUE
       ICNT = ICNT + IEND
       DO1 I=1,IEND
       READ3, (ID(J,I), J=1,3)
  3    FORMAT( 3A10 )
       READ 70, (HOLD(J,I), J=1,N)
  1    CONTINUE

C      CLEAR SUMMING VECTOR (ANSWER)                                   L  17

       NN = IEND*N
       DO10 I=1,NN
       A(I) = 0.
  10   CONTINUE
       IF( KCHOICE .EQ. 5 ) GO TO 20
       JGO = -1
  20   CONTINUE
       LLOC = LOC
       DO34 J=1,N

C          READ IN A COLUMN OF THE MATRIX.
C                ON FILENAM IF KCHOICE .LT. 5
C                ON TAPE3 IF KCHOICE .EQ. 5

       IF( KCHOICE .LT. 5 ) GO TO 22
       READ(3) (X(I), I=1,N)
       GO TO 24
  22   CONTINUE
       CALL RANDIO( X, X(N), LLOC )
       LLOC = LLOC + KSECT
  24   CONTINUE
       IF( NTRANV .EQ. 1 ) GO TO 28
       DO26 I=1,IEND
       DO25 K=1,N
       A(K,I) = A(K,I) + HOLD(J,I)*X(K)
  25   CONTINUE
  26   CONTINUE
       GO TO 34
  28   CONTINUE
       DO32 I=1,IEND
       DO30 K=1,N
       A(J,I) = A(J,I) + HOLD(K,I)*X(K)
```

```
 30     CONTINUE
 32     CONTINUE
 34     CONTINUE
        DO38 I=1,IEND
        Y(I) = 0.
        DO36 J=1,N
        Y(I) = Y(I) + A(J,I)
 36     CONTINUE
 38     CONTINUE
        PRINT 40, (TITLE(J), J=1,18), ((ID(I,J), I=1,3), J=1,IEND)
 40     FORMAT( 1H1, 40X, 8A10, /40X, 8A10, /40X, 2A10, /5X, 4(2X,3A10) )
        PRINT 45
 45     FORMAT( 3X, 4(9X, *VECTOR*, 10X, *PRODUCT*) )
        DO60 J=1,N
        PRINT 50, (HOLD(J,I), A(J,I), I=1,IEND )
 50     FORMAT( 5X, 4(4X, 2F14.5) )
60      CONTINUE
        PRINT 65, (Y(I), I=1,IEND)
 65     FORMAT( //10X, *SUM*, 7X, 3(E15.7,15X), E15.7)
        IF( ICNT .EQ. MX ) RETURN
        IEND = MINO(MX, IEND+4) - IEND
        GO TO 4
C
70      FORMAT (7F10.2)
        END
*DECK   NORMIZ
        SUBROUTINE NORMIZ (NSORT, NP, A, N, II, LN, KSECT, LOC )

C       READ IN NORMALIZING VECTOR, AGGREGATE IT IF NP .GT. 0
C       NORMALIZE IN 2 PASSES IF N.GT.II

        COMMON V1(600),V2(600),V3(600),TITLE(18),II7
        COMMON /RNDMIO/ FILENAM, JGO
        DIMENSION A(LN,II)
        DIMENSION V(600), HOLD(600), NSORT(600)
        EQUIVALENCE (V1,V), (HOLD,V2)
        READ 100, (V(I),I=1,N)
        IF (NP.EQ.0) GO TO 30
C       MOVE V TO HOLD
        DO 10 I=1,N
        HOLD(I)=V(I)
10      V(I)=0.
        DO 20 J=1,N
        K=NSORT(J)
        V(K)=V(K)+HOLD(J)
20      CONTINUE
        K=NP
        GO TO 40
30      K=N
40      CONTINUE
        LL=0
C       FROM HERE ON K=ORDER
        MB=1
        ME=MINO(K,II)
C       FILL MATRIX WITH LD COL OF MATRIX ON FILENAM COLWISE
        LLOC = LOC
50      LD=ME-MB+1
        JGO = -1
        KLOC = LLOC
        DO 60 J=1,LD
        CALL RANDIO( A(1,J), A(K,J), LLOC )
        LLOC = LLOC + KSECT
60      CONTINUE
        DO 70 J=1,LD
        LL=LL+1
        DO 70 I=1,K
C       COMPUTE A NORMALIZED
        A(I,J)=A(I,J)/V(LL)
```

```
70     CONTINUE
       JGO = 1
       DO 80 J=1,LD
       CALL RANDIO( A(1,J), A(K,J), KLOC )
       KLOC = KLOC + KSECT
80     CONTINUE
       IF (ME.EQ.K) GO TO 90
       ME=MINO(K,ME+II)
       MB=MB+II
       GO TO 50
90     CONTINUE
       PRINT 110, (I,V(I),I=1,K)
       RETURN
100    FORMAT (7F10.2)
110    FORMAT (1H1,18HNORMALIZING VECTOR///(I10,F10.2))
       END
*DECK  PRINT
       SUBROUTINE PRINT( N, IPP, NC, A, ND, IPAGE, J1, J2, TITLE )
       DIMENSION A(ND,IPP), TITLF(18)
       DIMENSION FMT(4), FT(9)
       DATA FMT / 10H(2X,*ROW*,, 3HI5,,0, 6HF13.5) /
       DATA FT / 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9 /
       FMT(3) = FT(IPP)
       I1 = 1
       I2 = NC
       DO2 I=1,N,NC
       PRINT 1, TITLE, IPAGE, (J, J=J1,J2)
 1     FORMAT( 1H1, 10X, 8A10, /11X, 8A10,/11X, 2A10, 60X, *PAGE*, I3,
      1 //7X, *COLUMN*, I8, 8I13, /)
       IPAGE = IPAGE + 1
       PRINT FMT, (II, (A(II,J), J=1,IPP), II=I1,I2)
       I1 = I2 + 1
       I2 = MINO( I2+NC, N )
 2     CONTINUE
       RETURN
       END
*DECK  TINYIN
       SUBROUTINE TINYIN (IPOT,ISING,ND,A)
       COMMON /2/ NUMROS, NUMROW, ROW(510,2)
       DIMENSION IPIV(200), INDEX(200,2)
       DIMENSION A(ND,ND)
C      THIS SUBROUTINE INVERTS A MATRIX (BY GAUSS-JORDAN REDUCTION)
C      OF MAXIMUM SIZE 155 X 155 INCORE. (ONE NINTH OF 465 X 465)
       DETERM=1.0
       ISING=0
       DO 10 J=1,IPOT
10     IPIV(J)=0
       DO 140 I=1,IPOT
C
C      SEARCH FOR PIVOT ELEMENT
       AMAX=0.0
       DO 60 J=1,IPOT
       IF (IPIV(J)) 20,20,60
20     DO 50 K=1,IPOT
       IF (IPIV(K)-1) 30,50,190
30     IF (ABS(AMAX)-ABS(A(J,K))) 40,50,50
40     NROW=J
       ICOLUM=K
       AMAX=A(J,K)
50     CONTINUE
60     CONTINUE
       IPIV(ICOLUM)=IPIV(ICOLUM)+1
C
C      INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL.
       IF (NROW-ICOLUM) 70,90,70
70     DETERM=-DETERM
       DO 80 L=1,IPOT
       SWAP=A(NROW,L)
       A(NROW,L)=A(ICOLUM,L)
```

