The person charging this material is responsible for its return to the library from which it was withdrawn on or before the Latest Date stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

ENGINEERING

CONFERENCE ROOM

AUG 5 1976

AUG 16 1977

AUG 23 1980

NOV 16 1984

L161—O-1096
This paper presents the preliminary documentation and user's guide for the Center for Advanced Computation economic and manpower forecasting model. Section I gives introductory and background information on the development of the model and presents a brief but rigorous theoretical basis for the on-line system. Section II gives a description of the basic MANPOWER/DEMAND program indicating the function of the program, the detailed workings of the system options, and the language in which it is written. Appendices contain specifications of the data tapes and disc files involved, flow charts of the computer processes, and sample date input and output.

17a. Descriptors
- Applications
- Social and Behavioral Sciences
- Economics
- Forecasting (Manpower)

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement
No restriction on distribution.
Available from National Technical Information Service, Springfield, Virginia 22151
ECONOMIC RESEARCH GROUP WORKING PAPER NO. 5

THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL: DOCUMENTATION AND USER'S GUIDE

By

Roger H. Bezdek
R. Michael Lefler
Albert L. Meyers
Janet H. Spoonamore

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

October 15, 1971

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the U.S. Army Research Office-Durham under Contract No. DAHC04 72-C-0001.

Approved for public release; distribution unlimited.
THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL: DOCUMENTATION AND USER'S GUIDE

by

Roger H. Bezdek
R. Michael Lefler
Albert L. Meyers
Janet H. Spoonamore
ABSTRACT

This paper presents the preliminary documentation and user's guide for the Center for Advanced Computation economic and manpower forecasting model. Section I gives introductory and background information on the development of the model and presents a brief but rigorous theoretical basis for the on-line system. Section II gives a description of the basic MANPOWER/DEMAND program indicating the function of the program, the detailed workings of the system options, and the language in which it is written. Appendices contain specifications of the data tapes and disc files involved, flow charts of the computer processes, and sample data input and output.
YESTERDAY

[Text continues on the next page]
# TABLE OF CONTENTS

## I. DESCRIPTION OF THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL ........................... 1

A. The Development of the Model ......................... 1

B. Theoretical Basis for the Program .................... 3

## II. DESCRIPTION OF THE MANPOWER/DEMAND PROGRAM .................................................. 11

A. Function of the Program ................................ 11

B. Description of EDITFILE .............................. 12

C. MANPOWER/DEMAND Processing ....................... 13

D. The Data .............................................. 17

E. Use of MANPOWER/DEMAND ............................ 18

## III. APPENDICES

Appendix A: Tape Specifications .......................... 24

Appendix B: Flow Charts .................................. 25

Appendix C: Source List of MANPOWER/DEMAND ........... 36

Appendix D: Sample Input and Output .................... 52
I. DESCRIPTION OF THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL

A. The Development of the Model

The Center for Advanced Computation (CAC) economic and manpower forecasting model was initially conceived in the summer of 1969 by Roger Bezdek and Hugh Folk. At that time it was clear that there was, and would continue to be, a pressing need for a general, consistent economic model capable of analyzing both direct and indirect effects of specified changes in the economic environment on the economy and labor market. No model available was capable of simulating in detail the overall effects of changes in expenditures on different types of economic programs and activities which corresponded to alternate national priorities. The development of such a model was undertaken by Roger Bezdek for his Ph.D. thesis in Economics at the University of Illinois at Urbana-Champaign.

The Manpower Administration of the U. S. Department of Labor supported the major portion of Bezdek's dissertation research through a Doctoral Dissertation Grant. Although Bezdek originally planned to develop both historical and projected versions of this model, the latter development was prevented by severe methodological and statistical difficulties. Bezdek's original model pertained to the year 1960. Its development and the results of simulations conducted with it are described in detail in Manpower Implications of Alternate Patterns of Demand for Goods and Services.¹

¹ Bezdek [2]
In the spring of 1971, Bezdek and James Scoville developed for the National Urban Coalition a projected version of the basic model which pertained to the mid-1970's. It was used to simulate the effects on the U. S. labor market which would likely be generated by the Urban Coalition's proposed reorderings of national goals and priorities contained in _Counter-budget_. The model is discussed in detail in Bezdek and Scoville's _Manpower Implications of Reordering National Priorities_.

Early in 1971, personnel from the newly established Center for Advanced Computation of the University of Illinois became interested in continuing work on this model at the Center. The Center for Advanced Computation is an outgrowth of the ILLIAC IV project. It is an independent unit of the Graduate College which provides an interdisciplinary environment for research projects requiring specialized and sophisticated computer facilities. The development of this type of economic model required sophisticated and efficient computer software and, from the Center's point of view, this model offered a feasible and potentially significant application for ILLIAC IV.

Agreement was reached and an Economic Research Group (ERG) was established at the Center. Bezdek spent the summer of 1971 transferring the basic model onto the Center's computer facilities and improving and expanding it. Work continues in this direction, with Bezdek supervising development of the demand side of the model and Hugh Folk directing development of the supply side. This booklet is written as a user's guide to the demand-generating portion of the CAC model.

Input components of the model include data tapes, disc files, and computer card decks which can be integrated into a number of consistent

---

2 Bezdek and Scoville [6].
systems via a program which will be explained in Sections II and III of this paper. While a brief theoretical outline of the basic model is included here, no attempt is made to explain the detailed workings of the CAC model. For this information the interested reader is referred to the references at the end of this report.

B. Theoretical Basis for the Program

Adhering to the traditional assumptions of input-output analysis, the economy may be disaggregated into a specified number of sectors, each composed of firms producing a similar product or group of products. Each industry combines a set of inputs in fixed proportions to produce its output which it sells to other industries to meet their input requirements. Letting \( x_{ij} \) denote the quantity of the output of industry \( i \) required by industry \( j \) as an input, letting \( y_i \) denote the quantity of the output of industry \( i \) destined for use by the autonomous sectors, and letting \( X_i \) denote the gross output of industry \( i \), a static open input-output model may be represented by the following set of relationships:

\[
\begin{align*}
\sum_{j=1}^{n} x_{1j} + x_{12} + \cdots + x_{1n} + y_1 &= X_1 \\
\sum_{j=1}^{n} x_{2j} + x_{22} + \cdots + x_{2n} + y_2 &= X_2 \\
&\vdots \\
\sum_{j=1}^{n} x_{nj} + x_{n2} + \cdots + x_{nn} + y_n &= X_n
\end{align*}
\]

\( ^3 \) A more complete development of the theoretical model involved here along with a discussion of the problems involved in its empirical implementation is contained in Bezdek [5].
Since it has been assumed that each industry possesses a linear production function with fixed coefficients, the technical structure of an industry may be described by as many homogeneous linear equations as there are separate cost elements involved:

\[ x_{ij} = a_{ij}X_j, \quad x_{2j} = a_{2j}X_j, \quad \ldots, \quad x_{nj} = a_{nj}X_j \]

The \( a_{ij} \)'s are referred to as coefficients of production and, writing these relationships in the form of equation set (1), we have:

\[
\begin{align*}
\text{a}_{11}X_1 + \text{a}_{12}X_2 + \ldots + \text{a}_{1n}X_n + y_1 &= X_1 \\
\text{a}_{21}X_1 + \text{a}_{22}X_2 + \ldots + \text{a}_{2n}X_n + y_2 &= X_2 \\
&\ldots \ldots \ldots \ldots \\
\text{a}_{n1}X_1 + \text{a}_{n2}X_2 + \ldots + \text{a}_{nn}X_n + y_n &= X_n \\
\end{align*}
\]

The elements \( a_{ij} \) form an \( n \times n \) technical coefficient matrix \( A \) and, letting \( x \) denote an \( n \)-order gross output vector and \( y \) denote an \( n \)-order final demand vector, equation set (2) may be written as:

\[
(3) \quad x = Ax + y
\]

The final demand vector \( y \) is the vector of outputs available for disposal outside the processing sector and, letting \( I \) denote an identity matrix of order \( n \), from (3), we have:

\[
(4) \quad x - Ax = (I-A)x = y
\]

Assuming that the elements of \( A \) are nonnegative and that at least some of the \( a_{ij} \)'s are positive insures that \( (I-A) \) is nonsingular.

Equation (4) may thus be solved for \( x \):

\[
(5) \quad x = (I-A)^{-1}y
\]
\((I-A)^{-1}\) is the Leontief inverse matrix and its elements \(a_{ij}\) indicate the output requirements generated directly and indirectly from industry \(i\) by industry \(j\) per delivery of a dollar's worth of output to final demand.

The final demand vector itself may be viewed as the sum of a number of vectors each of which represents the industrial requirements of a distinct component of final demand. Letting \(u\) denote the number of final demand activities, \(g_j\) denote an \(n\)-by-1 vector specifying the direct output requirements of exogenous activity \(j\), and \(e_j\) denote a vector indicating the portion of final demand consumed by exogenous activity \(j\), we have:

\[
y = g_1 + g_2 + \cdots + g_u; \sum_{i} y_i = \sum_{j} e_j (L y_i); \sum e_j = 1
\]

Writing out the first part of (6) specifically yields linear equations of the following form:

\[
y_i = g_{i1} + g_{i2} + \cdots + g_{ij} + \cdots + g_{iu}; i = 1, 2, \ldots n
\]

Consider an arbitrary element \(g_{ij}\) defined above. As indicated, \(g_{ij}\) shows the direct requirements for input \(i\) generated by exogenous activity \(j\) and the magnitude of this demand will generally be determined by two factors: the total amount of final demand absorbed by activity \(j\), and the portion of this amount devoted to the purchase of input \(i\). The first factor may be expressed as: \(\sum_{i} g_{ij}\), while the second factor is written as:

\[g_{ij}/\sum_{i} g_{ij}\] Letting \(q_j = \sum_{i} y_i\), and \(p_{ij} = g_{ij}/\sum_{i} g_{ij}\), equation (7) can be rewritten as:

\[
y_i = p_{i1} q_1 + p_{i2} q_2 + \cdots + p_{ij} q_j + \cdots + p_{iu} q_u; i = 1, 2, \ldots, n
\]
or, letting \(P\) denote an \(n\)-by-\(u\) activity-industry matrix of activity input coefficients, and letting \(q\) denote a \(u\)-by-1 activity-expenditure vector:
\( y = Pq \)

\( p_{ij} \) indicates the direct requirements generated for the output of industry \( i \) per dollar of expenditure in final demand sector \( j \), and \( q_j \) shows the amount of expenditures allocated to activity \( j \).

Within this framework it is possible to determine the direct output requirements generated by alternate distributions of national expenditures among economic activities. Here it is assumed that the elements of the \( P \) matrix are fixed over a limited range of expenditure redistribution; the activity-industry matrix thus represents a transformation of expenditures on economic activities into direct output requirements from every industry in the economy. Using equation (5), these direct output requirements can be translated into total output requirements from every industry.

