The Craft of Economic Modeling

Part 3. Multisectoral Models

Contents

Chapter 11. Input-Output Analysis in the Ideal case
Chapter 12. Balancing matrices -- the RAS method
Chapter 13. Trade and Transportation Margins and Indirect Taxes
Chapter 14. Making Product-to-Product Tables
Chapter 15. Modeling Personal Consumption Expenditure

Projected chapters

16. Production functions and investment
17. Interdyme software for building multisector models
18. Linking national models via international trade
CHAPTER 11

Input-Output in the Ideal Case

1. Input-Output Flow Tables

Multisectoral models begin from an accounting of the flows of goods and services among various industries of the economy. Table 1 shows a simple interindustry accounting, or input-output flow table, for an imaginary but not unrealistic eight-sector economy. The selling industries are listed down the left side of the table. The last, abbreviated to "GovInd," is "Government Industry", a fictitious industry which in this table simply supplies the government with the services of its own employees. Below these come the classes of factor payments, here Depreciation, Labor compensation, Capital income (such as interest, profits, rents, or proprietor income), and Indirect taxes (such as property taxes, sales taxes, and excise taxes as on alcohol, tobacco, and gasoline). Note the similarity of these categories of factor payments to the categories of national income. Their sum is the row Value added. Across the top of the table the same eight industries are listed as buyers of products. Here they are followed by columns corresponding to the principal divisions of the "product side" of the national accounts, namely

- Con Personal consumption expenditure
- Gov Government purchases of goods and services
- Inv Investment
- Exp Exports
- Imp Imports (as negative numbers)

In input-output terms, these are the final demand columns. The next-to-last column, labeled FD for "Final Demand," shows their sum. It is shaded to emphasized that it is derived by summing other columns. The next last column, also shaded, is the sum of all the (non-shaded) elements row.

Across each row of the table are shown the sales of that industry to each of the industries and final demand columns. Thus, the 100 in the Agriculture row and Manufacturing (Mfg) column means that Agriculture sold 100 billion dollars (bd) of products to Manufacturing in the year covered by this table. Typical sales here are grains to milling, live animals to meat packing, or fruits and vegetables to plants which can or freeze them. The 15 in the Personal consumption (Con) column of the same row means that Agriculture sold 15 bd of products directly to households during the year. These sales are primarily fresh fruits and vegetables and eggs. In the table shown here, which is said to be in producer prices, they are recorded at the price the farmer received for them. These products are not necessarily bought at the farm gate, however, for going through wholesale and retail trade channels does not change the industry of origin of a product; going through a manufacturing process does. Thus, an orange sold as an orange to she who eats it appears as a sale from Agriculture to Personal consumption, despite the fact that it went through a store. Another orange that was turned into frozen orange juice appears first as a sale from Agriculture to Manufacturing at the price received by the farmer. It then reappears as a sale from Manufacturing to Personal consumption at the manufacturer's price. But the price paid by the ultimate consumer is neither the price received by farmer in the first case nor by the manufacturer in the second. Where is the difference, the
Table 1. An Input-Output Flow Table

<table>
<thead>
<tr>
<th>Buyer</th>
<th>Agriculture</th>
<th>Mining</th>
<th>Gas &amp; Electric</th>
<th>Mfg</th>
<th>Commerce</th>
<th>Transport</th>
<th>Services</th>
<th>Gov</th>
<th>Ind</th>
<th>COLSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agriculture</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>15</td>
<td>40</td>
</tr>
<tr>
<td>Mining</td>
<td>4</td>
<td>3</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Gas &amp; Electric</td>
<td>6</td>
<td>4</td>
<td>10</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>25</td>
<td>0</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>Mfg</td>
<td>20</td>
<td>10</td>
<td>4</td>
<td>60</td>
<td>25</td>
<td>18</td>
<td>20</td>
<td>0</td>
<td>400</td>
<td>200</td>
</tr>
<tr>
<td>Commerce</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>350</td>
<td>6</td>
</tr>
<tr>
<td>Transport</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>17</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>130</td>
<td>20</td>
</tr>
<tr>
<td>Services</td>
<td>6</td>
<td>3</td>
<td>8</td>
<td>45</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td>0</td>
<td>500</td>
<td>40</td>
</tr>
<tr>
<td>Gov/Ind</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
</tr>
</tbody>
</table>

| Intermediate | 60          | 23     | 48             | 287 | 77       | 39        | 80       | 0   | 614 |
| Deprec.      | 11          | 5      | 60             | 130 | 35       | 40        | 25       | 0   | 306 |
| Labor        | 65          | 20     | 21             | 260 | 140      | 97        | 485      | 150 | 1238|
| Capital      | 20          | 2      | 56             | 60  | 40       | 12        | 59       | 0   | 249 |
| Indirect tax | 8           | 0      | 20             | 50  | 109      | 10        | 18       | 0   | 215 |

Value added: 104 27 157 500 324 159 587 150

ColSum: 164 50 205 787 401 198 667 150

FD Sum: 36 164
commercial margin? In this table, it is in the sales of Commerce to Personal consumption expenditure. Transportation margins are handled similarly. Tables made with this pricing convention are said to be "in producer prices". We shall look at other ways of handling the problem of margins in Chapter 2.

As we look down the column for an industry, we see all the products which it needs for making its own. In the Agriculture column, we see first of all 20 bd from Agriculture itself. These are sales primarily of feed grains to animal husbandry, but include also sales of seed, hay, manure, and other products. These sales within the industry are common and are referred to in input-output jargon as "diagonals" because they appear on the main diagonal of the table. Further down the Agriculture column we see 4 bd for Mining, primarily crushed limestone, but also some coal. The 20 bd spent on Manufacturing bought gasoline, fertilizers, and pesticides. The 2 bd spent on Commerce were trade margins on these manufactured products. The 2 bd spent on Transport included transportation margins on the products of the other industries as well as costs incurred by the farmer in getting products to market. The purchases from Services includes the services of veterinarians, lawyers, and accountants. All the purchases of the industries from each other are called "intermediate" purchases because they do not go directly to the final user but are "mediated" by other industries. The sum of the intermediate purchases by each industry are in the row labeled "Intermediate" and shaded, as before, to show that it is derived by adding other entries in the table. Many tables also have a total intermediate column; our table omits it for the simple reason that it would not fit on the page.

Below the "Intermediate row" are the value-added rows. We find that Depreciation of equipment came to 11 bd. Labor received 65 bd. (In our imaginary economy, we imagine that proprietor income has been divided between labor and capital income. In most actual tables, it will be shown separately or classified as capital income.) The 20 bd of capital income includes interest payments, corporate profits, and capital's portion of proprietor income. The 8 bd of Indirect taxes is mostly property taxes.

Now precisely because the Capital income row of value added -- which includes both corporate profits and proprietor income -- is the total of sales minus the total of expenses, the column sum for each industry is equal to its row sum. For example, the row sum of Agriculture is 164 and the column sum (of the unshaded entries) is 164, and so on for all eight industries. This fact has a remarkable consequence which is the cornerstone of national accounting, namely that the sum of all the value-added entries is equal to the sum of all the final demand entries. In our table, each of these groups of entries is surrounded by a double line and each adds to 2008. Why is the total the same? Since the sum of each of the eight industry rows, say R, is equal to the sum of the corresponding column, the sum of all eight rows, 2622, is equal to the sum of all eight columns, say C, which is also 2622. Thus we have with R = C. But the total of the final demands, D, is R minus the total of the intermediate flows, say X, or D = R - X. Likewise, the total value added, V, is C, the sum of all the industry columns, less the sum of that part of them which is intermediate, or V = C - X. But R = C implies that R - X = C - X or D = V. Naturally, this D or V has a name, and that name is Gross Domestic Product. We have thus proved the fundamental identity of national accounting: Gross Domestic Product (GDP) is the same whether measured by the products that go to final demand or
by the income which goes to factors. In our table, this identity appears in the fact that the sum of the FD column, 2008, is the sum of the Value added row, also 2008, which is the GDP of this economy. Arrayed in format of national accounts, our economy would appear as in Table 2.