```
80    A(ICOLUM,L)=SWAP
90    INDEX(I,1)=NROW
      INDEX(I,2)=ICOLUM
      PIVOT=A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT
      A(ICOLUM,ICOLUM)=1.
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT.
      DO 100 L=1,IPDT
100   A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
C
C     REDUCE NON-PIVOT ROWS.
      DO 130 L1=1,IPDT
      IF (L1-ICOLUM) 110,130,110
110   AMAX=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.
      DO 120 L=1,IPDT
120   A(L1,L)=A(L1,L)-A(ICOLUM,L)*AMAX
130   CONTINUE
140   CONTINUE
C
C     INTERCHANGE COLUMNS
      DO 170 I=1,IPDT
      L=IPDT+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 150,170,150
150   NROW=INDEX(L,1)
      ICOLUM=INDEX(L,2)
      DO 160 K=1,IPDT
      SWAP=A(K,NROW)
      A(K,NROW)=A(K,ICOLUM)
      A(K,ICOLUM)=SWAP
160   CONTINUE
170   CONTINUE
C
C     TEST FOR SINGULAR MATRIX.
C
      DO 180 I=1,IPDT
      IF (IPIV(I)-1) 190,180,190
180   CONTINUE
      GO TO 200
190   ISING=1
200   CONTINUE
      RETURN
      END
*DECK  TRANS
      SUBROUTINE TRANS( LOCA, LOCT, KSECTA, N, NT, A )
      COMMON V1(600),V2(600),V3(600),TITLE(18),I17
      COMMON /RNDMIO/ FILENAM, JGO
      DIMENSION A(NT,N), LOC(20)

C        WILL TRANSPOSE A MATRIX OF RANK N ON FILENAM AT LOCA USING THE
C        ARRAY A. WILL PUT THE TRANSPOSE BACK ON FILENAM AT LOCT.

      NFILE = N/NT
      IF( NT*NFILE .LT. N ) NFILE = NFILE + 1

C        A CAN CONTAIN AT MOST NT=NDS/N COLUMNS AT ONE TIME, WILL THUS HAVE
C        TO SET UP N/NT+1 FILES. EACH FILE, EXCEPT PROBABLY THE LAST, WILL
C        CONTAIN NT COLUMNS.

      KLOC = LOCA
      DO2 I=1,N
      JGO = -1
      CALL RANDIO( A, A(N), KLOC )
      KLOC = KLOC + KSECTA
      I1 = 1
      I2 = NT
      JGO = 0
```

```
      DO1 J=1,NFILE
      CALL RANDIO( A(I1), A(I2), LLOC )
      IF( I .EQ. 1 ) LOC(J) = LLOC
      I1 = I2 + 1
      I2 = I2 + NT
    1 CONTINUE
    2 CONTINUE
C        MUST NOW PUT THE INVERSE BACK ON FILENAM AT LOCT.
      KSECT = NT/64 + 1
      NCONS = NFILE*KSECT
      IEND = NT
      KLOC = LOCA
      DO5 I=1,NFILE
      JGO = -1
      LLOC = LOC(I)
      DO3 J=1,N
      CALL RANDIO( A(1,J), A(NT,J), LLOC )
      LLOC = LLOC + NCONS
    3 CONTINUE
      JGO = 0
      IF( I .EQ. NFILE ) IEND = N - NT*(NFILE-1)
      DO4 J=1,IEND
      DO32 K=I,N
      V1(K) = A(J,K)
   32 CONTINUE
      CALL RANDIO( V1, V1(N), KLOC )
      IF( I .EQ. 1 .AND. J .EQ. 1 ) LOCT = KLOC
    4 CONTINUE
    5 CONTINUE
      RETURN                                                          C  45-
      END
*DECK   RANDIO
              IDENT RANDIO
              ENTRY RANDIO
              VFD   42/0LRANDIO,18/3
              USE   / /
 STRG         BSS   1
              USE   /RNDMIO/
 FILENAM      BSS   1
 JGO          BSS   1
              USE   0
 ALAST        DATA  0
 MASK         DATA  777777B
 CIOP         DATA  4LCIOP
 RCAL         DATA  3LRCL
 RDIO         DATA  12B
 WTIO         DATA  26B
 REIO         DATA  226B
 RNDM         DATA  40000000020000000B
 NFET         BSS   7
 RANDIO       BSSZ  1
              SB6   1
              SA1   FILENAM
              SA4   JGO
              NZ    X1,J6
              SX0   B6
              ZR    X4,L0
              SA2   B3
              NG    X4,L4
              EQ    B0,B0,L1
 L0           SA2   ALAST
              BX6   X2
              SA6   B3
```

```
L1      SA3   B1
        BX7   X3
        IX2   X2+X0
        SB1   X0+B1
        SA7   X2+STRG-1
        GE    B2,B1,L1
        NZ    X4,RANDIO
        BX6   X2
        SA6   A2
        EQ    B0,B0,RANDIO
        NO
        NO
L4      SA3   X2+STRG
        BX7   X3
        IX2   X2+X0
        SA7   B1
        SB1   X0+B1
        GE    B2,B1,L4
        EQ    B0,B0,RANDIO
        NO
        NO
J6      SX0   65
        SB7   B6+B6                                  E  67
        SB5   B6+B7
        SA3   WTIO+X4
        BX6   X1+X3
        SX5   B1
        SA6   NFET
        SA2   RNDM
        BX7   X2+X5
        SA7   A6+B6
        BX7   X5
        SA7   A7+B7
        SX3   B2
        SA2   MASK
        SA5   B3
        IX7   X3+X0
        BX6   X2*X5
        NG    X4,J1
        NZ    X4,J5
        SA6   A5
        SX6   A6
J5      SA6   A7+B5
        SX7   X7+B6
        SA7   A6-B7
        SX6   X3+B6
        SA6   A7-B7
        EQ    B0,B0,J2
        NO
J1      SA7   A7+B6
        SA6   A7+B7
        SX6   B1
        SA6   A7-B7
J2      SA1   1
        NZ    X1,J2
        SA4   CIOP
        SX5   NFET
        IX6   X5+X4
        NO
        SA6   A1
        NO
```