Next, output requirements must be related to employment demands. To accomplish this it is assumed that the employment requirements of an industry are proportional to the industry's output and that this relationship may be expressed in terms of labor input coefficients. Letting \( x_i^e \) denote the total employment in industry \( i \), the labor input coefficient for industry \( i \), \( \theta_i \), is:

\[
\theta_i = \frac{x_i^e}{X_i}; \quad i = 1, 2, \ldots, n
\]

Labor input coefficients are thus derived by dividing industry employment by industry output and they show the employment requirements of an industry per unit of output. Employment in each industry may be related to the components of final demand by substituting the values given for \( X_i \) in (5) into equation (9). Equations of the following form are derived:

\[
x_i^e = \theta_{i1} \hat{a}_{i1} y_1 + \theta_{i2} \hat{a}_{i2} y_2 + \ldots + \theta_{ij} \hat{a}_{ij} y_j + \ldots + \theta_{in} \hat{a}_{in} y_n;
\]

\[
i = 1, 2, \ldots, n
\]
or, letting \( x^e \) denote an \( n \)-by-1 vector of elements \( x_1^e, x_2^e, \ldots, x_n^e \), and letting \( \Theta \) denote a diagonal matrix whose elements are \( \vartheta_1, \vartheta_2, \ldots, \vartheta_n \), the equations in (10a) may be written in matrix notation as:

\[
(10b) \quad x^e = \Theta (I-A)^{-1}y
\]

Consider the matrix \( M \) defined as \( M = \Theta(I-A)^{-1} \) whose elements \( m_{ij} \) are:

\[
(11) \quad m_{ij} = \vartheta_i \hat{a}_{ij}; \quad i, j = 1, 2, \ldots, n
\]

Any element \( m_{ij} \) of \( M \) shows the total employment required within industry \( i \) in order for industry \( j \) to deliver a dollar's worth of output to final demand. Each row of \( M \) indicates the manner in which employment is generated within industry \( i \) by required activity in industries 1, 2, \ldots, \( n \) and each column of \( M \) illustrates how the employment generated by industry \( j \) is distributed among all industries. This matrix is referred to as an interindustry-employment matrix.

The necessary theoretical framework has now been constructed which permits the transformation of alternate priority-expenditure distributions into distinct interindustry-employment demand patterns. Letting \( \hat{Y} \) denote an \( n \)-by-\( n \) diagonal final demand matrix, the "total" interindustry-employment matrix, \( M^* \), is derived by postmultiplying \( M \) by \( \hat{Y} \):

\[
(13) \quad M^* = MY
\]

The elements of \( M^* \) show the total employment generated by and within every industry for a generated distribution of final demand reflecting a specified priority alternative.

The final step in the construction of the theoretical model involves the relation of interindustry-employment requirements to demands for occupational categories of manpower resources. This transformation is accomplished by using an industry-occupation matrix showing the occupational
distribution of industry employment for the time period under consideration. Denote this matrix by B: the rows of B represent industries, the columns of B represent occupations, and any element \( b_{ik} \) of B shows the percent of total employment in industry \( i \) composed of persons classified within occupation \( k \).

Let \( R \) denote a diagonal matrix whose elements \( r_{ii} \) are the row sums of the interindustry-employment matrix and thus show the total employment generated within industry \( i \). One type of manpower information is derived by premultiplying the industry-occupation matrix by \( R \):

\[
\begin{bmatrix}
  r_{11} & 0 & \cdots & 0 \\
  0 & r_{22} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & r_{nn}
\end{bmatrix}
\begin{bmatrix}
  b_{11} b_{12} \cdots b_{1h} \\
  \vdots \quad \vdots \\
  b_{n1} b_{n2} \cdots b_{nh}
\end{bmatrix}
= \begin{bmatrix}
  (\alpha) & (\alpha) & \cdots & (\alpha) \\
  s_{11} s_{12} \cdots s_{1h} \\
  \vdots & \vdots & \ddots & \vdots \\
  s_{n1} s_{n2} \cdots s_{nh}
\end{bmatrix}
\]

or

\[
(14b) \quad RB = S(\alpha)
\]

\( S(\alpha) \) is a "type \( \alpha \)" interindustry-occupation matrix and the elements \( S_{ik}(\alpha) \) of it show the total demands for occupation \( k \) generated within industry \( i \) by a specified distribution of national expenditures.

Letting \( \hat{M} \) denote the transpose of the total interindustry-employment matrix, a second type of manpower impact matrix is derived by premultiplying the industry-occupation matrix by \( \hat{M} \):
(15a)

\[
\begin{bmatrix}
m_{11} & m_{12} & \cdots & m_{1n} \\
m_{21} & m_{22} & \cdots & m_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
m_{n1} & m_{n2} & \cdots & m_{nn}
\end{bmatrix}
\begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1h} \\
b_{21} & b_{22} & \cdots & b_{2h} \\
\vdots & \vdots & \ddots & \vdots \\
b_{n1} & b_{n2} & \cdots & b_{nh}
\end{bmatrix}
= 
\begin{bmatrix}
s_{11} & s_{12} & \cdots & s_{1h} \\
s_{21} & s_{22} & \cdots & s_{2h} \\
\vdots & \vdots & \ddots & \vdots \\
s_{n1} & s_{n2} & \cdots & s_{nh}
\end{bmatrix}
\]

or:

(15b) \[ \hat{S} = S^{(B)} \]

\(S^{(B)}\) is referred to as a "type \(B\)" interindustry-occupation matrix and the elements \(s^{(B)}_{ik}\) of it show the demands for occupation \(k\) generated by industry \(i\). So while the type \(A\) manpower matrix indicates the occupational employment demand generated in every industry, the type \(B\) manpower matrix indicates the occupational employment demands generated by every industry.

Finally, a third type of manpower impact matrix can also be derived. Letting \(B^k\) denote an \(n\)-by-\(n\) diagonal matrix whose elements correspond to the \(k\)th column of \(B\), the third type of manpower matrix is derived by premultiplying \(B^k\) by the transposed total interindustry employment matrix:
(16a)

\[
\begin{bmatrix}
m_{11} & m_{21} & \cdots & m_{n1} \\
\vdots & \vdots & \ddots & \vdots \\
m_{1n} & m_{2n} & \cdots & m_{nn}
\end{bmatrix}
\begin{bmatrix}
b_{11}(k) & \cdots & b_{22}(k) & \cdots & b_{nn}(k) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\begin{bmatrix}
s_{11}(k) & s_{12}(k) & \cdots & s_{n1}(k) \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \vdots & \ddots \\
\vdots & \vdots & \vdots & \vdots & s_{nn}(k)
\end{bmatrix}
\]

or:

(16b)

\[
\tilde{P}_B(k) = S^{(k)}; \quad k = 1, 2, \ldots, h
\]

Since there are \( h \) columns in \( B \)--one for each occupational classification--it is possible to derive \( h \) of these \( S^k \) matrices. Each \( S^k \) matrix is essentially an interindustry-employment matrix for the \( k^{th} \) occupation, and an element \( s_{ji}^k \) shows the employment requirements for occupation \( k \) generated within industry \( i \) by industry \( j \). These matrices are referred to as occupational employment profiles, and they contain a highly detailed description of the structure of demands generated for an individual occupation by a specified distribution of national expenditures.

Taken together, these three types of manpower impact matrices provide a comprehensive and highly detailed picture of the employment impacts likely to result from the implementation of alternate types of economic and social programs and priorities.
II. DESCRIPTION OF THE MANPOWER/DEMAND PROGRAM

A. Function of the Program

The MANPOWER/DEMAND program performs two data handling functions. First, it edits existing data structures, the input matrices to the model. Secondly, and most importantly, it performs the algebraic computations set forth by the theoretical model previously described, for generation of manpower demands based on alternative expenditure patterns and technological assumptions. In effect, the program permits the researcher to experiment by varying the data matrices which represent the input to the economic model and to study the results generated by the MANPOWER/DEMAND processes. For example:

The experimenter executes the general model for a given set of 58 proposed expenditure alternatives, noting the generated occupational employment. By changing the activity-expenditure elements to represent a different pattern of resource allocation, he can analyze the generated effects on the labor market produced by the program. In this case, he must modify the q-vector which represents the expenditure distribution. After observing these results, he may then modify the interindustry-employment matrix or the industry-occupation matrix. Another run on the model gives different results and insight into more modifications.

The MANPOWER/DEMAND program is essentially a model of economic processes represented by several matrix operations but, in addition, its flexibility permits modification of the input matrices prior to execution of these operations.

---

4 The CAC model described in this report is in the process of being expanded and improved. At periodic intervals additional documentation and user guide reports shall be published which specify the changes which have been made.
B. Description of EDITFILE

Editing is performed on files prior to execution of the MANPOWER/DEMAND routine. The following modifications can be accomplished for each matrix:

1. Element-by-element addition, subtraction, multiplication, or division.
2. Overwriting a column or columns with a new column or columns.
3. Deletion of a column or columns, thereby reducing the size of the matrix.
4. Insertion of a new column or columns between existing ones, thereby increasing the size of the matrix.
5. Scaling an entire matrix by multiplying each row by a given constant.

EDITFILE is invoked by the standard ALGOL subroutine call as follows:

```
EDITFILE(<File name>,<number of cols>,<number of rows>);
```

It accepts on card input the following commands in free form:

```
ADD(<column number>)
SUBTRACT(<column number>)
MULTIPLY(<column number>)
DIVIDE(<column number>)
DELETE(<column number>)
INSERT(<column number>)
SCALE
```

After each command, data cards are included which contain the operands for the above commands in free-field format, i.e., integers or decimal numbers separated by commas. This routine is not invoked if editing of existing matrices is not desired.
C. MANPOWER DEMAND Processing

Program Overview

ERGWORKS is the routine which performs the major operations dictated by the system. It can accept as input different vectors reflecting various levels and distributions of expenditures on economic activities and it returns generated employment requirements classified by industry or by occupation. Alternately, the expenditure vector can be held constant and manpower demands can be generated by changing rows, columns, or individual coefficients within the various matrices to reflect changes in technology, labor productivity, or occupational displacement. ERGWORKS presently consists of four distinct sections: a core section, which is always executed, and three branches, only one of which is executed during a run. The choice of branch is a user-input control option and depends upon what information the user wants the program to calculate. Branch three, for example, selects a single occupation and gives detailed information on the structure of demand for that occupation.

The Core Section

The first step is to read in and check the list of control options provided by the user. The first option is the year. If the user asks for 1972, the program calls in the data from the four disk files corresponding to 1972's data; if the user asks for 1976, the program calls in the 1976 projected data. If the user asks for a year other than 1972 or 1976, the program will form the three required matrices and the required q-vector by performing a standard linear interpolation of the data for both years. The interpolation is performed by a special subroutine which subtracts 1972 from the input year, then multiplies the difference between the 1976 data and
the 1972 data by one fourth of the difference in the years and adds the result to the 1972 data. It should be noted, however, that linear interpolation may not necessarily represent economic change within an interindustry model.

The next option read in indicates whether the user wants only the final results and certain selected intermediate results or whether he also wants all intermediate matrices printed out in full. The third option specifies which branch the program is to take. If this option is three, the program also reads in a column number corresponding to the column of that occupational classification in the B-matrix. The program tests both the branch option and the column to assure that the former lies within the range one to three and that the latter lies within the range one to one hundred eighty-five.