### Table 2. The Income and Product Account

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal Consumption</td>
<td>1477</td>
<td>- Depreciation</td>
<td>306</td>
</tr>
<tr>
<td>Investment</td>
<td>224</td>
<td>= Net domestic product</td>
<td>1702</td>
</tr>
<tr>
<td>Exports</td>
<td>215</td>
<td>- Indirect taxes</td>
<td>215</td>
</tr>
<tr>
<td>Imports</td>
<td>-220</td>
<td>= National income</td>
<td>1487</td>
</tr>
<tr>
<td>Government purchases</td>
<td>312</td>
<td>Labor income</td>
<td>1238</td>
</tr>
<tr>
<td>Capital income</td>
<td>249</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Before leaving Table 1, we must make a fundamental point about it. With one small exception, the table makes sense in physical units. We can measure the output of Agriculture in bushels, that of Mining in tons, that of Gas and Electricity in BTU's, Transport in ton-miles, Labor in worker hours, Capital income in ounces of gold, and so on. Detailed tables in physical terms have in fact been made for China. Wassily Leontief, maker of the first input-output table, used to often insist in seminars that any calculations had to make sense in physical terms.

The small exception, however, is important: the column sums of a table in physical terms are utterly meaningless since all the elements are in different units. Naturally, the row totals -- which are meaningful -- do not equal the meaningless totals of the corresponding columns. This point would seem so obvious as to be not worth making were it not for the fact that it is often forgotten, precisely by the makers of input-output tables. For if a table is made in the prices of some year other than the year to which it refers, it is essentially in physical units. Thus, we can make a table for 1995 in 1980 prices, where the physical measure in each row is "one 1980 dollar's worth" of the product. In other words, the physical unit for each product is how much of it one dollar would buy in 1980. For any product for which a price index can be made, 1995 dollar amounts can be converted into 1980-dollar physical units by the price index. For Capital income, there is no very natural unit, but one could take a market basket of things a "capitalist" is likely to buy with his income, make a price index for it, and deflate capital income by this index. One can have in this way a perfectly sensible, meaningful table. But its column sums are meaningless and certainly do not equal the row sums.

Unfortunately, some table makers have disregarded this obvious fact and have simply forced the value added in each industry of such a table to equal the difference between the row sum of the industry and the sum of the intermediate inputs into it. This practice is called "double deflation" because first the outputs are deflated and then the purchased inputs deflated and subtracted from the deflated output to obtain a measure of "constant-price value added". The results make just as much sense as saying that five squirrels minus three elephants equals two lions. The arithmetic is right but the units are crazy. In such a table, the deflators for, say, labor income would be different in different industries. They might well be negative. Take book publishing for example. The computer services which go into book writing and production in 1995, measured in physical units and priced in 1980 prices, would have cost millions, possibly billions of dollars per book in 1980 and totalled far more than the value of the output of book publishing in 1995 deflated back to 1980 prices. Thus, 1995 value added in 1980 prices in book publishing would have to be negative in such a table.
Such a nonsensical result is simply the natural consequence of the nonsensical practice of "double deflation" to make each column total equal the corresponding row total in a table in physical units. Even the distinguished French statistical office, INSEE, which has an outstanding record for publishing input-output tables in current prices on a timely schedule, tarnishes its reputation by also publishing a series of tables from 1980 to 1994 in prices of 1980 with row totals equal to the corresponding column totals. The intermediate and final demand portions of these tables may be most valuable. But the value-added portion is nonsense. The nonsense is compounded by the fact that these procedures are sanctioned by international statistical standards, and many statistical offices engage in them. Economists have made matters worse by taking these nonsensical measures of "real" value added as measures of "real" product in studies of productivity. They are supposed to deal with the problem that simple ratios of the output of an industry to its primary inputs may be affected by the industry switching to purchasing some intermediate components instead of making them itself. They clearly do not solve the problem satisfactorily and should simply be retired from all statistical publications.


An input-flow table describes an economy in a particular year. Its greatest value, however, lies in the ability it gives us to answer the question What would the outputs, value added, and intermediate flows have been had the final demands been different? To answer that question in the simplest possible way, we must assume that the ratio of each input into an industry to that industry's output remains constant when the final demands are changed. These ratios are known as the "input-output coefficients," and may be defined by

\[ a_{ij} = \frac{x_{ij}}{q_j} \]

where \( x_{ij} \) is the flow from industry \( i \) to industry \( j \) in Table 1.1 and \( q_j \) is the output of industry \( j \), that is, it is the sum of row \( j \) or column \( j \) in the same table. For example,

\[ a_{1,4} = \frac{100}{787} = 0.12706 \]

Table 3 shows the complete matrix of these input-output coefficients corresponding to Table 1.

If we are willing to suppose that these coefficients remain constant as the final demand vector changes, then for any vector of final demands, \( f \), we can calculate the vector of industry outputs, \( q \), from the equation

\[ q = Aq + f \]  \hspace{1cm} (1.2.1)\]

where \( A \) is the matrix of input-output coefficients in Table 3. If we happen to choose as \( f \) the column vector of final demands in Table 1, (the first eight elements of the FD column: (36,3,90, ..., 150)'), then \( q \) should be the column vector of industry outputs of Table 1 (the vector of row sums of the eight industry rows: (164,50,205,....,150)'). For other values of \( f \), of course, we will find other values of \( q \).
# Table 3. Input-Output Coefficients

<table>
<thead>
<tr>
<th></th>
<th>Agric</th>
<th>Mining</th>
<th>Gas&amp;Elec</th>
<th>Mfg</th>
<th>Com</th>
<th>Trans</th>
<th>Serv</th>
<th>GovInd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agriculture</td>
<td>0.12195</td>
<td>0.02000</td>
<td>0.00000</td>
<td>0.12706</td>
<td>0.01247</td>
<td>0.00000</td>
<td>0.00300</td>
<td>0.00000</td>
</tr>
<tr>
<td>Mining</td>
<td>0.02439</td>
<td>0.06000</td>
<td>0.09756</td>
<td>0.01906</td>
<td>0.00499</td>
<td>0.00505</td>
<td>0.00300</td>
<td>0.00000</td>
</tr>
<tr>
<td>Electricity</td>
<td>0.03659</td>
<td>0.08000</td>
<td>0.04878</td>
<td>0.05083</td>
<td>0.04988</td>
<td>0.05051</td>
<td>0.03748</td>
<td>0.00000</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>0.12195</td>
<td>0.20000</td>
<td>0.01951</td>
<td>0.07624</td>
<td>0.06234</td>
<td>0.09091</td>
<td>0.02999</td>
<td>0.00000</td>
</tr>
<tr>
<td>Commerce</td>
<td>0.01220</td>
<td>0.02000</td>
<td>0.00488</td>
<td>0.01271</td>
<td>0.00499</td>
<td>0.01515</td>
<td>0.00900</td>
<td>0.00000</td>
</tr>
<tr>
<td>Transportation</td>
<td>0.01220</td>
<td>0.02000</td>
<td>0.02439</td>
<td>0.02160</td>
<td>0.00748</td>
<td>0.01010</td>
<td>0.00750</td>
<td>0.00000</td>
</tr>
<tr>
<td>Services</td>
<td>0.03659</td>
<td>0.06000</td>
<td>0.03902</td>
<td>0.05718</td>
<td>0.04988</td>
<td>0.02525</td>
<td>0.02999</td>
<td>0.00000</td>
</tr>
<tr>
<td>GovInd</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>
One way of solving (1.2.1) is to rewrite it as

\[ (I - A)q = f \]

or

\[ q = (I - A)^{-1}f. \]

The matrix of \((I - A)^{-1}\) on the right of this equation is known as the Leontief inverse of the A matrix. For our example, it is shown in Table 4. Its elements have a simple meaning. Element \((i,j)\) shows how much of product \(i\) must be produced in order to produce one unit of final demand for product \(j\). This interpretation is readily justified by taking \(f\) to be a vector of zeroes except for a 1 in row \(i\). Then \(q\) will be the \(i\)th column of \((I - A)^{-1}\), and its \(j\)th element will show exactly how much of product \(j\) will have to be produced in order to supply exactly one unit of \(i\) to final demand. In our example, in order to supply one unit of Agricultural product to final demand, .1691 units of Manufacturing must be produced. Note that, in the example, all elements of the Leontief inverse are non-negative. In view of the economic interpretation, that result is hardly surprising. Later in this chapter, we will show mathematically that the Leontief inverse from an observed \(A\) matrix is always non-negative.