```
 J3         SA1   1
            NZ    X1,J3
            SA2   NFET
            NO
            LX2   59
            NG    X2,J4
            SA3   RCAL
            BX6   X3
            SA6   A1
            EQ    B0,B0,J3
 J4         SA1   JGO
            NG    X1,RANDIO
            SA1   B3
            SA2   MASK
            BX6   X1*X2
            SA6   A1
            EQ    B0,B0,RANDIO
            NO







            END
7
RUNF(S)
X                                                                     X
```

HERP Matrix Inversion Program

Program: HERP Matrix Inversion Routine

Author: Mary Carasso: Adapted from George S. Timson, Harvard Economics Research MAMMOTH Program Manual.

Language: Fortran IV, tested on CDC 6400, 6600 but easily adaptable to any computer. (Programmer note: should I-O become a problem with storage, the two subroutines GET and PUT can easily be modified.)

Method: The method used is described at the end of this writeup in detail. Basically, HERP partitions the matrix 3 x 3 to define 9 submatrices. The nine submatrices are manipulated as shown in the equations on page 116-117.

Restrictions: The matrix to be inverted must be square and nonsingular. The core allocation is dynamic and in theory the user's core is the limit. It should be noted that several vectors are set at 465 as the source deck currently stands. See USE section below. A consideration also is the lower limit of N. If N is $\leqslant$ 150, a standard inversion routine should be used as HERP uses considerable I-O and it would not be sensible if not needed.

Operation: The input matrix, the matrix to be inverted, must be rowwise, binary on user labeled TAPE19. The input tape is saved on TAPE19. The program requires 19 additional temporary tape or disk storage spaces (2 for each submatrix + one extra).

The program reads a control card, inverts the matrix, and leaves it on TAPE1. The control card has 2 items in 2(I4) format In column 1-4 is N the order of the matrix. In column 5-8 is ND. The program begins by dividing the order N/3. ND must be greater than or equal to N/3 (rounded up) + 5. For example, if N = 100, N/3 = 34. Therefore ND $\geqslant$ 39. The user must dimension A in the Fortran source deck (ND, ND), so obviously he (user) must use some care in determining ND so as to not go over his core size. (The program itself, not including matrix A, uses approximately 65000_8 Core).

Use: The program has been checked using a ROTHMAN Inverse check to determine accuracy. The inversions were accurate to 8 digits. There are several vectors restricted to 465 as the program stands but these could be changed. There is a row-summing subprogram which is not general in package; however, it could easily be generalized.

Timing: The program inverts a 462 order matrix in about 1/2 hour on a CDC 6600.

Output: The input and output matrices are printed. The print program prints 8 columns, 54 rows per page (a 460 order yields about a foot of 'ouptut). Error printout: If the value of ND on the control card does not coincide with or compare as less than the dimension of A in the main program, the statement NA, FIELD ERROR is printed, where NA is equal to $ND^2 + 1$.

HERP partitions the matrix 3 x 3 to define nine submatrices,

ABC

DEF

GHK

MAMMOTH then develops the nine submatrices that make up the inverse of the original matrix:

$$\begin{matrix} abc \\ def \\ ghi \end{matrix} \times \begin{matrix} ABC \\ DEF \\ GHK \end{matrix} = \begin{matrix} ABC \\ DEF \\ GHK \end{matrix} \times \begin{matrix} abc \\ def \\ ghi \end{matrix} = \begin{matrix} I00 \\ 0I0 \\ 00I \end{matrix}$$

The above gives the following equations:

$$Ac + Bf + Ci = 0$$
$$Ac = -Bf-Ci$$
$$c = -A^{-1}\ bf-A^{-1}\ Ci \qquad (1)$$

$$Dc + Ef + Fi = 0$$
$$Ef = -Dc-Fi = DA^{-1}\ (Bf + Ci)-Fi$$
$$(E-DA^{-1}\ B)f = (DA^{-1}\ C-F)i \qquad (2)$$

$$Gc + Hf + Ki = I$$
$$Ki = I-Gc-Hf = I + GA^{-1}\ (Bf + Ci)-Hf = I + GA^{-1}\ Ci + (GA^{-1}B=H)f$$
$$Ki = I + GA^{-1}\ Ci + (GA^{-1}\ B-H)\ (E-DA^{-1}\ B)^{-1}(DA^{-1}\ C-F)i$$
$$i = [K-GA^{-1}\ C - (GA^{-1}\ B-H)\ (E-DA^{-1}\ B)^{-1}\ (DA^{-1}\ C-F)]^{-1} \qquad (3)$$

N.B.: Subscripted capital letters indicate locations on disks, not necessarily the original submatrices that occupied those locations. Thus, a function formula, the 1 or 2, can be related to one of the 18 disk data sets.

$$gA + hD + iG = 0$$
$$g + -hDA^{-1} -iGA^{-1} \tag{4}$$

$$gB + hE + iH = 0$$
$$hE = -iH-gB = -iH + (hD + iG)A^{-1}B$$
$$h(E-DA^{-1}B) = -iH + iGA^{-1}B$$
$$h = i(GA^{-1} B-H)(E-DA^{-1}B)^{-1} \tag{5}$$

$$Ab + Be + Ch = 0$$
$$b = -A^{-1} Be-A^{-1} Ch \tag{6}$$

$$dB + eE + fH = I$$
$$eE = I-dB-fH = I + eDA^{-1} B + fGA^{-1} B-fH$$
$$e(E-DA^{-1} B) = I + f(GA^{-1} B-H)$$
$$e = (E-DA^{-1} B)^{-1} - f(H-GA^{-1} B)(E-DA^{-1} B)^{-1} \tag{7}$$

$$aA + bD + cG = I$$
$$a = A^{-1} -bDA^{-1} -cGA^{-1} \tag{8}$$

$$dA + eD + fG = 0$$
$$dA = -eD -fG$$
$$d = -eDA^{-1} -fGA^{-1} \tag{9}$$

The MAMMOTH INVERT subroutine solves the nine numbered equations in the following order: (3) (2) (5) (7) (1) (6) (9) (4) (8). (Note that Equation (3) expresses one of the inverse submatrices – i – in terms of all the submatrices of the original matrix. Once i is found, f can be found in terms of i, and so on.) To solve the equations, 24 multiplications of submatrices and three invertings of submatrices must be made. The 24 steps are shown below, the invertings taking place before steps 1, 9, and 11. In 8 of the 24 steps, the product resulting from the matrix multiplication is made negative. In 12 of the steps, the matrix product is added to a third matrix that already exists on disk. For each of the 24 matrix multiplications (X · Y) the "Y" matrix is put into core, the X matrix is read from whatever disk it is on, one row at a time, that row is multiplied by the whole Y matrix, and the resultant row of the product XY matrix is put onto disk before the next X-matrix row is read.

1.		$-G_2$	$\cdot A_2^{-1}$	$\rightarrow G_1 \rightarrow G_2 A^{-1} \rightarrow A_1, A_2$
2.		$-D_2$	$\cdot A_2^{-1}$	$\rightarrow D_1$
3.		$-A_2^{-1}$	$\cdot C_2$	$\rightarrow C_1$
4.	K_2	$-GA_1^{-1}$	$\cdot C_2$	$\rightarrow K_2$

5.	$F_2 - DA_1^{-1} \cdot C_2$	$\to F_2$
6.	$-A_2^{-1} \cdot B_2$	$\to B_1$
7.	$E_2 - DA_1^{-1} \cdot B_2$	$\to E_2$
8.	$H_2 - GA_1^{-1} \cdot B_2$	$\to H_2$
9.	$[H-GA^{-1}B]_2 \cdot [E-DA^{-1}B]_2^{-1}$	$\to H_1\ [E-DA^{-1}B]^{-1} \to E_2$
10.	$[K-GA^{-1}C]_2 - [(H-GA^{-1}B)(E-DA^{-1}B)^{-1}]_1 \cdot [F-DA^{-1}C]_2$	$\to K_2$
11.	$-[F-DA^{-1}C]_2 \cdot [i_2^{-1}]^{-1}$	$\to F_1 \qquad i \to K_1, K_2$
12.	$[E-DA^{-1}B]_2^{-1} \cdot [-(F-DA^{-1}C)i]_1$	$\to F_1\ (f) \to F_2$
13.	$-i_1 \cdot [(H-GA^{-1}B)(E-DA^{-1}B)^{-1}]$	$\to H_2\ (h)$
14.	$[E-DA^{-1}B]_2^{-1} - f_1 \cdot [(H-GA^{-1}B)(E-DA^{-1}B)^{-1}]_1$	$\to E_2\ (e)$
15.	$-[A^{-1}B]_1 \cdot f_2$	$\to C_2$
16.	$[-A^{-1}Bf]_2 - [A^{-1}C]_1 \cdot i_2$	$\to C_2\ (c)$
17.	$[-A^{-1}B]_1 \cdot e_2$	$\to B_2$
18.	$[-A^{-1}Be]_2 - [A^{-1}C]_1 \cdot h_2$	$\to B_2\ (b)$
19.	$-e_2 \cdot [DA^{-1}]_1$	$\to D_1$
20.	$-h_2 \cdot [DA^{-1}]_1$	$\to G_1$
21.	$A_1^{-1} - b_2 \cdot [DA^{-1}]_1$	$\to A_1$
22.	$[-eDA^{-1}]_1 - f_2 \cdot [GA^{-1}]_2$	$\to D_1\ (d) \to D_2$
23.	$[-hDA^{-1}]_1 - i_2 \cdot [GA^{-1}]_2$	$\to G_1\ (g) \to G_2$
24.	$[A^{-1} - bDA^{-1}] - C_2 \cdot [GA^{-1}]_2$	$\to A_1\ (a) \to A_2$

N.B.: Subscripted capital letters indicate locations on disks, not necessarily the original submatrices that occupied those locations. Through a function formula, the 1 or 2 can be related to one of the 18 disk data sets.