The program then reads in the user-given title of the run and the q-vector. Both are printed immediately. Next, each of the fifty-eight rows of $P^T$ is multiplied by the corresponding element of q. This is mathematically equivalent to converting q into a diagonal matrix and post-multiplying this diagonal matrix by $P^T$. If the "fullprint" option has been called by the user, this new 58 x 89 matrix is printed, first by columns, then by rows.

The same basic code is used to print out each of the matrices required by the fullprint option. The title of the matrix is written, and a FOR-loop selects columns in groups of ten until less than ten columns remain. For each of the rows of the matrix the ten elements corresponding to the ten columns are printed. When less than ten columns remain to be written, the routine prints the end of each row of the matrix, the number of elements printed corresponding to the number of columns left. Once this is done, the same operation is carried out for the transpose of the matrix, effectively causing the matrix to be written by rows instead of columns.
After the new 58 x 89 matrix is printed (if it is to be printed), a y-vector is created which is eighty-nine elements long. The elements of the y-vector are the column sums of the 58 x 89 matrix.

The y-vector is printed and aggregated to eighty-five elements by deleting four selected elements, and the aggregated vector is printed.

Each of the columns of M is then multiplied by the corresponding element of the y-vector. This corresponds mathematically to post-multiplying the M-matrix by a diagonal matrix created from the y-vector. The row sums and column sums of this new matrix are computed and printed and, if the fullprint option is on, the entire matrix is printed.

After the above operation is completed the new M-matrix is aggregated to a 66 x 85 matrix. Row sums and column sums of this aggregated matrix are taken and a sixty-six order vector, designated r, is created from the sixty-six row sums. To avoid double-counting, the column sums are actually calculated over selected rows of the M-matrix. The row sums, the column sums, the sums of selected elements of both, and (if the fullprint option is on) the matrix itself are printed.

If the program is to branch to the second or third branch, the last operation completed by this core section consists of multiplying each row of the aggregated M-matrix by the corresponding element of the \( \mu \)-vector and (if the fullprint option is on) printing the resultant matrix.

Branch One

Branch one requires the r-vector computed above. But each element of this vector is first multiplied by the corresponding element of the \( \mu \)-vector. Each row of the B-matrix is then multiplied by the corresponding element of the modified r-vector, forming a matrix called \( S^{(\alpha)} \). If the fullprint option is on \( S^{(\alpha)} \) is printed. In both cases row sums and column sums are calculated
for $S^{(a)}$ over selected columns and rows, respectively, to avoid double-counting. These row sums and column sums are printed and totaled and the program run then terminates.

Branch Two

Branch two postmultiplies the transpose of the modified aggregated M-matrix by the B-matrix, producing an 85 x 185 matrix called $S^{(β)}$. This operation is modified, however, by multiplying only selected columns and rows to avoid double-counting. If the fullprint option is on, the $S^{(β)}$ matrix is printed out.

The one hundred eighty-five column sums are computed and totaled (over selected rows to avoid double-counting), then printed. Similarly, the eighty-five row sums are computed and totaled (over selected columns to avoid double-counting), then printed. This terminates execution.

Branch Three

Branch three begins by selecting and printing a column from the B-matrix. Each of the sixty-six columns of the transpose of the modified aggregate M-matrix is then multiplied by the corresponding element of this column vector to form an 85 x 66 matrix called $S^{(k)}$. In the $S^{(k)}$ matrix $h$ represents the column of the B-matrix selected, where $1 \leq k \leq 185$. If the fullprint option is on, this matrix is printed.

The sixty-six column sums are computed and totaled (over selected rows to avoid double-counting), then printed. Similarly, the eighty-five row sums are computed and totaled (over selected columns to avoid double-counting), then printed. This terminates execution.
D. The Data

The input data for both routines reside on the same set of disk and card files. The eight disk files contain projected data for years 1972 and 1976 derived from data which were obtained from the Office of Business Economics, the Bureau of Labor Statistics, the Harvard Economic Research Project, the National Planning Association, and the National Urban Coalition, and which were, in part, derived independently by Roger Bezdek. These disk files are matrix representations for the $58 \times 89$ activity-industry matrix, designated by "$P$", the $85 \times 85$ interindustry-employment matrix, designated by "$M$", the $66 \times 185$ industry-occupation matrix, designated by "$B$", and a 66-order vector designated by "$u$".

The $P$ and the $M$ matrices are stored and handled in transposed form within the program. These files can be inputted directly to the model or modified first by EDITFILE and then used as direct inputs. The card file called CARD contains, first of all, the specifications for any editing to be done on the above disk files. Following these specifications are the run time options for ERGWORKS. These options include the projected year to be run, the fullprint option, and the branch of the routine to be invoked. Following the option, the $q$-vector (the specified expenditure distribution) is read in.
E. Use of MANPOWER DEMAND

Tapes:

MANPOWER/DEMAND is written in Burroughs B6500 ALGOL language. The source program as well as all data files are stored on nine-track tapes in the B6500 room, Room 10, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois. At present, several versions of the program are saved at points in time to enable programming changes to be made without risk to previous progress. In the appendix a list provides tape numbers and tape contents.

The tape name ERG is used to access any of the tapes. In this and any discussion of B6500 usage, the reader is referred to the Little Golden Book of the B6500\(^5\) for operating system details.

Control Cards:

In order to run MANPOWER/DEMAND on the B6500, a set of control statements must be entered in card form or from a terminal. Listed below are the cards which can be used:

The tape must first be loaded onto disk by the following:

```
?COPY ERG/ = FROM ERG
```

In order to compile the program:

```
?COMPILE ERG/MANPOWER/DEMAND WITH ALGOL LIBRARY

?ALGOL FILE CARD = ERG/OCTI DISK SERIAL

?END
```

The execution is accomplished by the following:

```
?EXECUTE ERG/MANPOWER/DEMAND

?DATA CARD
```
2, 1, 1,

TEST DATA

103351 48160

and other data cards

?END

(Note: ? is a control character on B6500 and is punched by multipunch 1, 2, 3.)

The execution takes about six minutes of processing time on the B6500 which amounts to about 15 minutes real time in the machine.

Sample Experiments:

Statement of Problem 1: Replace the first 12 columns of the 1972 P matrix with a given set of data, change column 17 to 17a and 17b and run the program for branch one.

Method of Solving: Use the EDITFILE procedure to modify the ERG/MATRIX/PI file. The following code must be inserted into the program to form executable code.

L: = 57

EDITFILE (Pl, L, 89);
REWIND (Pl);
ERGWORKS (L, 89, 85, 66, 185);
END

The program is then recompiled, and executed with the following card input:

?DATA CARD

REPLACE (1)
(Data cards to replace second column)

REPLACE (2)
(Data cards to replace second column)

. .

REPLACE (12)
(Data cards to replace 12th column)
REPLACE (17)

(Data cards to replace 17th column)
INSERT (18)

(Data cards to insert between the 17th and 18th columns—changes old 18th to 19th column)
STOP

72,1,1
TEST DATA
103351 418160 ...

(Change 57 to 58 long vector here)
14561 ...

?END

Statement of Problem 2: Scale the rows of the 1972 M matrix by a set of constants. Run the program using the modified P matrix above for branch one.

Method of Solving: Again use the EDITFILE procedure to modify the ERG/MATRIX/M3 file. If possible, store the modified P-matrix from the last example; otherwise, include the changes here as in the above program.

The program instructions follow:

L: = 58;

EDITFILE (M3, 85, 85);
REWIND (M3);
ERGWORKS (L, 89, 85, 66, 185);
END

The above statements are to be inserted in the executable code section of the program. Recompile and execute with the following:

?DATA CARD
MULTIPLY (1)
(85 elements to multiply row 1)
MULTIPLY (2)
(85 elements to multiply row 2)
.
.
.
MULTIPLY (85)
(85 elements to multiply row 85)
STOP
72, 1, 1,
TEST DATA
103351 418160 ...
.
.
.
.
14561 ...
? END
REFERENCES


[9] Meyers, Albert L.  An Introduction to the Pointer Mechanism in  
    Burroughs Corporation Algol.  ILLIAC IV Document no. 215,  
    University of Illinois at Urbana-Champaign, Urbana, Illinois,  
    May 1970.
Appendix A: Tape Specifications

B6500

<table>
<thead>
<tr>
<th>Tape Number</th>
<th>Name</th>
<th>File Names</th>
<th>Description of File</th>
</tr>
</thead>
<tbody>
<tr>
<td>202,204</td>
<td>ERG</td>
<td>(1) ERG/MATRIX/P1</td>
<td>1972 P-MATRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) ERG/MATRIX/P2</td>
<td>1976 P-MATRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) ERG/MATRIX/M3</td>
<td>1972 M-MATRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4) ERG/MATRIX/M4</td>
<td>1976 M-MATRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(5) ERG/MATRIX/B1</td>
<td>1972 B-MATRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6) ERG/MATRIX/B2</td>
<td>1976 B-MATRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(7) ERG/MATRIX/MU1</td>
<td>1972 - Mu vector</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(8) ERG/MATRIX/MU2</td>
<td>1976 - Mu vector</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(9) ERG/MANPOWER/SOURCE</td>
<td>Sourcecode for economic model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(10) ERG/MANPOWER/DEMAND</td>
<td>Compiled version of ERG/MANPOWER/DEMAND</td>
</tr>
</tbody>
</table>

122         ERG

files 1-9 same as 202, 204

(10) ERG/OCT1  Source code for economic model and edit routine.

(11) ERG/MANPOWER/DEMAND  Compiled version of (10.)

568         ERG

files 2-11 same as 122

(1) ERG/MATRIX/P1  modified in experiment
Appendix B

Flow Charts
PROCEDURE EDGWORKS

READ IN YEAR, FULL PRINT AND BRANCH

CHECK YEAR AND REDUCE TO '72 THRU '76

YEAR = '72?
YES READ FILES PI, MB, B1 AND MUI
NO

YEAR = '76?
YES READ FILES PB, M4, B2, MUI
NO

INTERPOLATE FILES FOR '73, '74, '75

BRANCH > 3 OR < 1
YES ERROR RETURN

BRANCH? COLUMN WRONG?
ERROR RETURN

READ TITLE AND Q-VECTOR

TOTAL Q-VECTOR AND PRINT Q-VECTOR AND TOTAL

FORM INDUSTRY ACTIVITY MATRIX P — P10

IF FULL PRINT, PRINT P AND TRANSPOSE

FORM FINAL DEMAND VECTOR Y(J) — SUM OF I^TH COLUMN OF P

PRINT Y AND SUM OF ELEMENTS

AGGREGATE, Y TO BS, LONG, AND PRINT

FORM INTER IND. EMP. MATRIX M4 — MAV

FORM ROW SUMS, PRINT TOTAL, PRINT OF M

A

B
IF FULL PRINT
PRINT INT.
IND. EMP.
MATRIX

AGGREGATE THE
INTER IND. EMP.
MATRIX TO
46 X 85

TAKE ROW
AND COL.
SUMS OF
M. PRINT

IF FULL PRINT
PRINT M, THE
INT. IND. EMP.
MATRIX

BRANCH
> 1
?