<table>
<thead>
<tr>
<th>Table 4. The Leontief Inverse ((I - A)^{-1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1647 0.0620 0.0107 0.1634 0.0263 0.0165 0.0096 0.0000</td>
</tr>
<tr>
<td>0.0405 1.0830 0.1126 0.0352 0.0144 0.0150 0.0092 0.0000</td>
</tr>
<tr>
<td>0.0617 0.1137 1.0683 0.0748 0.0623 0.0641 0.0452 0.0000</td>
</tr>
<tr>
<td>0.1691 0.2530 0.0538 1.1201 0.0791 0.1091 0.0396 0.0000</td>
</tr>
<tr>
<td>0.0184 0.0276 0.0093 0.0185 1.0077 0.0180 0.0106 0.0000</td>
</tr>
<tr>
<td>0.0210 0.0319 0.0304 0.0297 0.0120 1.0151 0.0102 0.0000</td>
</tr>
<tr>
<td>0.0604 0.0911 0.0548 0.0791 0.0612 0.0379 1.0368 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000</td>
</tr>
</tbody>
</table>

We may also ask how much of a primary resource, such as Labor or Capital, would be needed for the production of a given final demand. We may define the resource coefficients similarly to the input-output coefficients by

\[ r_{ij} = y_{ij}/q_j \]

where \(y_{ij}\) is the payment to factor \(i\) by industry \(j\). For example, from Table 1, \(y_{2,4}\), the payment to resource 2, Labor, by industry 4, Manufacturing, is 360. If we denote by \(R\) the matrix of the \(r_{ij}\), then the vector of total payments to each resource for an output vector \(q\) is \(Rq\), and for a final demand vector, \(f\), it is \(R(I - A)^{-1}f\).

If we now think of each row of this matrix as a row vector and sum these vectors -- a process which makes sense if all the rows are measured in monetary values in the prices of the year of the table -- we get a row vector, \(v\), of value-added per unit of output. Just as previously we asked how outputs, \(q\), would change if \(f\) changed while \(A\) remains constant, we can now ask how prices, \(p\), would change if \(v\) changed while \(A\) remains constant. The row vector \(p\) must satisfy the equations

\[(1.2.2) \quad p = pA + v.\]
These equations state simply that the price of a unit of each product is equal to the cost of all products used in producing that unit (the first term on the right) plus value-added per unit produced. Just as the equations (1.2.1) provide the fundamental connection in multisectoral models between final demands and outputs, so these equations provide the fundamental connection between unit value added and prices. If we want to know how specific changes in productivity or in wages in one or several industries will affect prices in all industries, these equations are the key. If we calculate the prices for \( v \) vector given in the table, we should find that all prices are equal to 1.

There is, furthermore, a relation of fundamental importance between the solutions of the two sets of equations. Namely, given any \( A, f, \) and \( v \), the \( q \) and \( p \) which satisfy \( q = Aq + f \) and \( p = pA + v \) also satisfy

\[
(1.2.3) \quad vq = pf.
\]

This equation says that the value of the final demands evaluated at the prices implied by equations (1.2.2) are equal to the payments to the resources necessary to produce those final demands by (1.2.1). Thus, if our outputs and prices satisfy the required equations, we can be certain that GDP measured by the final demands in current prices will be equal to the GDP measured by the payments to resources (or factors) in current prices. If we build these equations into our models, we can be certain that the models will satisfy the basic accounting identity in current prices. This relation may well be called the fundamental theorem of input-output analysis. Fortunately, it is as easy to prove as it is important, and you should produce your own proof. If you need help desperately, look in the upside-down footnote. \(^1\)

3. Introduction to Input-Output Calculations

The simple calculations described in the previous section can be done with many programs -- spreadsheets, matrix packages, and even statistical packages. We want to introduce the reader to yet another tool, however, for it is one which will serve us well through the most complicated problems of dynamic multisectoral modeling. We shall see in due course how it can be combined with the G program familiar from the previous volumes to combine econometric results with matrix calculations.

This tool is a C++ library of matrix and vector definitions called the Beginner's Understandable Matrix Package, or BUMP. Some knowledge of DOS or command line OS2 and some acquaintance with C but not with C++ is necessary to use it. Also, it is necessary to have a C++ compiler on your computer. BUMP gives you very easy ways to read in matrices, perform common calculations, and display or write the results. It also
gives the full power of the C or C++ languages to do things with your matrices beyond the basics provided by BUMP.

The main content of BUMP resides in two files. The larger is bump.cpp which contains the C++ programs called when BUMP is used. The smaller is the bump.h file, a "header" file to be "included" in programs which use the BUMP routines. It describes the programs in bump.cpp sufficiently for the C++ compiler to recognize and call them properly. Other files provide for compiling and linking these programs with a user-written program employing the BUMP structures and functions.

The best way to learn to use BUMP is probably to jump right into the calculations described in the previous section. Put the craft3.zip file, the software for this volume, in a directory, say, craft3, and unzip it with the command "pkunzip craft3.zip". The bump.mak and bump.res fileassume that one is using the Borland C++ located in c:\bc45. If you are using a different version of the compiler Borland compiler located elsewhere, you will have to modify these two files, replacing each occurrence of "c:\bc45" with the location of your compiler. If you are using a different compiler, you will have to figure out the right options for your compiler. Otherwise, the program is rather generic C++ and should work with any compiler. By using the "dir" command, you will note three files of data, flows.dat, fd.dat, and va.dat. These contain the non-shaded data of Table 1 for the intermediate flows, the final demands, and the value-added quadrants respectively. Our first task is to compute the q, f, and v vectors, the input-output coefficient matrix, the Leontief inverse, and to verify that the solution of (1.2.1) is indeed q, while that of (1.2.2) is a vector of 1's.

The program for doing these calculations is in the file ch1s3.cpp, which is also shown in the box labeled ch1s3.cpp (ch1s3 = chapter 1, section 3 in the numbering we shall use for all programs in this volume). To use it, just type at the DOS or OS2 prompt the lines

```
bump ch1s3
ch1s3
```

The first compiles and links the program with the BUMP utilities; the second runs the program.

To understand how to use BUMP, we shall study the ch1s3.cpp file. It begins with the statements

```c
#include <stdio.h>
#include "bump.h"
```

The first is a statement that appears in most C programs to inform the compiler that we will be using certain standard input and output routines (in the computer sense of input and output). The second informs the compiler about the capabilities BUMP. Without this file, very few of the statements in this program would make any sense to the compiler.

The next statement,

```c
void main(){
```

indicates the beginning of a main program in C or C++. The "void" means that it returns no value to the operating system when it exits. The following { together with the } at the end delimit the block of code which forms this main program.
#include <stdio.h> // for printf();
#include "bump.h"

void main(){
    // Declare integers.
    int i, m, n;
    // Declare input matrices.
    Matrix AF("flows.dat");
    AF.Display("Here is the flow matrix:",5,0);
    Matrix FD("fd.dat"),Y("va.dat");
    tap(); //Display message "Tap a key to continue" and wait.
    n = AF.rows();
    m = Y.rows();
    // Declare other matrices and vectors
    Matrix AC(n,n),IMA(n,n),LINV(n,n),R(m,n);
    Vector q(n),f(n),s(n),p(1,n),v(1,n);
    // Form f as the row sum of the FD matrix.
    f = rowsum(FD);
    // Form q as the rowsum of the intermediate flow matrix plus f.
    q = rowsum(AF) + f;
    // Display f and q.
    f.Display("Here is the f vector:",5,0);
    q.Display("and the q vector.",5,0); tap();
    // Form the coefficient matrix.
    AC = AF%q;
    AC.Display("The coefficient matrix",6,4);
    // Form the IMA = I - A matrix
    IMA = -1.*AC;
    for (i = 1; i <= n; i++)
        IMA(i,i) += 1.;
    // Invert IMA.
    LINV = !IMA;
    LINV.Display("The Leontief Inverse Matrix", 6,4);
    // Write out the Leontief Inverse to the file linv.dat.
    writemat(LINV,"linv.dat", 6,4); tap();
    // Compute outputs and display them.
    s = LINV*f;
    s.Display("The computed outputs.",10,4);
    //Compute the R matrix
    Y.Display("The Y matrix.",6,0);
    R = Y*q;
    R.Display("The R matrix.",6,4); tap();
    // and take its column sums to get v matrix.
    v = colsum(R);
    v.Display("The v vector",6,4);
    // Compute prices
    p = v*LINV;
    p.Display("The computed prices should all be 1.0 .",6,3); tap();
    printf("\nEnd of calculations.\n");
}
The line

    // Declare integers.

is a comment. In C++, anything which follows // on the same line is a comment. The next line

    short i, m, n;

declares i, m, and n to be short integers. They will be used for dimensions of matrices. Unlike Basic or Fortran, C and C++ require that every variable should be declared and its type specified before it is used. This apparent rigidity in fact catches many errors before they cause any difficulty. The short integer occupies two bytes of storage and can accommodate integers between -32768 and +32767, a range more than adequate for the dimensions of the matrices we are likely to use.