```
HERP,5,100,100000.800279,CARASSO COMPILE ONLY
RUN(S)
7
      PROGRAM HERPINV(INPUT,OUTPUT,TAPE1=1001,TAPE2=1001,TAPE3=1001,TAPE
     14=1001,TAPE5=1001,TAPE6=1001,TAPE7=1001,TAPE8=1001,TAPE9=1001,TAPE
     210=1001,TAPE11=1001,TAPE12=1001,TAPE13=1001,TAPE14=1001,TAPE15=100
     31,TAPE16=1001,TAPE17=1001,TAPE18=1001,TAPE19,TAPE20=1001)
      COMMON NUMROS,NUMROW,ROW(465,2)
      OIMENSION A(90,90)
      OIMENSION AC(2)
      OIMENSION INT(3), IWHERE(3), KON(3,24), M(2,24), MINUSQ(24), AOOQ(
     124)
      OIMENSION SET1(465), SET2(465), SET3(465), HOLO(465)
      DIMENSION INOKON(72), MC(2)
      EQUIVALENCE (KON(1,1),INOKON(1))
      EQUIVALENCE (A,SET1)
      EQUIVALENCE (A(500),SET2)
      EQUIVALENCE (A(1000),SET3)
      EQUIVALENCE (A(1500),HOLO)
      LOGICAL MINUSQ,AOOQ,REPEAT
C
C     THIS SUBROUTINE INVERTS THE MATRIX ORIGINALLY ON OISK 1 ACCOROING
C     THE ALGORITHM GIVEN IN THE MAMMOTH MANUAL, APPENOIX 3.
C     HERE 'A' IS SUB-MATRIX 1,1. 'B' IS SUB-MATRIX 1,2. ANO SO ON.
C     THE 'KON' ARRAY SHOWS THE ORDER OF THE 24 SUB-MATRIX MULTIPLICATIO
C     FOR EXAMPLE, IN THE FIRST MULTIPLICATION, 'C'--3,1--IS MULTIPLIED
C     A--1,1.
C     'M' TELLS WHICH OISK EACH OF THE TWO MATRICES IS ON. THE PROOUCT S
C     MATRIX MUST GO ONTO THE OISK THAT THE FIRST OF THE 2 SUB-MATRICES
C     NOT ON. IF THE PRODUCT IS MAOE NEGATIVE, 'MINUSQ' FOR THAT MULTIPL
C     CATION IS 'T'. IF THE PRODUCT IS ADDED TO ANOTHER MATRIX, ADDQ IS
      OATA KON/3,1,1,2,1,1,1,1,3,3,1,3,2,1,3,1,1,2,2,1,2,3,1,2,3,2,2,3,2
     1,3,2,3,3,2,2,3,3,3,2,2,3,2,1,2,3,1,3,3,1,2,2,1,3,2,2,2,1,3,2,1,1,2
     2,1,2,3,1,3,3,1,1,3,1/
      OATA M/2,2,2,2,2,2,1,2,1,2,2,2,1,2,1,2,2,2,1,2,2,2,2,1,1,1,1,1,1,2
     1,1,2,1,2,1,2,2,2,1,2,1,2,1,2,2,2,2,2,2/
      OATA MINUSQ/.TRUE.,.TRUE.,.TRUE.,.FALSE.,.FALSE.,.TRUE.,.FALSE.,.F
     1ALSE.,.FALSE.,.TRUE.,.TRUE.,.FALSE.,.TRUE.,.TRUE.,.FALSE.,.FALSE.,
     2.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./
      OATA AOOQ/.FALSE.,.FALSE.,.FALSE.,.TRUE.,.TRUE.,.FALSE.,.TRUE.,.TR
     1UE.,.FALSE.,.TRUE.,.FALSE.,.FALSE.,.FALSE.,.TRUE.,.FALSE.,.TRUE.,.
     2FALSE.,.TRUE.,.FALSE.,.FALSE.,.TRUE.,.TRUE.,.TRUE.,.TRUE./
      LIT(IJ,IK)=(((IJ-1)/NUMROS)+1)+((IK-1)*9)
      LOC(KA,KB)=((KA-1)*3+(KB-1))*NUMROS+1
      ISING=0
      REAO 460, N,NO
      MC(1)=N
      IF (N.GT.150) GO TO 10
      PRINT 470
10    NO3=NO*3
      IF (N.LE.NO3) GO TO 20
      PRINT 480, NO,N
      CALL EXIT
C     HAVE READ IN   N/3 + ROUNOUP AS NO
20    NOS=NO*NO
      NOO=NOS/N+1
      IP=(NOO/8)*8
      AC=9876.
      NA=(NO*NO)+1
      IF (A(NA).EQ.9876.) GO TO 30
      PRINT 490, NA
30    PRINT 500, N,NO,NO3,NOS,NOD,IP,NA
      REWINO 19
      REWIND 1
      DO 40 I=1,N
      READ (19) (ROW(J,1),J=1,N)
```