NO

MODIFY M-MATRIX
M ← MXMU

YES

IF FULL PRINT
PRINT MODIFIED
M MATRIX

SELECT BRANCH
1, 2 OR 3

R - ROW SUMS OF
M IS MODIFIED
R ← R X MU

PRINT R AND
TOTAL OVER
SPECIAL ROWS

FORM S (Q)
ON B,
B ← B X R

PRINT S (Q)
IF FULL PRINT

FORM IND. EMP.
VECTOR-ROW-
SUMS OVER
SPECIAL COLS
OF S (Q)

PRINT OUT
IND. EMP.
VECTOR AND
TOTAL

FORM OCC. EMP.
VECTOR-COLUMN
SUMS OVER
SELECTED ROWS

PRINT OUT
OCC. EMP.
VECTOR AND
TOTAL

RETURN
FORM 5(#) BY MXB REPLACING B

IF FULL PRINT PRINT 5(βETA) MATRIX

COMPUTE TOTAL OCCUR EMP. VECTOR = COL. SUMS OF 5(#)

PRINT COL. SUMS AND THEIR TOTAL

COMPUTE TOTAL IND. EMP. VECTOR = ROW SUMS OF 5(#)  

PRINT ROW SUMS AND THEIR TOTAL

RETURN

3

THIS BRANCH GIVES DETAILED INFO. ABOUT GIVEN OCC.

PRINT SELECTED OCCUPATION, i.e., COLUMN OF B MATRIX

STORE COLUMN IN R-VECTOR

FORM 5(h) MATRIX, MXR REPLACING M

IF FULL PRINT WRITE OUT 5(h) MATRIX

FORM COL. SUMS OF 5(h) GETTING EMP. GENERATED IN

SUM THESE COL. SUMS FOR SELECTED COLS AND PRINT

FORM ROW SUMS OF 5(h) FOR SPECIAL ROWS, GET EMP. GEN'D BY

FORM TOTAL OF ROW SUMS AND PRINT

RETURN
PROCEDURE INTERPOLATE

FORM F, SCALE FACTOR = (YEAR - 12) / 4

READ CORRESPONDING ROWS OF P1 AND P2 FILES

SCALE P1 ROWS BY ADDING F% (P2 - P1)

READ CORR. ROW OF M3, M4 AND INTERPOLATE

READ CORR. ROW OF B1, B2 AND INTERPOLATE

READ MU1 AND MU2 AND INTERPOLATE

RETURN
PROCEDURE PRTMATRIX

PROCEDURE PRTMATRIX
A, NROWS, NCOLS, HEADING

DETERMINE THE FIRST ELT IN LAST FULL ROW OF PRINT

FULLROW := NROWS - NROWS MOD 10 - 9

ITERATE K FROM 1 STEP 10 UNTIL FULLROW

L ← K + 9

IF HEADING
NO?

WRITE "ROWS K THRU L"

WRITE NCOLS ROWS EACH ROW CONTAINING 10 ROW ELEMENTS I.E. K THRU L, A[I, J]
DO LAST COLUMN FOR NON FULL ROW

CHECK HEADING AGAIN TO SEE WHICH ONE TO PRINT AS BEFORE

NOW PRINT REMAINING ROWS L THRU N, COLUMN 1 TO NCOLS, A[I,J]

RETURN
PROCEDURE PRTRANSP

A, NROWS, NCOLS, HEADING

DETERMINE FIRST ELT OF LAST FULL ROW OF PRINT CONTAINING TEN ELEMENTS, FULLROW := NCOL - NCOLS MOD 10 - 9

ITERATE K FROM 1 STEP 10 UNTIL FULLROW

L ← K + 9

IF HEADING = "?"

WRITE "ROWS K THRU L"

WRITE "COL K THRU L"

WRITE NROWS ROWS EACH CONTAINING 10 COLUMN ELEMENTS, I.E., K THRU L, A[J, I]
DO LAST ROWS FOR NON FULL COLS

CHECK HEADING AGAIN TO SEE WHICH ONE TO PRINT

NOW PRINT REMAINING COLUMNS L THRU N, ROW 1 TO N ROWS OF A[J, I]

RETURN
PROCEDURE EDITFILE

T2M ← 2*M

READ FILE INTO A[0,2*M,M]

INITIALIZE
B[i] ← 1, I=1,...,M

APP, SUBTRACT
MULTIPLY, DIVIDE

GET OPERANDS
AND CALL
PUTTER

OVERWRITE

READ ROW FROM
CARD INTO ROW
OF A.

DELETE

MOVE B[i]
DOWN 1 ELT
STARTING AT INPEX

INSERT

B[INDEX] ← M ← M+1

READ NEW ROW
INTO A[M,*]

READ IN SCALE
FACTOR INTO E[M]

MULTIPLY E'X A
AND REPLACE A

J
PROCEDURE "SCANNER J
WRITE OUT C(I,J) ONTO FILE
RETURN

PROCEDURE DOTTER
PERFORM INDICATED OPERATION +, -, *, / ON A AND B
PUT RESULT IN VECTOR A
RETURN

PROCEDURE NUMB
NUMB ← ROW NUMBER TO EDITED
RETURN

READ CONTROL STATEMENT FOR VERB AND INDEX

"ADD"?
NO

"SUB"?
NO

"MULT"?
NO

"DIV"?
NO

"OVER"?
NO

"DEL"?
NO

"INS"?
NO

"SCAL"?
NO

SCANNER ← SNUM ← 0
SCANNER ← SNUM ← 1
SCANNER ← SNUM ← 2
SCANNER ← SNUM ← 3
SCANNER ← SNUM ← 4
SCANNER ← SNUM ← 5
SCANNER ← SNUM ← 6
SCANNER ← SNUM ← 7
INDEX ← NUMB
RETURN
Appendix C

Source List of MANPOWER/DEMAND
BEGIN

DEF FILE SEG = LABEL NF*SEGMENT;

FILE LINE(KIND) = 135, BUFFER = 2, MAXRECSIZE = 22, INTHDLE = 4, MYUSE = 2;
FILE CARDS(KIND) = 9, BUFFER = 2, MAXRECSIZE = 14, INTHDLE = 4;
FILE PI(KIND) = 1, ACCESS = 0, TITLE = "ERG/MATRIX/P1", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;
FILE P2(KIND) = 1, ACCESS = 0, TITLE = "ERG/MATRIX/P2", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;
FILE M1(KIND) = 1, ACCESS = 0, TITLE = "ERG/MATRIX/M1", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;
FILE M4(KIND) = 1, ACCESS = 0, TITLE = "ERG/MATRIX/M4", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;
FILE B1(KIND) = 1, ACCESS = 0, TITLE = "ERG/MATRIX/B1", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;
FILE B2(KIND) = 1, ACCESS = 0, TITLE = "ERG/MATRIX/B2", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;
FILE M11(KIND) = 1, ACCESS = 0, TITLE = "ERG/VECTOR/M1", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;
FILE M22(KIND) = 1, ACCESS = 0, TITLE = "ERG/VECTOR/M2", BUFFER = 2,
MAXRECSIZE = 14, BLOCKSIZE = 420;

FORMAT IN FR(111), F9(0,9F4,4*3);
FORMAT OUT HEAD1("ELEMENT VALUE") F14(F20.5),
HEAD2("P RUN", "RUN SUM") F2("TOTAL", F19,5),
HEAD3("COLUMN COLUMNS") F7(13,9(X1,R11,4)),
HEAD4("COLS", 14") F7(13,9(X1,R11,4)),
HEAD5("ROW", 14") F7(13,9(X1,R11,4)),
FREMESS("**INCORRECT OPTION OR GRAMMY")
REAL ARRAY P[11] 0.644, M5[185, 0.4311];
INTEGER I,J,K,L;
LABEL FORD;

LINE PRINT ROUTINES

A PROCEDURE PRINTMATRIX AND PRINTTRANSP ARE USED FOR FULLPRINT
A TO PRINT MATRIX OR ITS TRANSPOSE
PROCEDURE PRINTTRANSP(A, NRHS, NCOLS, HEADING);
VALUE NRHS, NCOLS, HEADING;
REAL ARRAY A(0,0);
INTEGER NRHS, NCOLS, HEADING;
BEGIN

INTEGER FULLROW, I, J, K, L;
FULLROW = NCOLS - NCOLS MUU 10 = 0;
FOR K1 = 1 STEP 10 UNTIL FULLROW DO BEGIN
L = K + 1;
IF HEADING = 0 THEN WRITE(LINE, HEAD, K, L) ELSE
WRITE(LINE, HEAD4, K, L);
FOR J = 1 STEP 1 UNTIL NRHS DO WRITE(LINE, F10), FOR II = K STEP 1 UNTIL L
DO A(I, J) = 0;
END;
IF NCOLS > L THEN BEGIN
IF HEADING = 0 THEN WRITE(LINE, HEADS, L + 1, NCOLS) ELSE
WRITE(LINE, HEAD4, L + 1, NCOLS);
FOR J = 1 STEP 1 UNTIL NRHS DO WRITE(LINE, F7, J FOR I = L + 1 STEP 1 UNTIL NCOLS MO A(I, J));
END;
WRITE(LINE[STEP II]);
END PRTRANSP;
PROCEDURE PRTRANS(A,NROWS,NCOLS,HEADING);
VALUE NROWS,NCOLS,HEADING;
INTEGER NROWS,NCOLS,HEADING;
REAL ARRAY A[0,0];
BEGIN
INTEGER FULLROW,I,J,L,K;
FULLROW = NROWS - NROWS MOD 10 - 9;
FOR K:= 1 STEP 10 UNTIL FULLROW DO BEGIN
L := K + 9;
IF HEADING = 0 THEN WRITE(LINE,HEAD4,K,L) ELSE WRITE(LINE,HEAD5,K,L);
FOR J:= 1 STEP 1 UNTIL NCOLS DO WRITE(LINE,F10,J,FOR I:= K STEP 1 UNTIL L DO A[I,J]);
END
IF NROWS > L THEN BEGIN
IF HEADING = 0 THEN WRITE(LINE,HEAD4,L+1,NROWS) ELSE WRITE(LINE,HEAD5,L+1,NROWS);
FOR J:= 1 STEP 1 UNTIL NCOLS DO WRITE(LINE,F7,J,FOR I:= L+1 STEP 1 UNTIL NROWS DO A[I,J]);
END
WRITE(LINE(SKIP 1));
END PRTRANS;
END OF LINE PRINT ROUTINES