The statements

    // Declare input matrices.
    Matrix AF("flows.dat");

is partly comparable to the previous declaration of integers in that the compiler will now recognize AF as the name of a Matrix. But, thanks to the programs in bump.cpp, it also causes the dimensions and values of this matrix to be read from the file "flows.dat". For our case, the flows.dat file is shown in the box labeled “The AF Intermediate Flows Matrix.”

```
The AF Intermediate Flows Matrix.

<table>
<thead>
<tr>
<th>Agriculture</th>
<th>Agricul</th>
<th>Mining</th>
<th>Elect</th>
<th>Mfg</th>
<th>Commerce</th>
<th>Transp</th>
<th>Services</th>
<th>Govt</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>10</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>25</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>4</td>
<td>60</td>
<td>25</td>
<td>18</td>
<td>20</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>17</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>8</td>
<td>45</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
```

The first line of this file begins with free form text information. It must not contain a digit preceded by a blank, for that signals the beginning of data on the line. (Thus, A1 could appear in this text but not 1A.) Then comes the year to which the data in the matrix refer. This year will be important later as we move into dynamic modeling, but for the moment it is ignored. If there is no year to which the data refer, you can put a 0 here. Then come four integers which are the numbers of the first and last rows of this matrix followed by the first and last columns. (The numbering of rows and columns does not have to begin with 1, although that is the default value.) Last on the first line comes the number of spaces to allow at the beginning of each data line for the title. Then come the data lines. Scattered among these data lines may be comment lines beginning with the # sign; they are ignored by the program.

It must be emphasized that a Matrix or Vector (with a capital M or V) in BUMP is not just an array of numbers. Rather it is an "object," that is, a data structure for which certain "actions" have been defined. The data structure of a Matrix contains, to be sure, the rectangular array of numbers; but it also has elements to inform functions which use the Matrix how many rows and how many columns the array has and what is
number of the first row or column. The "actions" defined for a Matrix by BUMP include "Display". Thus, after AF has been declared to be a Matrix, the statement

\[
\text{AF.Display("Here is the flow matrix: ", 5, 0);} \]

will display the matrix on the screen with the message in quotes above it. Each number is displayed in a field of five columns and is shown with 0 decimal places -- as indicated by the "5,0" at the end of the AF.Display statement. The Display action shows each row of the matrix as a folded vector. Each line is preceded by the column number of the first column on the line; the column number is preceded by the row number whenever a new row starts. (We call "actions" of an object what some authors refer to as "functions" and others as "methods" of an object. "Functions" is confusing because one can also define functions which have a Matrix among their arguments and should therefore be said to be "functions" of the Matrix, while "methods" in this use is simply an abuse of language.)

The advantages of a Matrix object over a simple rectangular array are many. We have already seen that BUMP provides easy ways of reading and writing the Matrix objects. Furthermore, if A, B, C and D have each been declared to be a Matrix and the compiler encounters a statement like

\[
\text{D = A*(B + C);} \]

it will put in calls to the correct routines in BUMP to do the matrix addition and multiplication and to store the result properly. We shall soon see other actions, functions, and operators provided by BUMP for matrices. Finally, we have lost nothing in flexibility in handling the array, for should we need to refer to element i,j of A, we can just write A(i,j), and the values of i and j will be checked that they are indeed within the limits of the matrix before returning the value of the element or storing a value in it. Vectors are like Matrices, but one of their dimensions, either the number of rows or the number of columns, must be 1.

Returning now to Ch1s1.cpp, we come next to the statements

\[
\text{Matrix FD("fd.dat"), Y("va.dat"); tap(); //Display message "Tap a key to continue" and wait.} \]

These declare and read the final demand and value-added matrices exactly as the intermediate flow matrix was declared. The tap() function stops the program to allow the screen to be read. The next statements,

\[
\text{n = AF.rows(); m = Y.rows();} \]

use the rows() method of a Matrix return the number of rows in AF and Y respectively. Notice that these values are determined by the data files, not by the program. These values will now be used to lay out space for other matrices and vectors by the following declarations.

\[
\text{// Declare other matrices and vectors Matrix AC(n,n), IMA(n,n), LINV(n,n), R(m,n);} \]
\[
\text{Vector q(n), f(n), s(n), p(1,n), v(1,n);} \]

The first makes up blank matrices AC, IMA, LINV, and R of the specified number of rows and columns. R, for example, has m rows numbered 1 to m and n columns, numbered 1 to n. The second line creates q, f, and s as column Vectors of n elements (numbered 1 to n) while p and v are made row Vectors of n elements.

The full form of a Vector declaration is

\[
\text{Vector x(nr,nc,temp,fr,fc)} \]


Where \( nr \) is the number of rows, \( nc \) is the number of columns, \( temp \) is either 'y' or 'n' according to whether the vector is temporary or not, \( fr \) is the number of the first row, and \( fc \) is the number of the first column. BUMP supplies default values for the last four of these as follows: \( nc = 1 \), \( temp = 'n' \), \( fr = 1 \), \( fc = 1 \). If you are willing to accept the last \( m \) of these default values, you need to specify only \( nr \) (the number of rows) and the first \( 4 - m \) of the rest. Thus, for the column vectors \( q \), \( f \), \( s \) we were willing to accept all the defaults, so we have just \( q(n) \), \( f(n) \), and \( s(n) \). For the row vectors \( v \) and \( p \), we want \( nr = 1 \) and \( nc = n \), while the defaults can be accepted for the remaining values, so we have to declare them as \( v(1,n) \) and \( p(1,n) \). The value of "temp" will almost always be 'n' for the user of Bump; in Bump's internal routines, however, it is often 'y'. Similarly, the full form for a Matrix is

\[
\text{Matrix } x(nr,nc,temp,fr,fc)
\]

where the parameters have the same meaning and same default values as before except that there is no default for \( nc \). In the program, \( AC \) will hold the intermediate coefficient matrix, \( IMA \), the \( I - A \) matrix; and \( R \), the resource coefficient matrix. (You may wonder why BUMP provides separate forms for vectors and matrices, since a column vector can be thought of as an \( n \)-by-1 matrix. The reason is that, given the way BUMP stores matrices, the space required for a \( nx1 \) vector as a Matrix would be twice what is required for the same information as a Vector.)

We now want to form the \( f \) vector as the row sum of the vectors in the \( FD \) matrix and then add to \( f \) the sum across the rows of \( AF \) to get the \( q \) vector. That, plus displaying the results, is simply accomplished by the statements:

```c
t// Form f as the row sum of the FD matrix.
f = rowsum(FD);
t// Form q as the rowsum of the intermediate flow matrix plus f.
q = rowsum(AF) + f;
t// Display f and q.
f.Display("Here is the f vector:",5,0);
q.Display("and the q vector."); tap();
```

Next we want to form the coefficient matrix, \( AC \), by dividing each column of \( A \) by the corresponding element of \( q \). That operation is represented in BUMP by the \% sign, which looks somewhat like a division sign. The statements are:

```c
t// Form the coefficient matrix.
AC = AF\%q;
AC.Display("The coefficient matrix",6,4);
```

Now we need to form the \( I - A \) matrix. We can easily get the -\( A \) part into \( IMA \) by writing \( IMA = -1*AC \). (Writing \( IMA = -AC \) will not work.) To add the 1’s to the diagonal, however, there is no special BUMP device, and we have to use just plain C as follows:

```c
t// Form the IMA = I - A matrix
IMA = -1.*AC;
for (i = 1; i <= n; i++)
    IMA(i,i) += 1.1;
```

The "for" statement can be read, "for \( i \) starting and 1, and while \( i \) is less than or equal \( n \), do the following statement and increment \( i \) by 1 each time it is done." In C, \( i++ \) means to increment the value of \( i \) by 1. (Thus the name of the C++ language meant C+1 to C programmers.) In C, the statement "\( a += y; \)" means to replace the value of \( a \) by \( a + y \). It is not only shorter but usually faster than \( a = a + y \), which, however, works.
For matrix inversion, BUMP uses the symbol ! in front of the matrix to be inverted. (A / in front of the matrix might have been more a intuitive symbol for inversion, but this choice was not available because / is not a unary operator in C. The ! seemed the closest to it among the possible choices.) We now calculate the inverse, store it in LINV and write it to the file linv.dat, from where it was taken for display in the previous section.

```
// Invert IMA.
LINV = !IMA;
LINV.Display("The Leontief Inverse Matrix", 6,4);
// Write out the Leontief Inverse to the file linv.dat.
writemat(LINV,"linv.dat", 6,4);
tap();
```

Multiplication of a Matrix, like LINV, by a vector, like f, is represented in BUMP by a * between the names of the matrix and vector: LINV * f.

```
// Compute outputs and display them.
s = LINV*f;
s.Display("The computed outputs.",10,4);
```

The rest of the program should now be self-explanatory, save to say that the colsum() function of a matrix gives the vector of column sums.