```
      WRITE (1) (ROW(J,1),J=1,N)
40    CONTINUE
      REWIND 1
      REWIND 19
      CALL PRINT (N,N,1,20HINPUT-TRANSPOSED PRT,IP,A)
      CALL SECOND (TIMEE)
      PRINT 510, TIMEE
      INT(1)=(MC(1)-1)/3+1
      INT(3)=(MC(1)-INT(1))/2
      INT(2)=MC(1)-INT(1)-INT(3)
      NMPERO=MIN0(465/INT(1),INT(1))
      NUMROS=(INT(1)-1)/NMPERO+1
C     NUMROW=LENGTH OF SEGMENT IN ROW
      NMAX=MAX0(INT(1),INT(2),INT(3))
      NUMROW=NMAX*NMPERO
      PRINT 520, N,INT(1),INT(2),INT(3),NMPERO,NUMROS,NUMROW,NMAX
      SUM=INT(1)
      ISQRT=SQRT(SUM)
C
C     PUT MATRIX ONTO DISK 2, BY NINTHS
      IBEG=0
      DO 110 KC=1,3
      REWIND 1
      IGROW=1
      INTKC=INT(KC)
      DO 100 KA=1,3
      IAROW=1
      IPROW=LOC(KA,KC)
      IT=LIT(IPROW,2)
      REWIND IT
50    JBEG=0
      DO 70 I=1,NMPERO
      READ (1) (ROW(L,1),L=1,N)
      IGROW=IGROW+1
      DO 60 J=1,INTKC
      IX=J+IBEG
      JX=J+JBEG
60    ROW(JX,2)=ROW(IX,1)
      IAROW=IAROW+1
      IF (IAROW-INT(KA)) 70,70,80
70    JBEG=JBEG+INTKC
80    DO 90 J=1,NUMROW
90    ROW(J,1)=ROW(J,2)
      CALL PUTROW (IPROW,2,IT)
      IPROW=IPROW+1
      IF (IAROW-INT(KA)) 50,50,100
100   CONTINUE
110   IBEG=IBEG+INT(KC)
      REPEAT=.FALSE.
120   CONTINUE
C
C
C     DO 24 MULTIPLICATIONS, ACCORDING TO 'KON','M','MINUSQ',AND 'ADDQ'
      DO 380 KOUNT=1,24
      KA=KON(1,KOUNT)
      KB=KON(2,KOUNT)
      KC=KON(3,KOUNT)
      INTKA=INT(KA)
      INTKB=INT(KB)
      INTKC=INT(KC)
      IWHERE(1)=M(1,KOUNT)
      IWHERE(2)=M(2,KOUNT)
      IWHERE(3)=3-IWHERE(1)
      ASSIGN 250 TO ISIGN
      IF (MINUSQ(KOUNT)) ASSIGN 240 TO ISIGN
      ASSIGN 290 TO IADDQ
```

```
      IF (AODQ(KOUNT)) ASSIGN 310 TO IAODQ
C
C     PUT MULTIPLIER MATRIX (KB BY KC) INTO CORE
      IGROW=LOC(KB,KC)
      IT=LIT(IGROW,IWHERE(2))
      REWIND IT
      IF (KOUNT.EQ.1) GO TO 130
      IF ((KB.EQ.KON(2,KOUNT-1)).AND.(KC.EQ.KON(3,KOUNT-1))) GO TO 220
130   DO 140 IAROW=1,INTKB,NMPERO
      CALL GETROW (IGROW,IWHERE(2),IT)
      IGROW=IGROW+1
      K=0
      IARWPL=IAROW+NMPERO-1
      DO 140 J=IAROW,IARWPL
      DO 140 I=1,INTKC
      K=K+1
140   A(J,I)=ROW(K,1)
C
C     INVERT MATRIX IN CORE, IF NECESSARY, AND PUT INVERSE BACK ON DISK
      IF ((KOUNT-1)*(KOUNT-9)*(KOUNT-11)) 220,150,220
150   CALL TINYIN (INTKC,ISING,ND,A)
      IF (ISING.EQ.0) GO TO 170
      IF ((KOUNT.NE.1).OR.(REPEAT)) GO TO 450
      REPEAT=.TRUE.
      DO 160 I=1,72
160   INDKON(I)=4-INDKON(I)
      GO TO 120
170   ASSIGN 190 TO IBOTHQ
      IPROW=LOC(KB,KC)
      IT=LIT(IPROW,1)
      LT=LIT(IPROW,2)
      REWIND LT
      REWIND IT
      IF (KOUNT.EQ.9) ASSIGN 200 TO IBOTHQ
      DO 210 IAROW=1,INTKB,NMPERO
      K=0
      IARWPL=IAROW+NMPERO-1
      DO 180 J=IAROW,IARWPL
      DO 180 I=1,INTKC
      K=K+1
180   ROW(K,1)=A(J,I)
      GO TO IBOTHQ, (190,200)
190   CALL PUTROW (IPROW,1,IT)
200   CALL PUTROW (IPROW,2,LT)
210   IPROW=IPROW+1
C
C     MULTIPLY THRU BY THE MULTIPLICAND (KA BY KB) AND PUT INTO PRODUCT
220   IPROW=LOC(KA,KC)
      IGROW=LOC(KA,KB)
      REWIND 20
      IT=LIT(IPROW,IWHERE(3))
      LT=LIT(IGROW,IWHERE(1))
      REWIND IT
      REWIND LT
      DO 340 IAROW=1,INTKA,NMPERO
      CALL GETROW (IGROW,IWHERE(1),LT)
      IGROW=IGROW+1
      JBEG=0
      IBEG=0
      DO 280 I=1,NMPERO
      DO 270 L=1,INTKC
      LX=L+IBEG
      ROW(LX,2)=0.0
      DO 260 JJJ=1,ISQRT
      SUM=0.0
      DO 230 J=JJJ,INTKB,ISQRT
      JX=J+JBEG
```