PROCEDURE EDITFILE(FIL,M,N);
VALUE M,N;
INTEGER FIL,N;
BEGIN
LABEL OUT,EOF;
ARRAY A[0,2*M-1,N];
B[0,2*M];
C[0,N];
D[0,13];
F[0,2*M];
POINTER P,Q;
INTEGER I,J,
INDEX,SNUM,T2M;
DEFINE
G(I,J) = A(R[I],J);
GETC = READ(CARD,*/FOR I:=1 STEP 1 UNTIL N DO C(I));
DOTS = BEGIN
INTEGERPROCEDURENUMB;
BEGIN
SCANP;UNTIL="(";
SCANP;UNTILINALPHA;
SCANSWHILEINALPHA;
NUMB:=INTEGER(P-DELTA(P-1));
ENDNUMB;

PROCEDUREDITTER(A*R*1*);
VALUE
INTEGERI_,j; ARRAYA*,*{0};
BEGIN
INTEGERK;
CASEI OF
BEGIN
FORK:=1STEP1UNTILJ DONK1:=**N{K1};
ENDCASESTATE:LEFT;
ENDDITTER;

INTEGERPROCEDUREScanner;
BEGIN
LABELLEFT;
READ(CARD*14*);{1}{LEFT:};
LEFT:SCANNER=SNUMF:=7;
SCANP;INTERFER(O)WHILE="";
SCANNER=SNUMF:=
IFP=AND" THEN0FLSF
IFP=SUN" THEN1FLSF
IFP=MULT" THEN2ELSE
IFP=DTV" THEN3FLSF
IFP=DFR" THEN4FLSF
IFP=DFL" THEN5FLSF
IFP=INS" THEN6ELSE
IFP=SCAL" THEN7FLSF
IFP=ANJU" THEN8FLSF9;
IFSNUMFNEQ7ANDSNUMFNEW9ANDSNUMFNEQ9THEN
INDEX:=NUMB;
ENDScanner;

BEGINEXECUTABLEEDITFILECIDE

%INITIALIZE

T2M:=M+M;
FORI:=1STEP1UNTILMDO
READ(FIL); FOR J := 1 STEP 1 UNTIL N DO A[I,J];
FOR I := 1 STEP 1 UNTIL M DO B[I];
%
% MAIN PROGRAM
%
WHILE TRUE DO CASE SCANNER UF
BEGIN
  DOTS;
  DOTS;
  DOTS;
  DOTS;
  READ(CARD,F9, FOR I := 1 STEP 1 UNTIL N DO G(INDEX+I));
BEGIN
  REPLACE POINTER(A(INDEX)) BY POINTER(B(INDEX+1)) FOR
  T2M=INDEX-1 WORDS;
  M := M+1;
END;
BEGIN
  B(INDEX) := M := M+1;
  FOR II := M STEP -1 UNTIL INDEX DO R[I] := B[I+1];
  READ(CARD,F9, FOR J := 1 STEP 1 UNTIL N DO A[M,J]);
END;
BEGIN
  READ(CARD,F9, FOR I := 1 STEP 1 UNTIL M DO E[I]);
  FOR J := 1 STEP 1 UNTIL N DO A[I,J];
END;
BEGIN
  READ(CARD,F9, FOR I := 1 STEP 1 UNTIL M DO E[I]);
  FOR J := 1 STEP 1 UNTIL N DO A[I,J];
END;
GU OUT;
END CASE;
%
OUT;
%
% WRAP-UP
%
REWIND(FIL);
FOR I := 1 STEP 1 UNTIL M DO
  WRITE(FIL,F9, FOR J := 1 STEP 1 UNTIL N DO G(I,J));
END EDITFILE;
%
PROCEDURE ERGWORKS(I,T,J,K,KK,LL,MM);
VALUE
  I,T,J,K,KK,LL,MM;
INTEGER
  I,T,J,K,KK,LL,MM;
BEGIN
REAL ARRAY A[0:13], M[L:1], Q[0:11], K(0:11), R[0:11], SH[0:11],
  P[0:11], O, I,J,K,L, BRANCH, YEAR, COLUMN;
REAL SUM, TOTAL;
LABEL FINISH;
%  INTERPOLATION PROCEDURE FOR YEARS OTHER THAN '72 OR '76
PROCEDURE INTERPOLATE:
BEGIN
REAL F;
F := (YEAR - 72)/4;
FOR I := 1 STEP 1 UNTIL II DO BEGIN
READ(P1, FOR J := 1 STEP 1 UNTIL JJ DO P(I,J));
READ(P2, FOR J := 1 STEP 1 UNTIL JJ DO Y(J));
FOR J := 1 STEP 1 UNTIL JJ DO R(I,J) := P(I,J) + F * (Y(J) - Pt + J));
END;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
READ(M3, FOR J := 1 STEP 1 UNTIL KK DO M(I,J))
READ(M4, FOR J := 1 STEP 1 UNTIL KK DO Y(J));
FOR J := 1 STEP 1 UNTIL KK DO M(I,J) := M(I,J) + F * (Y(J) - M(I,J));
END;
FOR I := 1 STEP 1 UNTIL LL DO BEGIN
READ(R1, FOR J := 1 STEP 1 UNTIL MM DO R(I,J));
READ(R2, FOR J := 1 STEP 1 UNTIL MM DO S4(J));
FOR J := 1 STEP 1 UNTIL MM DO R(I,J) := R(I,J) + F * (S4(J) - R(I,J));
END;
READ(MU1, FOR I := 1 STEP 1 UNTIL LL DO MU(I));
READ(MU2, FOR I := 1 STEP 1 UNTIL LL DO R(I,I));
END.

DEFINE SPECIAL LISTS FREQUENTLY USED FOR OUTPUT

DEFINE SPECIAL ROWS = I := 2, 3, 5, 6, 7, 8, 9, 11, 12, 14, 15, 16, 18, 20, 21, 23, 25, 28, 30, 31, 34, 37, 38, 47, 49, 50, 51, 54, 55, 58, 59, 60, 61, 62, 63, 65, 67, 70, 81, 83, 84, 95, 112, 116, 123, 136, 139, 143, 153, 158, 167, 168, 172, 177, 184, 211

DEFINE SPECIAL COLUMNS = J := 2, 14, 21, 26, 31, 33, 38, 47, 49, 60, 69, 70, 71, 81, 83, 94, 95, 112, 116, 123, 136, 139, 143, 153, 158, 167, 168, 172, 177, 184, 211

BEGIN PROCESSING BY READING IN OPTION CARD AND DATA
READ(CARD, FOR YEAR, I, BRANCH);
IF I = 0 THEN FULLPRINT := FALSE ELSE FULLPRINT := TRUE;
IF YEAR GT 1900 THEN YEAR := YEAR - 1900;
IF YEAR = 72 THEN BEGIN SEG;
FOR I := 1 STEP 1 UNTIL II DO
READ(P1, FOR J := 1 STEP 1 UNTIL JJ DO P(I,J));
FOR I := 1 STEP 1 UNTIL KK DO
READ(M3, FOR J := 1 STEP 1 UNTIL KK DO M(I,J));
FOR I := 1 STEP 1 UNTIL LL DO
READ(R1, FOR J := 1 STEP 1 UNTIL MM DO R(I,J));
READ(MU1, FOR I := 1 STEP 1 UNTIL LL DO MU(I));
END
ELSE IF YEAR = 76 THEN BEGIN SEG;
FOR I := 1 STEP 1 UNTIL II DO
READ(P2, FOR J := 1 STEP 1 UNTIL JJ DO P(I,J));
FOR I := 1 STEP 1 UNTIL KK DO
READ(M4, FOR J := 1 STEP 1 UNTIL KK DO M(I,J));
FOR I := 1 STEP 1 UNTIL LL DO
READ(R2, FOR J := 1 STEP 1 UNTIL MM DO R(I,J));
READ(MU2, FOR I := 1 STEP 1 UNTIL LL DO MU(I));
IF BRANCH GTR 3 OR BRANCH LSS 1 THEN BEGIN
  WRITE(LINE*ERRMESS1);
  GO TO FINISH;
END;

IF BRANCH = 3 THEN READ(CARD.*COLUMN);
IF BRANCH = 3 AND (COLUMN LSS 1 OR COLUMN GTR MM) THEN BEGIN
  WRITE(LINE.ERRMESS1);
  GO TO FINISH;
END;

READ AND WRITE TITLE AND Q-VECTOR

READ(CARD,13A[*]);
WRITE(LINE,13A[*]);
I = 0;
THRU ( (II DIV 8))
  DU READ(CARD,F8,FOR J = 1 STEP 1 UNTIL B DO Q[I-I+1]);
READ(CARD,F8,THRU II MOD 8 DO Q[I-I+1]);
WRITE(LINE,"/" ALTERNATIVE EXPENDITURE VECTOR"));
WRITE(LINE,HEAD1);
TOTAL = 0;
FOR I = 1 STEP 1 UNTIL II DO BEGIN
  TOTAL = TOTAL + Q[I];
  WRITE(LINE,F1,I,Q[I]);
  END;
WRITE(LINE,HEAD2);

FORM INDUSTRY ACTIVITY MATRIX (OVERLAYING P)

FOR J = 1 STEP 1 UNTIL II DO
  FOR I = 1 STEP 1 UNTIL JJ DO P[I,J] = P[I,J] * Q[I];

ROUTINE TO PRINT INDUSTRY ACTIVITY MATRIX AND ITS TRANSPOSE

IF FULLPRINT THEN BEGIN
WRITE(LINE,"INDUSTRY ACTIVITY MATRIX"));
FOR K = 1 STEP 10 UNTIL 41 DO BEGIN
  L = K + 91
  WRITE(LINE,HEAD4,K,L);
  FOR J = 1 STEP 1 UNTIL JJ DO
    WRITE(LINE,F10,J,FOR I = 1 K STEP 1 UNTIL L DO P[I,J]);
  END;
WRITE(LINE,HEAD4,51,II);
FOR J = 1 STEP 1 UNTIL JJ DO
  WRITE(LINE,F7,J,FOR I = 51 STEP 1 UNTIL II DO P[I,J]);
WRITE(LINE,SKIP 1));
WRITE(LINE,"TRANSPOSE PRINT OF INDUSTRY ACTIVITY MATRIX"));
FOR K = 1 STEP 10 UNTIL 71 DO BEGIN
  L = K + 91
  WRITE(LINE,HEAD5,K,L);
  FOR J = 1 STEP 1 UNTIL II DO
    WRITE(LINE,F10,J,FOR I = 1 K STEP 1 UNTIL L DO P[J,I]);
  END;
WRITE(LINE,HEAD5,81,JJ);
FOR J = 1 STEP 1 UNTIL II DO
  WRITE(LINE,F10,J,FOR I = 81 STEP 1 UNTIL JJ DO P[J,I]);
WRITE(LINE(SKIP 11));

END;

GENERATE FINAL DEMAND VECTOR AND PRINT

WRITE(LINE(" GENERATED FINAL DEMAND VECTOR")); WRITE(LINE("HEAD1"));
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL JJ DO BEGIN
  Y[I] := 0;
  FOR J := 1 STEP 1 UNTIL II DO Y[I] := Y[I] + P[J*I];
  WRITE(LINE(F1*I,Y[I]));
  TOTAL := TOTAL + Y[I]; END;
WRITE(LINE(SKIP 11,F2,TOTAL));