Here is a quick overview of the actions, operators, and functions available with BUMP. Some of them we have seen, but others were not needed in the examples above.

To read a Matrix or Vector A, use
```
A.ReadA("filename")
```
Where filename is the name of the file containing the matrix or vector. For a matrix, the file should just contain the numbers in a rectangular array. For vectors, just a list of the numbers is all that is needed.

The readmat() and similar readvec() functions described above are convenient for large matrices or vectors.

To display a Matrix or Vector A on the screen, use
```
A.Display("message", fieldwidth, decimals)
```
To write a Matrix A to a file use
```
writemat(A, filename,fieldwidth, decimals)
```
To write a Vector A to a file use
```
writevec(A, filename, fieldwidth, decimals).
```

If A is a Matrix or Vector and k is a scalar (a float in C terms), then
```
k*A
```
multiplies each element of A by k.
```
A/k
```
divides each element of A by k.

If A and B are both Matrices or both Vectors of the same dimension, then
```
A + B
```
gives the matrix or vector sum
```
A - B
```
gives the matrix or vector difference
```
ebemul(A,B)
```
gives the element-by-element product
```
ebediv(A,B)
```
gives the element-by-element quotient, the elements of A being divided by the corresponding elements of B. If, an element of B is zero, the corresponding element of A is returned in that position.

If A has the same number of columns as B has rows, then
```
A*B
```
gives the matrix product.
If A and B have the same number of columns, then
\[ A/B \]
gives the same thing as \(~A*B\), that is, the transpose of A multiplied by B, but without actually forming the transpose of A. This operator is provided in BUMP because the form \(A'B\) occurs frequently in statistics.

If x and y are both Vectors with the same number of elements:
\[ \text{dot}(x,y) \]
gives the inner product as a float.

If A is a Matrix and x a Vector with the same number of elements as A has columns,
\[ A\%x \]
gives the "coefficient" Matrix obtained by dividing each column of A by the corresponding element of x.

For a Matrix A, Vector v, float z, and int k,
\[ v.set(z) \]
sets all elements of Vector v to z.
\[ A.set(z) \]
sets all elements of Matrix A to z.
\[ \text{pulloutcol}(v, A, k) \]
pulls column k of A into v.
\[ \text{putincol}(v, A, k) \]
puts v into column k of A.
\[ \text{pulloutrow}(v, A, k) \]
pulls row k of A into v.
\[ \text{putinrow}(v, A, k) \]
puts v in row k of A.
\[ v = \text{colsum}(A) \]
puts the column sums of A into the vector v.
\[ v = \text{rowsum}(A) \]
puts the row sums of A into the vector v.
\[ z = v.sum() \]
puts the sum of the elements of v into z.
\[ v.First() \]
gives the number of the first row of v if v is column and vice versa.

If A is a square, non-singular matrix,
\[ !A \]
gives the inverse of A.
\[ A.invert(i,j) \]
transforms A into its inverse by Gauss-Jordan pivoting. The pivot operations start in row i and stop when the pivot has been in row j. If these arguments are omitted, the pivoting starts in the first row and continues through the last, to produce the true inverse.

The difference here is that \( !A \) does not change A but creates a new matrix for the inverse while \( A.invert() \) transforms A into its inverse. Thus, if memory space is scarce, the invert action may be preferable. The algorithm in both cases is Gauss-Jordan pivoting with no niceties. Don't trust it if your matrix poses any problems for inversion.

If A a is either a Vector or Matrix object, then
\[ ~A \]
gives the transpose of A.
\[ A.rows() \]
gives the number of rows as an integer.
\[ A.columns() \]
gives the number of columns as an integer.
\[ A.firstrow() \]
gives the number of the first row as an integer.
\[ A.lastrow() \]
gives the number of the last row as an integer.
\[ A.firstcolumn() \]
gives the number of the first column as an integer.
\[ A.lastcolumn() \]
gives the number of the last column as an integer.
\[ A.freeh() \]
frees all the heap memory occupied by A. This is useful when you need A for several operations but then no longer need it and want to use the space it occupies for something else. Don't try to use A after doing A.freeh().
For completeness, we mention three functions which will be explained in following sections. A Matrix $A$ can be balanced to have the row sums given by Vector $a$ and column sums given by Vector $b$ by the function
\[
\text{int ras}(A, a, b)
\]
If the sum of the elements of $a$ and $b$ are not equal, the user is required to pick which governs.

If $A$ is a square Matrix and $q$ and $f$ are Vectors of the appropriate dimension, the equation
\[
q = Aq + f
\]
can be solved by the Seidel iterative method (if it converges) by the function
\[
\text{Seidel}(A, q, f, \text{triang}, \text{toler})
\]
where $\text{triang}$ is an array of integers giving the order in which the rows of $A$ should be selected in the Seidel process, and $\text{toler}$ is a float giving the tolerance which is accepted in the iterative solution. Similarly, the equation
\[
p = pA + v
\]
can be solved by $\text{PSeidel}(A, p, v, \text{triang}, \text{toler})$;

Exercises
1. For the economy of our example, what levels of output and use of primary inputs would be required for the final demand (40, 6, 100, 600, 400, 170, 700, 148)? (You may modify the end of the ch1s3.cpp file to produce the program you need. You may also use $\text{f.ReadA()}$ to get the new final demand into the $f$ vector and $\text{q.Display()}$ and $\text{writevec()}$ to display and write the results to a file, respectively.
2. How much of each of the four factors does one dollar of each of the final demands contain?
3. Was this economy a next exporter or importer of depreciation?
4. What would happen to the prices of each of the eight products if all indirect taxes were eliminated?
5. Greenhouse gases are emitted by the production of the various sectors of our model economy. Measured in tons per billion dollars of output, the emission coefficients for the various sectors of our economy are
\[
2.1 \ 1.3 \ 6.1 \ 1.8 \ 1.0 \ 4.3 \ 0.8 \ 0.0
\]
What is the emission of greenhouse gases per billion dollars of final demand for each of the eight products? How much is attributable to a billion dollars of each of the types of final demand -- consumption, government, etc.? Was this country a next exporter or importer of greenhouse gas emissions?
6. The input-output flow table illustrated in the text was for year A. A comparable table for the same country but for a later year, year B, may be found in the file "yearb.prn" in the ch1.zip file of the software. Price indexes for the eight sectors from year A to B are given by the vector
\[
(1.01 \ 1.10 \ 1.06 \ 1.07 \ 1.15 \ 1.24 \ 1.18 \ 1.20),
\]
while the cost of labor increased twenty percent between the two years. (The price indexes are in the file $\text{pindex.dat}$.) What has happened between the two years to total labor requirements for producing one unit of final demand for each product? In answering this question, consider that the depreciation and capital income are produced with material inputs in the proportions given by the investment vector of the year in question. Ignore the indirect taxes and imports. The reciprocals of the labor requirements are productivity indexes for the economy in producing the various products supplied to final demand.
As we noted in section 1, it is impossible to know what has happened to productivity in a single industry, because the industry may have reduced its primary inputs while increasing its intermediate inputs; and the double-deflation method, supposed to handle this problem, is totally fallacious. The same problem does not arise in looking at total labor required, indirectly as well as directly, for the production of each unit delivered to final demand, for if the direct supplier to final demand has shifted required labor to other industries by buying more intermediate goods, that indirect labor will be automatically picked up. Thus, input-output calculations may offer a way of studying trends in productivity by product which elude methods which do not take into account indirect effects.
4. Iterative solutions of input-output equations

In actual input-output computations, the Leontief inverse is seldom used, for the equations \( q = Aq + f \) or \( p = pA + v \) can be solved directly from the \( A \) matrix in about the same time required to multiply \((I - A)^{-1}\) by \( f \) or \( v \). Thus, the effort of calculating \((I - A)^{-1}\) would be pointless. Moreover, for large matrices, many cells of \( A \) are zero. This fact can be exploited to reduce the computer storage required for the matrix. But the Leontief inverse will have non-zeroes nearly everywhere, so there is no way to reduce the space required for it. Further, changes to \( A \) are easily recorded and applied, but a change of one element in \( A \) can easily change all the elements in the inverse. Thus, from the point of view of solving the equations, nothing is gained and a good deal lost by computing the inverse.