```
230   SUM=SUM+ROW(JX,1)*A(J,L)
      GO TO ISIGN, (240,250)
240   ROW(LX,2)=ROW(LX,2)-SUM
      GO TO 260
250   ROW(LX,2)=ROW(LX,2)+SUM
260   CONTINUE
270   CONTINUE
      JBEG=JBEG+INTKB
280   IBEG=IBEG+INTKC
      GO TO IADDQ, (290,310)
290   DO 300 L=1,NUMROW
300   ROW(L,1)=ROW(L,2)
      GO TO 330
310   CALL GETROW (IPROW,IWHERE(3),IT)
      DO 320 L=1,NUMROW
320   ROW(L,1)=ROW(L,1)+ROW(L,2)
330   CALL PUTROW (IPROW,IWHERE(3),20)
340   IPROW=IPROW+1
      REWIND 20
      REWIND IT
      DO 350 I=1,NUMROS
      READ (20) (ROW(L,1),L=1,NUMROW)
      WRITE (IT) (ROW(L,1),L=1,NUMROW)
350   CONTINUE
      REWIND IT
      REWIND 20
C
C     IN 5 CASES, PRODUCT MUST GO ON DISK 2, AS WELL AS DISK 1
      IF ((KOUNT-1)*(KOUNT-12)*(KOUNT/22-1)) 380,360,380
360   IGROW=LOC(KA,KC)
      IPROW=3-IWHERE(3)
      IT=LIT(IGROW,IPROW)
      LT=LIT(IGROW,IWHERE(3))
      REWIND IT
      REWIND LT
      DO 370 I=1,NUMROS
      CALL GETROW (IGROW,IWHERE(3),LT)
      CALL PUTROW (IGROW,IPROW,IT)
370   IGROW=IGROW+1
380   CONTINUE
C
C     AFTER 24 MULTIPLICATIONS, PUT MATRIX BACK ON DISK 1 THE WAY IT WAS
      IT=1
      REWIND 1
      I1=INT(1)
      I2=I1+INT(2)
      I3=I2+INT(3)
      I1P=I1+1
      I2P=I2+1
      DO 440 KC=1,3
      IGROW1=LOC(KC,1)
      IGROW2=LOC(KC,2)
      IGROW3=LOC(KC,3)
      IT1=LIT(IGROW1,2)
      IT2=LIT(IGROW2,2)
      IT3=LIT(IGROW3,2)
      REWIND IT1
      REWIND IT2
      REWIND IT3
      LOOP=INT(KC)
      ICT=0
      DO 430 II=1,NUMROS
      READ (IT1) (SET1(L),L=1,NUMROW)
      READ (IT2) (SET2(L),L=1,NUMROW)
      READ (IT3) (SET3(L),L=1,NUMROW)
      J1=0
```

```
      J2=0                                                              A 263
      J3=0                                                              A 264
      DO 420 K=1,NMPERO                                                 A 266
      IF (ICT.GE.LOOP) GO TO 440                                        A 267
      ICT=ICT+1                                                         A 268
      DO 390 J=1,I1                                                     A 269
      J1=J1+1                                                           A 270
390   HOLD(J)=SET1(J1)                                                  A 271
      DO 400 J=I1P,I2                                                   A 272
      J2=J2+1                                                           A 273
400   HOLD(J)=SET2(J2)                                                  A 274
      DO 410 J=I2P,I3                                                   A 275
      J3=J3+1                                                           A 276
410   HOLD(J)=SET3(J3)                                                  A 277
      WRITE (1) (HOLD(L),L=1,N)                                         A 278
420   CONTINUE                                                          A 280
430   CONTINUE                                                          A 281
440   CONTINUE                                                          A 282
      REWIND 1                                                          A 283
      CALL SECOND (TIMEE)                                               A 284
      PRINT 510, TIMEE                                                  A 285
      CALL SUMSUM(1,N)
C     THIS ROUTINE NEEDS THE BIG FIX BUT FOR THE MOMENT
C     DIMENSIN DOES NOT FIT WITH CODE
      CALL PRINT (N,N,1,20H I-A INVERSE MATRIX ,IP,A)                   A 286
      CALL EXIT                                                         A 288
450   PRINT 550                                                         A 289
      CALL EXIT                                                         A 290
C                                                                       A 291
460   FORMAT (2I4)                                                      A 292
470   FORMAT (*1THEORDEROFYOURMATRIXISLT150USEOTHERINVERSIONROUTINE....I A 293
     1NRUTH2*)                                                          A 294
480   FORMAT (*1ND(*,I4,*)TIMES3MUSTBEGEN(*,I4,*)ITISNOT,EXIT*)         A 295
490   FORMAT (I6,*FIELDERROR*)                                          A 296
500   FORMAT (*1CONTROLCARD+CALCVALUES*,7I6)                            A 297
510   FORMAT (*1TIME=*,E15.4)                                           A 298
520   FORMAT (10I6)                                                     A 299
530   FORMAT (1H1)                                                      A 300
540   FORMAT (**,8F15.7)                                                A 301
550   FORMAT (*SINGULAREND*)                                            A 302
      END                                                               A 303
      SUBROUTINE PUTROW (I,IT,INEW)                                     B   1
      COMMON NUMROS,NUMROW,ROW(465,2)                                   B   2
      WRITE (INEW) (ROW(J,1),J=1,NUMROW)                                B   3
      RETURN                                                            B   4
C                                                                       B   5
10    FORMAT (*P*,5I6)                                                  B   6
      END                                                               B   7
      SUBROUTINE SUMSUM(NTAP,NR)
      COMMON NUMROS,NUMROW,ROW(465,2)                                   C   2
      DIMENSION A(231,21)
      DIMENSION SAVE(231,12)
      REWIND NTAP                                                       C   5
      DO 10 K=1,12
      DO 10 I=1,NR                                                      C   7
      SAVE(I,K)=0.                                                      C   8
10    CONTINUE                                                          C   9
      DO 60 K=1,11
      DO 20 J=1,21
20    READ (NTAP) (A(I,J),I=1,NR)                                       C  12
      DO 40 J=1,21
      DO 30 I=1,NR                                                      C  14
      SAVE(I,K)=SAVE(I,K)+A(I,J)                                        C  15
30    CONTINUE                                                          C  16
40    CONTINUE                                                          C  17
```