AGGREGATE FINAL DEMAND VECTOR AND PRINT

WRITE(LINE(" GENERATED FINAL DEMAND VECTOR")); WRITE(LINE("HEAD1"));
Y[00] := Y[81];
Y[01] := Y[22];
Y[02] := Y[31];
Y[03] := Y[41];
Y[04] := Y[44];
Y[05] := Y[41];
Y[06] := Y[41];
Y[07] := Y[41];
Y[08] := Y[41];
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  TOTAL := TOTAL + Y[I];
  WRITE(LINE(F1*I,Y[I])); END;
WRITE(LINE(SKIP 11,F2,TOTAL));

GENERATE THE INTERINDUSTRY EMPLOYMENT MATRIX

WRITE(LINE(" ROW AND COLUMN SUMS OF INTERINDUSTRY EMPLOYMENT MATRIX")); WRITE(LINE("HEAD3"));
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  SUM := 0;
  FOR J := 1 STEP 1 UNTIL KK DO BEGIN
    M[I,J] := M[I,J] * Y[I];
    SUM := SUM + M[I,J];
  END;
  WRITE(LINE(F1*I,SUM));
  TOTAL := TOTAL + SUM;
END;
WRITE(LINE(SPACE 2),F2,TOTAL);
WRITE(LINE("HEAD2"));
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  SUM := 0;
  FOR J := 1 STEP 1 UNTIL KK DO SUM := SUM + Y[J,I];
  WRITE(LINE(F1*I,SUM));
  TOTAL := TOTAL + SUM;
END;
WRITE(LINE(SKIP 11,F2,TOTAL));

ROUTINE TO PRINT INTERINDUSTRY EMPLOYMENT MATRIX AND TRANSPOSE

IF FULLPRINT THEN BEGIN
WRITE(LINE(" INTERINDUSTRY EMPLOYMENT MATRIX"));
FOR K := 1 STEP 10 UNTIL 71 DO BEGIN

L := K + 9;
WRITE(LINE,HEAD4,K,L);
FOR J := 1 STEP 1 UNTIL KK DO
  WRITE(LINE,F10,J, FOR I := K STEP 1 UNTIL L DO M[I,J]);
END;
WRITE(LINE,HEAD4,R1,KK);
FOR J := 1 STEP 1 UNTIL KK DO
  WRITE(LINE,F7,J, FOR I := R1 STEP 1 UNTIL KK DO M[I,J]);
WRITE(LINE,SKIP 1);
% WRITE(LINE,"\* TRANSPOSE PRINT OF INTERINDUSTRY EMPLOYMENT MATRIX\*\*\")
FOR K := 1 STEP 10 UNTIL 71 DO BEGIN
  L := K + 9;
  WRITE(LINE,HEAD5,K,L);
  FOR J := 1 STEP 1 UNTIL KK DO
    WRITE(LINE,F10,J, FOR I := K STEP 1 UNTIL L DO M[J,I]);
  END;
END;
%
AGGREGATE THE INTERINDUSTRY EMPLOYMENT MATRIX TO LL X KK
(OVERLAYING ITSELF) AND TAKE THE ROW AND COLUMN SUMS
%
FOR I := 1 STEP 1 UNTIL KK DO BEGIN SEG:
  M[I,2] := M[I,1] + M[I,2];
  M[I,3] := M[I,3] + M[I,4];
  M[I,5] := M[I,2] + M[I,3];
  M[I,6] := M[I,5] + M[I,6];
  M[I,7] := M[I,7];
  M[I,8] := M[I,8];
  M[I,9] := M[I,9] + M[I,10];
  M[I,12] := M[I,12];
  M[I,13] := M[I,13];
  M[I,14] := M[I,14];
  M[I,15] := M[I,15];
  M[I,16] := M[I,16];
  M[I,17] := M[I,17];
  M[I,18] := M[I,18];
  M[I,19] := M[I,19];
  M[I,21] := M[I,22] + M[I,23];
  M[I,23] := M[I,25];
  M[I,24] := M[I,24];
  M[I,27] := M[I,30];
  M[I,28] := M[I,26] + M[I,27];
  M[I,29] := M[I,31];
  M[I,30] := M[I,32];
FOR J := 44 STEP 1 UNTIL 52 DO M[i, 38] := M[i, 38] + M[i, J];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 39] := M[i, 44];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 40] := M[i, 51];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 41] := M[i, 39] + M[i, 40];


FOR J := 44 STEP 1 UNTIL 52 DO M[i, 43] := M[i, 59] + M[i, 60] + M[i, 61];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 44] := M[i, 59];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 45] := M[i, 60];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 46] := M[i, 61];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 47] := M[i, 62] + M[i, 63];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 48] := M[i, 64];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 50] := M[i, 65];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 51] := M[i, 66] + M[i, 67];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 52] := M[i, 67];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 53] := M[i, 66];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 54] := M[i, 68];


FOR J := 44 STEP 1 UNTIL 52 DO M[i, 55] := M[i, 69];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 56] := M[i, 70] + M[i, 71];


FOR J := 44 STEP 1 UNTIL 52 DO M[i, 58] := M[i, 72];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 59] := M[i, 73] + M[i, 74];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 60] := M[i, 75];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 61] := M[i, 76];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 62] := M[i, 77];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 63] := M[i, 84];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 65] := M[i, 78] + M[i, 82];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 66] := M[i, 79] + M[i, 83];

FOR J := 44 STEP 1 UNTIL 52 DO M[i, 64] := M[i, 65] + M[i, 66];

END;

% Routine to print out aggregated interindustry employment matrix and row and column sums
% WRITE(LINE, "AGGREGATED INTERINDUSTRY EMPLOYMENT MATRIX");
WRITE(LINE, "(ROW AND COLUMN SUMS OVER SELF-CONTAINED ROWS)");
WRITE(LINE, HEAD2);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL LL DO BEGIN
  M[i, 1] := 0;
  FOR J := 1 STEP 1 UNTIL RK DO R[I, J] := R[I] + M[J, I];
  WRITE(LINE, F1, I, R[T(I)]);  END;
FOR SPECIALROWS DO TOTAL := TOTAL + R[J];
WRITE(LINE, SPACE 2, F2, TOTAL);
WRITE(LINE, HFAD3);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  L[I] := 0;
  WRITE(LINE, F3, LL, R[L]);  END;
FOR SPECIALCOLUMNS DO TOTAL := TOTAL + R[J];
WRITE(LINE, SPACE 2, F4, TOTAL);
WRITE(LINE, HFAD4);
TOTAL := 0;
FOR SPECIALROWS DO Y(I) := Y(I) + M(I,J);
WRITE(LINE,F1,I,Y(I));
TOTAL := TOTAL + Y(I); END;
WRITE(LINE(SKIP 1),F2,TOTAL);

% ROUTINE TO PRINT AGGREGATE INTERINDUSTRY EMPLOYMENT MATRIX
% IF FULLPRINT THEN BEGIN
WRITE(LINE,"AGGREGATE INTERINDUSTRY EMPLOYMENT MATRIX"); FOR K := 1 STEP 10 UNTIL 71 BEGIN
L := K + 9;
WRITE(LINE,HEA4,K,L);
FOR J := 1 STEP 1 UNTIL LL DO
WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M(I,J));
END;
WRITE(LINE,HEA4,81,KK);
FOR J := 1 STEP 1 UNTIL LL DO
WRITE(LINE,F7,J,FOR I := 81 STEP 1 UNTIL KK DO M(I,J));
WRITE(LINE(SKIP 1));
% AGGREGATE INTERINDUSTRY EMPLOYMENT TRANSPOSE
FOR K := 1 STEP 10 UNTIL 51 BEGIN
L := K + 9;
WRITE(LINE,HEA5,K,L);
FOR J := 1 STEP 1 UNTIL KK DO
WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M(J,I));
END;
WRITE(LINE,HEA5,61,LL);
FOR J := 1 STEP 1 UNTIL KK DO
WRITE(LINE,F7,J,FOR I := 61 STEP 1 UNTIL LL DO M(J,I));
WRITE(LINE(SKIP 1));
% FOR BRANCHES 2 AND 3 MODIFY THE M-MATRIX WITH MU
% IF BRANCH GTR 1 THEN BEGIN
WRITE(LINE,HEAD2);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL LL BEGIN
SUM := 0;
FOR J := 1 STEP 1 UNTIL KK BEGIN
M(J,I) := M(J,I) * MU(I);
SUM := SUM + M(J,I);
END;
WRITE(LINE,F1,I,SUM);
TOTAL := TOTAL + SUM;
END;
WRITE(LINE(SKIP 1),F2,TOTAL);
% ROUTINE TO PRINT MODIFIED M-MATRIX
% IF FULLPRINT THEN BEGIN
WRITE(LINE,"MODIFIED INTERINDUSTRY EMPLOYMENT MATRIX"); FOR K := 1 STEP 10 UNTIL 71 BEGIN
L := K + 9;
WRITE(LINE,HEA4,K,L);
FOR J := 1 STEP 1 UNTIL LL DO
WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M(I,J));
END;
```
WRITE(LINF, "H\tA\tD\t4\tH1\tKK");
FOR J := 1 STEP 1 UNTIL LL DO
  WRITE(LINE\#F7\#J, FOR I := 81 STEP 1 UNTIL KK DO M[I, J]);
WRITE(LINF[SKIP 11]);
WRITE(LINF, "TRANPOSE PRINT OF MODIFIED M-MATRIX");
FOR K := 1 STEP 10 UNTIL 51 DO BEGIN
  L := K + 9;
  WRITE(LINE\#HEAD5\#K\#L);
  FOR J := 1 STEP 1 UNTIL KK DO
    WRITE(LINE\#F10\#J, FOR I := K STEP 1 UNTIL L DO M[J, I]);
  WRITE(LINF[SKIP 11]);
  END;
END;

SELECT BRANCH FOR REMAINING PROCESSING
CASE BRANCH\# OF BEGIN:

\# BRANCH U N E

BEGIN BY USING THE M\# VECTOR TO MODIFY R

BEGIN SEG;
WRITE(LINE\#SPACE 2\#, "NEW R-VECTOR (MODIFIED BY M\#)\"");
WRITE(LINE\#HEAD1); FOR I := 1 STEP 1 UNTIL LL DO BEGIN
  R[I] := R[I] \# MULTI;
  WRITE(LINE\#F1\#I, R[I]);
END;
TOTAL := 0;
FOR SPECIAL ROWS DO TOTAL := TOTAL \# R[J];
WRITE(LINF[SKIP 11\#F2\#TOTAL]);
FORM S(ALPH) OVERLAYING THE B-MATRIX
FOR I := 1 STEP 1 UNTIL LL DO
  FOR J := 1 STEP 1 UNTIL MM DO R[I, J] := R[I] \# B[I, J];

RUN IF TO PRINT S(ALPH) AND ITS TRANSPOSE
IF FULLPRINT THEN BEGIN
WRITE(LINE\#SPACE 2\#, "S(ALPH) MATRIX");
FOR K := 1 STEP 10 UNTIL 171 DO BEGIN
  L := K + 9;
  WRITE(LINE\#HEAD4\#K\#L);
  FOR J := 1 STEP 1 UNTIL LL DO
    WRITE(LINE\#F10\#J, FOR I := K STEP 1 UNTIL L DO R[J, I]);
  WRITE(LINE\#HEAD4\#181\#MM);
  FOR J := 1 STEP 1 UNTIL LL DO
    WRITE(LINE\#F7\#J, FOR I := 181 STEP 1 UNTIL MM DO R[J, I]);
WRITE(LINE[SKIP 11]);
```
WRITE(LINE,""TRANPOSE PRINT OF S(\(\text{ALPHA}\))")
FOR K = 1 STEP 10 UNTIL 51 DO BEGIN
L = K + 9;
WRITE(LINE,HEADD5,K,L);
FOR J = 1 STEP 1 UNTIL MM DO
WRITE(LINE,F10,J,FOR I = K STEP 1 UNTIL L DO B[I,J]);
END;
WRITE(LINE,HEADD5,61,LL);
FOR J = 1 STEP 1 UNTIL MM DO
WRITE(LINE,F7,J,FOR I = 61 STEP 1 UNTIL LL DO B[I,J]);
WRITE(LINE(SKIP 1));
END;