How to solve the equations without the use of the inverse is the subject of this section. We will explain two methods of successive approximation, for it is worth knowing that both work even though we mainly use the second. The first, the simple iterative method, takes as a first approximation of \( q \), \( q^0 = f \). Then, given the \( k^{th} \) approximation, \( q^{(k)} \), the next approximation is

\[
q^{(k+1)} = Aq^{(k)} + f. \tag{1.4.1}
\]

If the process converges so that one \( q \) is indistinguishable from the previous one, then the vector to which it has converged is clearly the solution of the equation. In economic terms, we first set the output equal to the final demands. Then we increase it to allow for the intermediate goods needed by the first approximation and then increase it again for the intermediate goods needed for the second approximation, and so on.

It is clear from equation (1.4.1) that if the matrix \( A \) is non-negative and \( f \) is non-negative, then no element of \( q \) ever becomes negative in the course of the iterations. Thus, the conditions on \( A \) that insure the convergence also insure that a non-negative \( f \) leads to a non-negative \( q \). Thus, our inquiry, initially motivated by considerations of practical computation, also provides an answer to the theoretical question of whether an economy could exist with a given \( f \) and \( A \), for the economic interpretation of \( Aq \) is dependent on all elements of \( q \) being non-negative.

The second method, the Seidel process, takes the same first approximation, and then, to get the second approximation, solves first the first equation for \( q_1 \), given all the other elements of \( q \). Then, using this new value of \( q_1 \) and the old values of \( q_2, q_3, \ldots \), it solves the second equation for \( q_2 \), and so on. If the \( A \) matrix is triangular, that is, if all the entries above the main diagonal are zero, this method gives the right answer with one iteration. If it is not triangular, the whole process is repeated until little or no change occurs with each new iteration. While no actual input-output matrix is ever exactly triangular, the sectors can often be taken in an order which makes the matrix almost triangular, and this almost-triangularity speeds the convergence process.

Instead of starting this process with the final demands, it is also possible to start with any guess of \( q \). In dynamic models, a good guess, namely the previous year's \( q \) is available. With a good starting point, four or five iterations of the Seidel process is usually sufficient to produce adequately accurate solutions. If twenty percent of the elements of \( A \) are non-zero -- a fairly typical situation -- we can make five iterations of the Seidel process in the same time which would be required to multiply \( f \) by the inverse if we had it.

If \( A \) is not an input-output matrix but just any old matrix you happen to meet on the street, there is not much chance that either of these methods will converge and give a solution. What then makes us so sure that they will converge for an input-output matrix? To discuss convergence, we need to be able to say how far apart two vectors are. The concept of the norm of a vector gives us that ability. We even need to be able to say how
far a given vector is from the solution when we do not know what the solution is. The concept of the norm of a matrix enables us to turn that trick. We will now explain these two concepts.

We can say how far apart two vectors are if we can say how "long" a vector $x$ is, that is, how long the line is which connects $x$ with the origin or zero point. For if $\|x\|$ represents the length of any vector, then the length of the difference of two vectors $a$ and $b$, $\|a-b\|$, serves as a measure of how far apart they are. How shall we measure the length of a vector? In two dimensions, the usual length of the vector $(x_1, x_2)$ is $\sqrt{x_1^2 + x_2^2}$. This concept of length readily generalizes to vectors of any dimension by the definition $\|x\| = \sqrt{x^T x}$. This formula, called the Euclidean length (or norm), gives one possible way of measuring length.

Why, however, do we bother to take the square root in the Euclidean norm? Because we certainly want any way of calculating the length of $x$ to be such that multiplying each element of $x$ by a scalar, $\lambda$, multiplies the length of $x$ by the absolute value of $\lambda$:

(a) $\|\lambda x\| = |\lambda| \|x\|$.  
Other properties which any definition of length should have are

(b) $\|0\| = 0$ and $\|x\| > 0$ if $x \neq 0$  
and

(c) $\|x+y\| \leq \|x\| + \|y\|$.  

Property (c) expresses the requirement that the shortest distance between any two points must be a straight line. Let us denote the points by $x$ and $-y$. Then we must have $\|x-(-y)\| \leq \|x\| + \|-y\|$, since $\|x\|$ is the distance from $x$ to 0 (the origin of the vector space) and $\|-y\|$ is the distance for 0 to $-y$, while $\|x-(-y)\|$ is the distance directly from $x$ to $-y$. By applying property (a) to the second term on the right, this requirement may be written more simply as (c) above.

Any way of assigning a number, $\|x\|$, to each vector, $x$, of the vector space in such a way that (a), (b), and (c) are satisfied is called a norm of the space, and $\|x\|$ is read "the norm of $x$". It is quite remarkable that we can often prove the convergence of a process in terms of a norm without knowing exactly which norm we are using. Besides the Euclidean norm, there are two more important examples of norms:

the l-norm: $\|x\| = \sum_{i=1}^{n} |x_i|$  

the m-norm: $\|x\| = \max_{i} |x_i|$  

You may easily verify that each of these norms has the required three properties, though the values they give as the norm of a given vector may be quite different. For example, the vector $(1, -3, 2)$ has a Euclidean norm of 3.74, while its l-norm is 6 and its m norm is 3. (The l in l-norm refers to Henri Lebesgue, a French mathematician of the early years of the twentieth century.)

Exercise 7: Draw the unit circle for each of these three norms. (The unit circle is the locus of points with norm 1.)
With each of these three norms, if $x^k$, for $k = 0, 1, 2, \text{ etc.}$, is a sequence of vectors and $x^*$ is a vector such that
\[ \lim_{k \to \infty} \| x^k - x^* \| = 0, \]
then
\[ \lim_{k \to \infty} x^k = x^*. \]
That is, convergence of a sequence of vectors in norm implies element-by-element convergence. This property is easily seen for the examples of the three norms and is a characteristic of finite dimensional vector spaces.

What we now want to show is that if $q^*$ is a solution of the input-output equations, so that
\[ q^* = Aq^* + f, \tag{1.4.2} \]
then the sequence $q^0$, $q^1$, $q^2$, ... defined by
\[ q^{k+1} = Aq^k + f \tag{1.4.3} \]
converges in norm to $q^*$. Subtracting the first equation, (1.4.2), from the second, (1.4.3), gives
\[ q^{k+1} - q^* = A(q^k - q^*), \quad k = 1, 2, 3, \ldots . \tag{1.4.4} \]
If we have computed to iteration $m$, then setting $k = m$ in this equation gives
\[ q^{m+1} - q^* = A(q^m - q^*). \]
But setting $k = m+1$ in (1.4.4) gives
\[ q^{m+2} - q^* = A(q^{m+1} - q^*). \]
Together the last two equations imply
\[ q^{m+2} - q^* = A(q^{m+1} - q^*) = A^2(q^m - q^*). \]
For any positive integer, $p$, similar reasoning applied $p$ times gives
\[ q^{m+p} - q^* = A^p(q^m - q^*). \tag{1.4.5} \]
We would like to be able to show that the norm of the vector on the left of (1.4.5) goes to zero as $p$ goes to infinity. To do so, we need to extend the concept of norm to matrices. We introduce that extension by a question:

Is there a number, call it $|A|$, such that
\[ \|Ax\| \leq |A| \|x\| \]
for all $x$?

There are indeed such numbers, and we call the least of them (for any norm of the vectors) the norm of $A$. Intuitively speaking, the norm of the matrix $A$ is the greatest "stretch" which multiplication by $A$ performs on any vector. For the I-norm and m-norms of the vectors, the corresponding norms of a matrix are easily
computed, as we shall see in a moment. Note that the norms of matrices also have the three basic properties of the norms of vectors:

a) \( \|A\| = 0 \) if and only if \( A = 0 \).

b) \( \|\lambda A\| = |\lambda| \|A\| \)

c) \( \|A + B\| \leq \|A\| + \|B\| \)

plus a fourth, which can be easily verified from the definition

d) \( \|AB\| \leq \|A\|\|B\| \).