```
      DO 50 I=1,NR
      SAVE(I,12)=SAVE(I,12)+SAVE(I,K)
50    CONTINUE
60    CONTINUE
      PRINT 110
      PRINT 121,(I,I=1,8)
      JA =1
      JB=8
62    CONTINUE
      DO 70 I=1,NR
      PRINT 150,I,(SAVE(I,K),K=JA,JB)
70    CONTINUE
      IF(JA.GT.1)GO TO 72
      JA=JA+8
      JB=12
      PRINT 110
      PRINT 120,(I,I=9,12)
      GO TO 62
72    CONTINUE
      DO 90 K=1,11
      DO 80 I=1,NR
      SAVE(I,K)=SAVE(I,K)/SAVE(I,12)
80    CONTINUE
90    CONTINUE
      PRINT 130
      PRINT 121,(I,I=1,8)
      JA =1
      JB=8
92    CONTINUE
      DO 100 I=1,NR
      PRINT 150,I,(SAVE(I,K),K=JA,JB)
100   CONTINUE
      IF(JA.GT.1) GO TO 102
      JA=JA+8
      JB=11
      PRINT 130
      PRINT 140,(I,I=9,11)
      GO TO 92
102    CONTINUE
      REWIND NTAP
      RETURN
C
110   FORMAT(*1 SUM DF 21 COLUMNS AT A TIME + TOTAL COLUMN*)
120   FORMAT(10X,4(I2,13X),*TOTAL*)
121   FORMAT(10X,8(I1,14X))
140   FORMAT(10X,3(I2,13X))
130   FORMAT(*1 PERCENTAGE DF EACH CDLUMN DF THE TDTAL SUM COLUMN*)
150    FORMAT(* *,I4,8F15.7)
      END
      SUBROUTINE GETROW (I,IT,INEW)
      COMMON NUMROS,NUMROW,ROW(465,2)
      READ (INEW) (ROW(J,1),J=1,NUMROW)
      RETURN
C
10    FORMAT (*G*,5I6)
      END
      SUBROUTINE TINYIN (IPDT,ISING,ND,A)
      DIMENSION IPIV(200), INDEX(200,2)
      COMMON NUMRDS,NUMRDW,RDW(465,2)
      OIMENSION A(ND,ND)
C     THIS SUBROUTINE INVERTS A MATRIX (BY GAUSS-JORDAN REDUCTIDN)
C     DF MAXIMUM SIZE 155 X 155 INCORE. (ONE NINTH OF 465 X 465)
      DETERM=1.0
      DO 10 J=1,IPDT
10    IPIV(J)=0
      DO 140 I=1,IPDT
```

```
C
C     SEARCH FOR PIVOT ELEMENT
      AMAX=0.0
      DO 60 J=1,IPDT
      IF (IPIV(J)) 20,20,60
20    DO 50 K=1,IPDT
      IF (IPIV(K)-1) 30,50,190
30    IF (ABS(AMAX)-ABS(A(J,K))) 40,50,50
40    NROW=J
      ICOLUM=K
      AMAX=A(J,K)
50    CONTINUE
60    CONTINUE
      IPIV(ICOLUM)=IPIV(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL.
      IF (NROW-ICOLUM) 70,90,70
70    DETERM=-DETERM
      DO 80 L=1,IPDT
      SWAP=A(NRDW,L)
      A(NROW,L)=A(ICOLUM,L)
80    A(ICOLUM,L)=SWAP
90    INDEX(I,1)=NRDW
      INDEX(I,2)=ICOLUM
      PIVOT=A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT
      A(ICOLUM,ICOLUM)=1.
C
C     DIVIDE PIVOT ROW BY PIVDT ELEMENT.
      DO 100 L=1,IPDT
100   A(ICOLUM,L)=A(ICDLUM,L)/PIVOT
C
C     REDUCE NDN-PIVDT RDWS.
      DO 130 L1=1,IPDT
      IF (L1-ICDLUM) 110,130,110
110   AMAX=A(L1,ICDLUM)
      A(L1,ICOLUM)=0.
      DO 120 L=1,IPDT
120   A(L1,L)=A(L1,L)-A(ICOLUM,L)*AMAX
130   CONTINUE
140   CDNTINUE
C
C     INTERCHANGE CDLUMNS
      DD 170 I=1,IPDT
      L=IPDT+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 150,170,150
150   NRDW=INDEX(L,1)
      ICDLUM=INDEX(L,2)
      DD 160 K=1,IPDT
      SWAP=A(K,NRDW)
      A(K,NRDW)=A(K,ICDLUM)
      A(K,ICDLUM)=SWAP
160   CDNTINUE
170   CDNTINUE
C
C     TEST FDR SINGULAR MATRIX.
C
      DD 180 I=1,IPDT
      IF (IPIV(I)-1) 190,180,190
180   CDNTINUE
      GD TD 200
190   ISING=1
200   CDNTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINT (NR,NC,NTAP,TITLE,IP,A)
C     PRINTS(NR,NC) MATRIX ON TAPE NTAP HEADED BY TITLE
C     WITH 52 LINES (RDWS/PAGE)  IN IP BATCHES
      COMMON NUMROS,NUMROW,ROW(465,2)
      DIMENSION A(NR,IP)
      DIMENSION TITLE(2)
      ISEE=2
      REWIND NTAP
      IPAGE=1
      LC=0
      KC=0
      JB=0
      MB=1
      ME=MINO(IP,NC)
10    CONTINUE
      MC=ME-MB+1
C     MC=IS COL BATCH AMT
      KC=LC
      LC=LC+MC
      DO 20 J=1,MC
      READ (NTAP) (A(I,J),I=1,NR)
20    CONTINUE
C     PRINT MC COL, NR ROWS
30    JA=JB+1
      IA=1
      IB=0
      IF (JB.GE.LC) GO TO 120
      JC=JA+7
      IF (NC.LE.JC) GO TO 40
      JB=JC
      GO TO 50
40    JB=NC
50    IF (IB.GE.NR) GO TD 30
      IC=IA+52
      IF (NR.LE.IC) GO TO 60
      IB=IC
      GO TO 70
60    IB=NR
70    PRINT 150, (TITLE(I),I=1,2),IPAGE
      JD=JA-KC
      JE=JB-KC
      PRINT 160, (J,J=JA,JB)
      IF (ISEE.NE.2) GO TO 90
      DO 80 I=IA,IB
80    PRINT 170, I,(A(I,J),J=JD,JE)
      GO TO 110
C     PRINT E FORMAT FOR DEBUG
90    DO 100 I=IA,IB
100   PRINT 140, I,(A(I,J),J=JD,JE)
110   IA=IB+1
      IPAGE=IPAGE+1
      GO TO 50
120   IF (ME.EQ.NC) GO TO 130
      MB=MB+IP
      ME=MINO(ME+IP,NC)
      GO TO 10
130   REWIND NTAP
      RETURN
C
140   FORMAT (*ROW*,I4,**,8(E15.3))
150   FORMAT (*1*,2A10,*PAGE*,I3/)
160   FORMAT (*COLUMN*,8(I14,**))
170   FORMAT (*ROW*,I4,**,8(F15.7))
      END
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) U. S. Army Engineer Institute for Water Resources	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE

A Multisectoral Model of Pacific and Mountain Interstate Trade Flows

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (First name, middle initial, last name)

H. Craig Davis and Everard M. Lofting

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1972	125	1 6-part table

8a. CONTRACT OR GRANT NO. DACW07-69-C-0013	9a. ORIGINATOR'S REPORT NUMBER(S) IWR Report 73-2
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	None

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	US Army Engineer Division, South Pacific; Institute for Water Resources, US Army Corps of Engineers

13. ABSTRACT

The report presents a multisectoral model of Pacific and mountain interstate trade flows for 1963. The basic concept of input-output analysis and the general methodology for the report are discussed in Part A, and the eleven Western states inter-industry models are presented in Part B of the report. The 1963 interregional transaction for the eleven states is shown in a 6-part oversize table and is inclosed in the back pocket of the report.

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Input-output Interregional and Interindustry Analysis						

UNCLASSIFIED
Security Classification

☆ U. S. GOVERNMENT PRINTING OFFICE 1973 O - 494-915

TABLE B-5.

ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963[a]

[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

	WYOMING	Sector Sums by State	Percent by State
ARIZONA	[illegible]	[illegible]	[illegible]
CALIFORNIA	[illegible]	[illegible]	[illegible]
COLORADO	[illegible]	[illegible]	[illegible]
NEW MEXICO	[illegible]	[illegible]	[illegible]
OREGON	[illegible]	[illegible]	[illegible]
UTAH	[illegible]	[illegible]	[illegible]
WASHINGTON	[illegible]	[illegible]	[illegible]
NEVADA	[illegible]	[illegible]	[illegible]
IDAHO	[illegible]	[illegible]	[illegible]
MONTANA	[illegible]	[illegible]	[illegible]
WYOMING	[illegible]	[illegible]	[illegible]

TABLE B-5 (part 6 of 6 parts), TABLE B-6, and TABLE B-7.
ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963
[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

[a]Each entry shows, per dollar of deliveries to final demand by industry and region at left, total dollar production directly and indirectly required from industry and region at top.

TABLE B-5.
ELEVEN WESTERN STATES INTERREGIONAL
TRANSACTIONS, 1963[a]

[Direct and Indirect (Total) Requirements per
Dollar of Final Demand]

	IDAHO	MONTANA
ARIZONA	[illegible]	[illegible]
CALIFORNIA	[illegible]	[illegible]
COLORADO	[illegible]	[illegible]
NEW MEXICO	[illegible]	[illegible]
OREGON	[illegible]	[illegible]
UTAH	[illegible]	[illegible]
WASHINGTON	[illegible]	[illegible]
NEVADA	[illegible]	[illegible]
IDAHO	[illegible]	[illegible]
MONTANA	[illegible]	[illegible]
WYOMING	[illegible]	[illegible]

TABLE B-5. (part 5 of 6 parts)
ELEVEN WESTERN STATES INTERREGIONAL
TRANSACTIONS, 1963
[Direct and Indirect (Total) Requirements per
Dollar of Final Demand]

[a]Each entry shows per dollar of deliveries to final demand by industry and region at left, total dollar production directly and indirectly required from industry and region at top.

TABLE B-5.
ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963[a]
[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

WASHINGTON | NEVADA

ARIZONA

CALIFORNIA

COLORADO

NEW MEXICO

OREGON

UTAH

WASHINGTON

NEVADA

IDAHO

MONTANA

WYOMING

TABLE B-5. (part 4 of 6 parts)
ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963
[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

[a]Each entry shows, per dollar of deliveries to final demand by industry and region at left, total dollar production directly and indirectly required from industry and region at top.

TABLE B-5.

ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS 1963[a]

[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

OREGON | UTAH

TABLE B-5. (part 3 of 6 parts)
ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963
[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

[a]Each entry shows, per dollar of deliveries to final demand by industry and region at left, total dollar production directly and indirectly required from industry and region at top.

TABLE B-5.
ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963*
[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

COLORADO | NEW MEXICO

ARIZONA

CALIFORNIA

COLORADO

NEW MEXICO

OREGON

UTAH

WASHINGTON

NEVADA

IDAHO

MONTANA

WYOMING

[illegible]

TABLE B-5. (part 2 of 6 parts)
ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963
[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

*Each entry shown per dollar of deliveries to final demand by industry and region at left, total dollar production directly and indirectly required from industry and region at top.

TABLE B-5. TABLE B-6. and TABLE B-7.

ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963[a]

[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

ARIZONA

CALIFORNIA

[illegible]

TABLE B-5. (part 1 of 6 parts)

ELEVEN WESTERN STATES INTERREGIONAL TRANSACTIONS, 1963

[Direct and Indirect (Total) Requirements per Dollar of Final Demand]

[a] Each entry shows, per dollar of deliveries to final demand by industry and region at left, total dollar production directly and indirectly required from industry and region at top.