\% \% CALCULATE ROWSUNS OVER SPECIAL COLUMNS, OVERLAYING R \%
WRITE(LINE,""GENERATED INDUSTRY EMPLOYMENT VECTOR")
WRITE(LINE,HEADD1);
FOR I = 1 STEP 1 UNTIL LL DO BEGIN
R[I] = 0;
FOR SPECIALCOLUMNS DO R[I] = R[I] + R[I,J];
WRITE(LINE,F1,I,R[I]);
END;
TOTAL = 0;
FOR SPECIALROWS DO TOTAL = TOTAL + R[J];
WRITE(LINE,SPACE 2),"TOTAL EMPLOYMENT = ",F17.5,TOTAL);
WRITE(LINE,""GENERATED OCCUPATIONAL EMPLOYMENT VECTOR")
WRITE(LINE,HEADD1);

\% \% TAKE COLUMNSUMS OVER SELECTED ROWS AND PRINT \%
FOR I = 1 STEP 1 UNTIL MM DO BEGIN
SH[I] = 0;
FOR SPECIALROWS DO SH[I] = SH[I] + B[J,I];
WRITE(LINE,F1,I,SH[I]);
END;
TOTAL = 0;
FOR SPECIALCOLUMNS DO TOTAL = TOTAL + SH[J];
WRITE(LINE,""TOTAL EMPLOYMENT = ",F17.5,TOTAL));
END OF BRANCH ONE;

- - - - -

\% \% MULTIPLY M * B TO GET S(BETA), OVERLAYING R. \%
\% HIT ONLY SPECIAL ROWS TO AVOID DOUBLE-COUNTING. \%
BEGIN SEG1
FOR I = 1 STEP 1 UNTIL KK DO FOR K = 1 STEP 1 UNTIL MM DO BEGIN
SB[I,K] = 0;
END;
\% \% ROUTINE TO PRINT S(BETA) AND TRANPOSE \%
IF FULLPRINT THEN BEGIN
WRITE(LINE,""***S(BETA) MATRIX")
FOR K = 1 STEP 10 UNTIL 171 DO BEGIN
L = K + 9;
WRITE(LINE,""***TRANSPOSE PRINT OF S(\(\text{ALPHA}\))")
FOR K = 1 STEP 10 UNTIL 51 DO BEGIN
L = K + 9;
WRITE(LINE,HEADD5,K,L);
FOR J = 1 STEP 1 UNTIL MM DO
WRITE(LINE,F10,J,FOR I = K STEP 1 UNTIL L DO B[I,J]);
END;
WRITE(LINE,HEADD5,61,LL);
FOR J = 1 STEP 1 UNTIL MM DO
WRITE(LINE,F7,J,FOR I = 61 STEP 1 UNTIL LL DO B[I,J]);
WRITE(LINE(SKIP 1));
END;
\begin{verbatim}
\textbf{R A N C H T H R E E}

\texttt{\textbf{CALCULATE VECTORS OF COLUMN SUMS AND PRINT}}
\texttt{\textbf{CALCULATE AND PRINT VECTORS OF ROW SUMS (OVER SPECIAL COLUMNS)}}

\texttt{\textbf{PULL OUT A COLUMN VECTOR FROM B, OVERWRITING R, AND PRINT IT}}
\end{verbatim}
% FORM S(H) MATRIX, OVERWRITING M
% FOR I = 1 STEP 1 UNTIL KK DO FOR J = 1 STEP 1 UNTIL LL DO
% M[I,J] := M[I,J] * R[J];
% Routines to print S(H)
% IF FULLPRINT THEN BEGIN
WRITE(LINE(SKIP 1));
WRITE(LINE("***S(H) MATRIX"));
FOR K = 1 STEP 10 UNTIL 51 DO BEGIN
L := K + 9;
WRITE(LINE(HEAD4,K,L));
FOR J = 1 STEP 1 UNTIL KK DO
WRITE(LINE(F10,J,FOR I = K STEP 1 UNTIL L DO M[J,I]));
END;
WRITE(LINE(HEAD4,61,LL));
FOR J = 1 STEP 1 UNTIL KK DO
WRITE(LINE(F7,J,FOR I = 61 STEP 1 UNTIL LL DO M[J,I]));
WRITE(LINE(SKIP 1));
WRITE(LINE("TRANSPOSE PRINT OF S(H)"));
FOR K = 1 STEP 10 UNTIL 71 DO BEGIN
L := K + 9;
WRITE(LINE(HEAD5,K,L));
FOR J = 1 STEP 1 UNTIL LL DO
WRITE(LINE(F10,J,FOR I = K STEP 1 UNTIL L DO M[T,J]));
END;
WRITE(LINE(HEAD5,81,KK));
FOR J = 1 STEP 1 UNTIL LL DO
WRITE(LINE(F7,J,FOR I = 81 STEP 1 UNTIL KK DO M[I,J]));
WRITE(LINE(SKIP 1));
END;
% CUMPUTE COLUMNSUMS OF S(H), OVERWRITING R* AND PRINT
% WRITE(LINE("EMPLOYMENT GENERATED IN: ");
% FOR I = 1 STEP 1 UNTIL LL DO BEGIN
R[I] := M[I,I];
FOR J = 2 STEP 1 UNTIL KK DO R[I] := R[I] + M[J,I];
WRITE(LINE(F1,I,R[I]));
END;
% SUM THE COLUMNSUMS FOR SPECIAL COLUMNS
% TOTAL I = 0;
% FOR SPECIALROWS DO TOTAL := TOTAL + R[J];
WRITE(LINE(SP=4,F2,TOTAL));
% CUMPUTE ROWSUMS OVER SPECIAL COLUMNS, PRINT, AND TOTAL
% WRITE(LINE("EMPLOYMENT GENERATED BY: ");
% TOTAL I = 0;
% FOR I = 1 STEP 1 UNTIL KK DO BEGIN
SUM I = 0;
FOR SPECIALROWS DO SUM := SUM + M[I,J];
WRITE(LINE(F1,I,SUM));
%
TOTAL = TOTAL + SUM; END;
WRITE(LINE='TOTAL');
END OF ARNACH THREE;
END;
FINISH; END ERGWORKS.
L=57;
FUR I=1 STEP 1 UNTIL L DO
READ(PI,/*FOR J=1 STEP 1 UNTIL 89 DO P[I,J]);
RE|IND(P1);
WRITE(LINE='"INDUSTRY ACTIVITY MATRIX"');
PHTMATEX(P*L,89,0);
WRITE(LINE='"TRANSPOSE OF INDUSTRY ACTIVITY MATRIX"');
PHTTRANS(P*L,89,0);
EUIWFILF(P*L,89);
RE|IND(P1);
FUR I=1 STEP 1 UNTIL L DO
READ(PI,/*FOR J=1 STEP 1 UNTIL 89 DO P[I,J]);
RE|IND(P1);
WRITE(LINE='"MODIFIED P MATRIX"');
PHTMATEX(P*L,89,0);
WRITE(LINE='"TRANSPOSE OF P MATRIX"');
PHTTRANS(P*L,89,0);
EUIWFILF(P*L,89);
RE|IND(P1);
FUR I=1 STEP 1 UNTIL L DO
READ(PI,/*FOR J=1 STEP 1 UNTIL 89 DO P[I,J]);
RE|IND(P1);
WRITE(LINE='"SHEU P MATRIX"');
PHTMATEX(P*L,89,0);
WRITE(LINE='"TRANSPOSE OF SHEU P MATRIX"');
PHTTRANS(P*L,89,0);
FUR I=1 STEP 1 UNTIL 85 DO
READ(M3,/*FOR J=1 STEP 1 UNTIL 85 DO M[J,I]);
RE|IND(M3);
WRITE(LINE='"M-MATRIX"');
PHTMATEX(M*85,85,0);
WRITE(LINE='"TRANSPOSE OF M-MATRIX"');
PHTTRANS(M*85,85,0);
EUIWFILF(M*85,85,0);
RE|IND(M3);
FUR I=1 STEP 1 UNTIL 85 DO
READ(M3,/*FOR J=1 STEP 1 UNTIL 85 DO M[J,I]);
RE|IND(M3);
WRITE(LINE='"MODIFIED M-MATRIX"');
PHTMATEX(M*85,85,0);
WRITE(LINE='"TRANSPOSE OF MODIFIED M-MATRIX"');
PHTTRANS(M*85,85,0);
EN|WORKS(L*89,85,66*165);
Appendix D