We can apply this inequality repeatedly to equation (1.4.5). After applying it \( p \) times, we have

\[
\| q^{m-p} - q^* \| = \|A\|^p \| q^m - q^* \|
\]

If we can show that \( \|A\| \leq 1 \) for some norm, then \( \|A\|^p \rightarrow 0 \) as \( p \rightarrow \infty \), and therefore \( q^k \rightarrow q^* \) as \( k \rightarrow \infty \), and the iterative calculations converge to the solution.

The norm of the \( n \times n \) matrix \( A \) induced by the \( m \)-norm of vectors, and therefore called the \( m \)-norm of the matrix, is

\[
\|A\|_m = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|.
\]

while the norm of \( A \) induced by the \( l \)-norm of vectors, and therefore called the \( l \)-norm of the matrix, is

\[
\|A\|_l = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|.
\]

We shall prove the formula for the \( l \)-norm, and leave that for the \( m \)-norm as an exercise. (The Euclidean norm of \( A \) is more complicated and not of immediate concern to us. It is the largest characteristic root of \( A^TA \).)

For the \( l \)-norm, let

\[
\alpha = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|.
\]

Then \( \|Ax\|_l \leq \alpha \), because

\[
\|Ax\|_l = \sum_{i=1}^n \left| \sum_{j=1}^n a_{ij}x_j \right| \leq \sum_{i=1}^n \sum_{j=1}^n |a_{ij}| \cdot |x_j| = \sum_{i=1}^n |x_j| \sum_{j=1}^n |a_{ij}| \leq \sum_{i=1}^n |x_j| \alpha \leq \alpha \|x\|_l.
\]

On the other hand, let \( k \) be the number of the column with the largest sum of absolute values, so that

\[
\alpha = \sum_{i=1}^n |a_{ik}|
\]

and then choose a vector, \( x \), with \( x_k = 1 \) and \( x_j = 0 \) for \( j \neq k \). Then \( \|x\| = 1 \), and

\[
\|Ax\|_l = \sum_{i=1}^n |\sum_{j=1}^n a_{ij}x_j| = \sum_{i=1}^n |a_{ik}| = \alpha = \alpha \|x\|_l.
\]

Therefore, \( \|A\|_l \geq \alpha \). But we have already shown the opposite inequality, so the only possibility is that \( \|A\|_l = \alpha \).
If an input-output $A$ matrix comes from an observed economy with a positive value-added in every industry, then the column sums of every column are less than 1.0 and therefore the l-norm of the matrix is less than 1. Thus, returning to the iterative solution of the input-output equations, we see that it will indeed converge if such is the source of $A$. Furthermore, in that case, $(I - A)^{-1}$ will be non-negative, because if we start from an $f$ vector which is all zero except for a 1 in some position, the resulting solution will never have any opportunity to acquire any negative elements in the course of the iterative process. But the columns of $(I - A)^{-1}$ are precisely the solutions of such equations, so the whole matrix is non-negative.

The norm of the $A$ matrix not only allows us to be sure that the iterative process converges, it also allows us to set an upper bound on how far we are from the solution at any stage. If, as before, $q^k$ indicates approximation $k$, then

$$q^{k+p} - q^k = q^{k+1} - q^k + q^{k+2} - q^{k+1} + ... + q^{k+p} - q^{k+p-1}$$  \hspace{1cm} (1.4.7)

But since

$$q^{m+1} = Aq^m + f$$ and $$q^m = Aq^{m-1} + f$$

for any positive integer $m$, subtraction gives

$$q^{m+1} - q^m = A(q^m - q^{m-1}).$$

Repeatedly applying this equation gives

$$q^{k+1} - q^k = A(q^k - q^{k-1})$$

$$q^{k+2} - q^k = A^2(q^k - q^{k-1})$$

$$q^{k+p} - q^{k+p-1} = A^p(q^k - q^{k-1})$$

and substitution in the above equation (1.4.7) gives

$$q^{k+p} - q^k = (A + A^2 + A^3 + ... + A^p)(q^k - q^{k-1})$$

Taking the norms of both sides and applying properties c and d of the norms of matrices gives

$$\|q^{k+p} - q^k\| \leq \|(A + A^2 + A^3 + ... + A^p)\|\|q^k - q^{k-1}\|$$

$$\leq (\|A\| + \|A\|^2 + \|A\|^3 + ... + \|A\|^p)\|q^k - q^{k-1}\|.$$  

Now as $p \to \infty$, $q^{k+p} - q^k$ and the sum of the geometric progression on the right goes to $\|A\|/(1 - \|A\|)$ because $\|A\| < 1$. Thus, when we have reached iteration $k$, we know that the distance to the true solution is less than $\|q^k - q^{k-1}\| \|A\|/(1 - \|A\|)$. In other words, when the differences of the successive approximations get small, we can be sure that we are close to the true solution.

Now suppose for a moment that $A$ is a matrix in physical units -- with coefficients in units like kilowatt hours per pound -- so that column sums are meaningless and the l-norm perhaps much greater than 1. Further let $w$ be an all-positive vector of the hours of labor -- the only primary input -- required per physical unit of output in each industry. Can an economy exist with this technology? In other words, if the vector $f$ of final demands is all positive, will the vector of outputs, $q$, such that $q = Aq + f$ also be all positive? (Mathematically, it is quite possible for some element of $q$ to be negative, but it is economic nonsense to run
an industry at a negative level. Coal can be converted into electricity, but all the electricity in the world can’t make a ton of coal.)

The answer to these questions lies in the solution of \( p = pA + w \) (where \( p \) is a row vector). If \( p \) is all positive, then it can be thought of as a vector of prices (with an hour of work as the numeraire) at which each process has a positive value added. If we now change the units of measurement of output of each product to one “hour’s worth,” the coefficient matrix, say \( A^* \), in these new units corresponding to \( A \) in the old units will have columns whose sums are each less than 1. Thus, in these units, the iterative procedure will converge. But the iterative procedure in the original units (with \( A \)) would give successive approximations which differ from those with \( A^* \) only in their units. Hence the process would converge in the original units as well and \((I - A)^{-1}\) will be non-negative. Since the Leontief inverse is non-negative, any vector of non-negative final demands can be met by non-negative levels of output of all the industries.

1.5 The Seidel method and triangulation.

As mentioned at the outset of the previous section, there is a variation of the iterative method, known as the Seidel method, which generally converges even faster. In it, one starts with any initial guess, \( q^{(0)} \), of the solution just as in the simple iterative method but then as the value of each variable is calculated in each iteration, the new value for that variable is used in calculating the values of the remaining variables in that iteration. Formally, on iteration \( k+1 \),

\[
q_i^{(k+1)} = \sum_{j=1}^{n} a_{ij}q_j^{(k)} + f_i \tag{1.5.1}
\]

and for successive variables

\[
t_i^{(k+1)} = \sum_{j=1}^{i-1} a_{ij}q_j^{(k+1)} + \sum_{j=i}^{n} a_{ij}q_j^{(k)} + f_i \tag{1.5.2}
\]

Note that if the input-output matrix is triangular in the sense that all elements above the main diagonal are zero – that is, \( a_{ij} = 0 \) if \( j > i \) – then one iteration of the Seidel process is enough to give the exact solution of system. Many input-output matrices can be arranged so that they are approximately triangular. Thus, if one puts the three industries (1) Bread, (2) Flour milling, and (3) Agriculture in that order, the major material inputs to the first two will lie below the main diagonal. For such matrices, the Seidel process may converge much more quickly than the simple iterative method. We need to show, however, that the process is sure to converge if \( \|A\|_1 \leq 1 \), the condition which input-output matrices usually satisfy. We will then deal with speed of convergence.