Sample Input and Output
<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>103851.0000</td>
</tr>
<tr>
<td>2</td>
<td>48161.0000</td>
</tr>
<tr>
<td>3</td>
<td>80999.0000</td>
</tr>
<tr>
<td>4</td>
<td>75496.0000</td>
</tr>
<tr>
<td>5</td>
<td>72994.0000</td>
</tr>
<tr>
<td>6</td>
<td>21283.0000</td>
</tr>
<tr>
<td>7</td>
<td>24001.0000</td>
</tr>
<tr>
<td>8</td>
<td>67269.0000</td>
</tr>
<tr>
<td>9</td>
<td>20663.0000</td>
</tr>
<tr>
<td>10</td>
<td>6402.0000</td>
</tr>
<tr>
<td>11</td>
<td>7083.0000</td>
</tr>
<tr>
<td>12</td>
<td>3547.0000</td>
</tr>
<tr>
<td>13</td>
<td>62604.0000</td>
</tr>
<tr>
<td>14</td>
<td>25794.0000</td>
</tr>
<tr>
<td>15</td>
<td>27444.0000</td>
</tr>
<tr>
<td>16</td>
<td>5906.0000</td>
</tr>
<tr>
<td>17</td>
<td>-6500.0000</td>
</tr>
<tr>
<td>18</td>
<td>4610.0000</td>
</tr>
<tr>
<td>19</td>
<td>3960.0000</td>
</tr>
<tr>
<td>20</td>
<td>796.0000</td>
</tr>
<tr>
<td>21</td>
<td>3034.0000</td>
</tr>
<tr>
<td>22</td>
<td>14176.0000</td>
</tr>
<tr>
<td>23</td>
<td>19391.0000</td>
</tr>
<tr>
<td>24</td>
<td>347.0000</td>
</tr>
<tr>
<td>25</td>
<td>9299.0000</td>
</tr>
<tr>
<td>26</td>
<td>3137.0000</td>
</tr>
<tr>
<td>27</td>
<td>2472.0000</td>
</tr>
<tr>
<td>28</td>
<td>0.0000</td>
</tr>
<tr>
<td>29</td>
<td>0.0000</td>
</tr>
<tr>
<td>30</td>
<td>1374.0000</td>
</tr>
<tr>
<td>31</td>
<td>371.0000</td>
</tr>
<tr>
<td>32</td>
<td>298.0000</td>
</tr>
<tr>
<td>33</td>
<td>41937.0000</td>
</tr>
<tr>
<td>34</td>
<td>706.0000</td>
</tr>
<tr>
<td>35</td>
<td>907.0000</td>
</tr>
<tr>
<td>36</td>
<td>4201.0000</td>
</tr>
<tr>
<td>37</td>
<td>173.0000</td>
</tr>
<tr>
<td>38</td>
<td>304.0000</td>
</tr>
<tr>
<td>39</td>
<td>467.0000</td>
</tr>
<tr>
<td>40</td>
<td>1544.0000</td>
</tr>
<tr>
<td>41</td>
<td>879.0000</td>
</tr>
<tr>
<td>42</td>
<td>1002.0000</td>
</tr>
<tr>
<td>43</td>
<td>2200.0000</td>
</tr>
<tr>
<td>44</td>
<td>3408.0000</td>
</tr>
<tr>
<td>45</td>
<td>2250.0000</td>
</tr>
<tr>
<td>46</td>
<td>434.0000</td>
</tr>
<tr>
<td>47</td>
<td>0.0000</td>
</tr>
<tr>
<td>48</td>
<td>1474.0000</td>
</tr>
<tr>
<td>49</td>
<td>50984.0000</td>
</tr>
<tr>
<td>50</td>
<td>32460.0000</td>
</tr>
<tr>
<td>51</td>
<td>9112.0000</td>
</tr>
<tr>
<td>52</td>
<td>5097.0000</td>
</tr>
</tbody>
</table>

**TOTAL** | **774599.0000**
## Row and Column Sums of Interindustry Employment Matrix

<table>
<thead>
<tr>
<th>Column Sum</th>
<th>Row Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>289249.77572</td>
</tr>
<tr>
<td>2</td>
<td>53753.977118</td>
</tr>
<tr>
<td>3</td>
<td>35981.67789</td>
</tr>
<tr>
<td>4</td>
<td>10166.31912</td>
</tr>
<tr>
<td>5</td>
<td>1164.87117</td>
</tr>
<tr>
<td>6</td>
<td>1912.45546</td>
</tr>
<tr>
<td>7</td>
<td>30317.41706</td>
</tr>
<tr>
<td>8</td>
<td>-5886.32801</td>
</tr>
<tr>
<td>9</td>
<td>85487.19556</td>
</tr>
<tr>
<td>10</td>
<td>1289.42814</td>
</tr>
<tr>
<td>11</td>
<td>1965794.72206</td>
</tr>
<tr>
<td>12</td>
<td>374313.86243</td>
</tr>
<tr>
<td>13</td>
<td>508285.99495</td>
</tr>
<tr>
<td>14</td>
<td>5323077.88461</td>
</tr>
<tr>
<td>15</td>
<td>325388.33854</td>
</tr>
<tr>
<td>16</td>
<td>107572.60661</td>
</tr>
<tr>
<td>17</td>
<td>131263.56008</td>
</tr>
<tr>
<td>18</td>
<td>25640794.45716</td>
</tr>
<tr>
<td>19</td>
<td>2776975.3723</td>
</tr>
<tr>
<td>20</td>
<td>570368.05696</td>
</tr>
<tr>
<td>21</td>
<td>1091.82942</td>
</tr>
<tr>
<td>22</td>
<td>595306.97073</td>
</tr>
<tr>
<td>23</td>
<td>275403.17227</td>
</tr>
<tr>
<td>24</td>
<td>202781.84626</td>
</tr>
<tr>
<td>25</td>
<td>16843.36638</td>
</tr>
<tr>
<td>26</td>
<td>494342.14611</td>
</tr>
<tr>
<td>27</td>
<td>161161.03176</td>
</tr>
<tr>
<td>28</td>
<td>11464.323</td>
</tr>
<tr>
<td>29</td>
<td>631403.32544</td>
</tr>
<tr>
<td>30</td>
<td>301036.64342</td>
</tr>
<tr>
<td>31</td>
<td>651283.30303</td>
</tr>
<tr>
<td>32</td>
<td>286463.25607</td>
</tr>
<tr>
<td>33</td>
<td>28549643</td>
</tr>
<tr>
<td>34</td>
<td>475570.09705</td>
</tr>
<tr>
<td>35</td>
<td>502681.48864</td>
</tr>
<tr>
<td>36</td>
<td>534949.52473</td>
</tr>
<tr>
<td>37</td>
<td>172235.51539</td>
</tr>
<tr>
<td>38</td>
<td>108967.18843</td>
</tr>
<tr>
<td>39</td>
<td>77514.6235</td>
</tr>
<tr>
<td>40</td>
<td>952897.71494</td>
</tr>
<tr>
<td>41</td>
<td>786960.4070</td>
</tr>
<tr>
<td>42</td>
<td>2379674.7212</td>
</tr>
<tr>
<td>43</td>
<td>1185750.10398</td>
</tr>
<tr>
<td>44</td>
<td>175247.30404</td>
</tr>
<tr>
<td>45</td>
<td>2793966.60373</td>
</tr>
<tr>
<td>46</td>
<td>168259.4792</td>
</tr>
<tr>
<td>47</td>
<td>2877593.37409</td>
</tr>
<tr>
<td>48</td>
<td>3066004.7697</td>
</tr>
<tr>
<td>49</td>
<td>2527133.5514</td>
</tr>
<tr>
<td>50</td>
<td>2179075311</td>
</tr>
<tr>
<td>51</td>
<td>5272871.0543</td>
</tr>
<tr>
<td>52</td>
<td>28940361609</td>
</tr>
<tr>
<td>53</td>
<td>41972311205</td>
</tr>
<tr>
<td>54</td>
<td>47762805799</td>
</tr>
<tr>
<td>55</td>
<td>211199.49912</td>
</tr>
<tr>
<td>56</td>
<td>92162675039</td>
</tr>
<tr>
<td>57</td>
<td>10456496353</td>
</tr>
<tr>
<td>58</td>
<td>9001042261</td>
</tr>
<tr>
<td>59</td>
<td>25737685533</td>
</tr>
<tr>
<td>60</td>
<td>104567912975</td>
</tr>
<tr>
<td>61</td>
<td>5279226251</td>
</tr>
<tr>
<td>62</td>
<td>2946622764</td>
</tr>
<tr>
<td>63</td>
<td>2367728627</td>
</tr>
<tr>
<td>64</td>
<td>5527094274</td>
</tr>
<tr>
<td>65</td>
<td>164477458768</td>
</tr>
<tr>
<td>66</td>
<td>61063580489</td>
</tr>
<tr>
<td>67</td>
<td>327132644</td>
</tr>
<tr>
<td>68</td>
<td>100211642013</td>
</tr>
<tr>
<td>69</td>
<td>162449973886</td>
</tr>
<tr>
<td>70</td>
<td>24953568854</td>
</tr>
<tr>
<td>71</td>
<td>24456615823</td>
</tr>
<tr>
<td>72</td>
<td>319268644809</td>
</tr>
<tr>
<td>73</td>
<td>10127766483</td>
</tr>
<tr>
<td>74</td>
<td>950405720</td>
</tr>
<tr>
<td>75</td>
<td>82803142501</td>
</tr>
<tr>
<td>76</td>
<td>7481933227</td>
</tr>
<tr>
<td>77</td>
<td>764022148395</td>
</tr>
<tr>
<td>78</td>
<td>24970460015</td>
</tr>
<tr>
<td>79</td>
<td>963155464</td>
</tr>
<tr>
<td>80</td>
<td>3503041712</td>
</tr>
<tr>
<td>81</td>
<td>3428615020</td>
</tr>
<tr>
<td>82</td>
<td>2023994370</td>
</tr>
<tr>
<td>83</td>
<td>10966549253</td>
</tr>
<tr>
<td>84</td>
<td>254741243765</td>
</tr>
<tr>
<td>85</td>
<td>2782975352</td>
</tr>
<tr>
<td>86</td>
<td>2782975352</td>
</tr>
<tr>
<td>TOTAL</td>
<td>840325941662</td>
</tr>
</tbody>
</table>

**Row Sum:** 168925497548

**Column Sum:** 55
<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.233482, 0.02488</td>
</tr>
<tr>
<td>2</td>
<td>0.250919, 0.0629</td>
</tr>
<tr>
<td>3</td>
<td>0.212944, 0.07713</td>
</tr>
<tr>
<td>4</td>
<td>0.63477, 0.04391</td>
</tr>
<tr>
<td>5</td>
<td>5.6776, 0.77111</td>
</tr>
<tr>
<td>6</td>
<td>1.25179, 0.1265</td>
</tr>
<tr>
<td>7</td>
<td>7.49817, 0.04974</td>
</tr>
<tr>
<td>8</td>
<td>2.47253, 0.21681</td>
</tr>
<tr>
<td>9</td>
<td>9.42534, 0.0328</td>
</tr>
<tr>
<td>10</td>
<td>9.50563, 0.2717</td>
</tr>
<tr>
<td>11</td>
<td>5.61218, 0.01386</td>
</tr>
<tr>
<td>12</td>
<td>5.14354, 0.07182</td>
</tr>
<tr>
<td>13</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>14</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>15</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>16</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>17</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>18</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>19</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>20</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>21</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>22</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>23</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>24</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>25</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>26</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>27</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>28</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>29</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>30</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>31</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>32</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>33</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>34</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>35</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>36</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>37</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>38</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>39</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>40</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>41</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>42</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>43</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>44</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>45</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>46</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>47</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>48</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>49</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>50</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>51</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>52</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>53</td>
<td>5.16394, 0.05826</td>
</tr>
<tr>
<td>54</td>
<td>5.16394, 0.05826</td>
</tr>
</tbody>
</table>

**TOTAL EMPLOYMENT** = 0.094317, 49
This paper presents the preliminary documentation and user's guide for the Center for Advanced Computation economic and manpower forecasting model. Section I gives introductory and background information on the development of the model and presents a brief but rigorous theoretical basis for the on-line system. Section II gives a description of the basic MANPOWER/DEMAND program indicating the function of the program, the detailed workings of the system options, and the language in which it is written. Appendices contain specifications of the data tapes and disc files involved, flow charts of the computer processes, and sample data input and output.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
<tr>
<td>Applications</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social and Behavioral Sciences</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Economics</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>