For the exact solution vector, \( q^* \), we have

\[
t_i^* = \sum_{j=1}^{i-1} a_{ij}q_j^* + \sum_{j=i}^{n} a_{ij}q_j^* + f_i \tag{1.5.3}
\]

Subtracting equation (2) from (3) gives
\[ q_i^* - q_i^{(k+1)} = \sum_{j=1}^{i-1} a_{ij}(q_j^* - q_j^{(k+1)}) + \sum_{j=i}^{n} a_{ij}(q_j^* - q_j^{(k)}), \quad \text{for } i = 1, 2, \ldots, n; \]

and so

\[ q_i^* - q_i^{(k+1)} \leq \sum_{j=1}^{i-1} |a_{ij}| |(q_j^* - q_j^{(k+1)})| + \sum_{j=i}^{n} |a_{ij}| |(q_j^* - q_j^{(k)})| \quad \text{for } i = 1, 2, \ldots, n. \]

Summing this last set of inequalities gives

\[ \sum_{i=1}^{n} |q_i^* - q_i^{(k+1)}| \leq \sum_{i=1}^{n} \sum_{j=1}^{i-1} |a_{ij}| |(q_j^* - q_j^{(k+1)})| + \sum_{i=1}^{n} \sum_{j=i}^{n} |a_{ij}| |(q_j^* - q_j^{(k)})|. \]

The first term on the right involves only elements of \( A \) below the main diagonal, while the second term involves only elements on or above the diagonal. We can rearrange the order of summation of these two terms to give

\[ \sum_{i=1}^{n} |q_i^* - q_i^{(k+1)}| \leq \sum_{j=1}^{n} \left( \sum_{i=1}^{j-1} |a_{ij}| |(q_j^* - q_j^{(k+1)})| + \sum_{i=j+1}^{n} |a_{ij}| |(q_j^* - q_j^{(k)})| \right) j \sum_{i=1}^{j} |a_{ij}|. \quad (1.5.4) \]

To simplify our equations, we introduce the notation

\[ s_j = \sum_{i=j+1}^{n} |a_{ij}| \quad \text{and} \quad t_j = \sum_{i=1}^{j} |a_{ij}| \quad \text{for } j = 1, 2, \ldots, n-1, \text{ and} \]

\[ s_n = 0 \quad \text{and} \quad t_n = \sum_{i=1}^{n} |a_{ij}|. \]

It may help to remember that \( t_j \) is the sum of the top of column \( j \) above the diagonal while \( s_j \) is the substance of the column, the part that “stands under” the diagonal. Clearly

\[ s_j + t_j = \sum_{i=1}^{n} |a_{ij}| \leq \| a \|_1 < 1, \]

so \( s_j < 1 \) and

\[ t_j \leq \| a \|_1 - s_j \leq \| a \|_1 - \| a \|_1 s_j = \| a \|_1 (1-s_j). \quad (1.5.5) \]

One final notation is necessary to make the argument clear. Let

\[ \rho = \max_j \frac{t_j}{1-s_j} \leq \| a \|_1 < 1, \]
so \[ \frac{t_j}{1 - s_j} \leq \rho \] and \( t_j \leq \rho (1 - s_j) \) for all \( j \).

By using this notation, equation (4) becomes

\[
\sum_{i=1}^{n} |q_i^* - q_i^{(k+1)}| \leq \sum_{j=1}^{n} s_j (q_j^* - q_j^{(k-1)}) + \sum_{j=1}^{n} t_j (q_j^* - q_j^{(k)})
\]

or

\[
\sum_{i=1}^{n} (1 - s_j)|q_i^* - q_i^{(k+1)}| \leq \sum_{j=1}^{n} t_j (q_j^* - q_j^{(k)}) \leq \rho \sum_{j=1}^{n} (1 - s_j) |(q_j^* - q_j^{(k)})|.
\] (1.5.6)

By repeating this inequality for each iteration from the first, we have

\[
\sum_{i=1}^{n} (1 - s_j)|q_i^* - q_i^{(k+1)}| \leq \rho \sum_{j=1}^{n} (1 - s_j) |(q_j^* - q_j^{(0)})|.
\] (1.5.7)

Since \( \rho \leq \| a \|_p < 1 \), in the limit as \( k \to \infty \)

\[
\sum_{i=1}^{n} (1 - s_j)|q_i^* - q_i^{(k)}| \to 0,
\]

so, since \( s_j < 1 \) for all \( j \), \( f_i^{(k)} \to q_i^* \), as we wanted to show. Moreover, since \( \rho \leq \| a \|_p \) the convergence is potentially faster and certainly no slower than with the simple iterative method.

It remains to find an upper bound on the error in the solution at any stage. If we had started, not by subtracting equation (2) from (3) but by subtracting the previous iteration of (2), namely

\[
q_i^{(k)} = \sum_{j=1}^{n} a_j q_j^{(k)} + \sum_{j=1}^{n} a_j q_j^{(k-1)} + f_i
\]

from (2) and performed exactly the same transformations as before we would have reached, at the point where we had previously reached (6),

\[
\sum_{i=1}^{n} (1 - s_j)|q_i^{(k-1)} - q_i^{(k)}| \leq \rho \sum_{j=1}^{n} (1 - s_j) |(q_j^{(k)} - q_j^{(k-1)})|.
\] (1.5.8)

Here it will help to define the quantity on the left of (8) to be \( \sigma_{k+1} \) so that (8) becomes

\[
\sigma_{k+1} \leq \rho \sigma_k,
\]

whence

\[
\sigma_{k+p} \leq \rho^p \sigma_k.
\]
If we let $s$ be the largest value of $s_j$, then
\[
(1 - s) \sum_{i=1}^{n} |q_i^* - q_i^{(k)}| \leq \sum_{i=1}^{n} (1 - s_j)|q_i^* - q_i^{(k)}| \leq \frac{\rho}{(1 - s)(1 - \rho)} \sigma_k.
\]

This last inequality gives us an upper bound on the error at any iteration. Note that it shrinks with each iteration because $\sigma_k$ is shrinking.

In this description of the Seidel procedure, the diagonal elements of the matrix have been multiplied, at each iteration, by the outputs from the previous iteration. In practice, these diagonal elements are often relatively large and convergence is faster if, instead of equation (2), we use
\[
q_i^{(k+1)} = \left( \sum_{j=1}^{i-1} a_{ij}q_j^{(k+1)} + \sum_{j=i+1}^{n} a_{ij}q_j^{(k)} + f_i \right)/(1 - a_{ii}).
\]

If all the non-zero elements of $A$ are on or below the main diagonal, $A$ is said to be triangular. If $A$ is triangular, all the $t$ are zero, $\rho$ is zero, and one pass of the Seidel process is sufficient to reach the exact solution. If $A$ is merely almost triangular, a few iterations will suffice for a good solution. It general, input-output matrices arrive from the statistical offices more or less triangulated in exactly the wrong way. They start with Agriculture first, later Textiles, then Apparel. The right order for a fast Seidel solution is the reverse, Apparel, Textiles, Agriculture. It is not, however, necessary to physically re-arrange the rows and columns. All that is necessary is to take the rows in the Seidel operation in the order that would make the matrix nearly triangular.

For small matrices, one can usually specify a fairly good triangular order from just a general knowledge of the relations among industries. For large matrices, however, it may be convenient to have a mechanical way to generate an approximately triangular order. A simple but effective is to pick as the first industry the one which has the smallest ratio of intermediate to final demand in its row. Then move into final demand all the inputs into this industry and again pick from the remaining sectors the one with the lowest ratio of intermediate to final in its row. Continue until all industries have been selected.

Solving input-output equations by the Seidel method is not only generally much faster than inverting the $I - A$ matrix by Gauss-Jordan reduction, it may even be faster than multiplying $(I - A)^{-1}$ by $f$ when $(I - A)^{-1}$ is already known. How can that be? It is common for the $A$ matrix to be quite sparse. A 300-by-300 matrix may have some 9,000 non-zero elements, not 90,000. It can be stored in a “packed” form in which only non-zero elements are stored, and the Seidel algorithm can be written to use this packed form, so that only as many multiplications and additions are required per iteration as there are non-zero elements. Thus, if the Seidel process requires less than ten iterations in our example, it will require less than 90,000 multiplications and additions. The Leontief inverse, however, will generally have 90,000 non-zeroes and thus multiplying it by
f involves exactly 90,000 multiplications and additions. To economize on both space and solution time, large, sparse matrices are thus best stored in a packed form; and equations involving them should be solved by the Seidel process without ever inverting the matrix.

Exercises

8. Using Bump, write a program to compute the triangular order of a matrix. Apply it to the flow matrix used as an example in this chapter. Write the results as a vector of integers, the first being the number of the equation to be taken first; the second, that of the equation to be taken second, etc.

9. Write a program to use the Seidel method to solve input-output equations, taking the equations in the order specified by the vector produced in exercise 7. Apply the program to solve exercise 1 earlier in this chapter. (Bump has a Seidel method. Try to create yours without looking at it.)

Answers

3.1. Below the line "R.Display("The R matrix.",6,4); tap(); ", add

    // Exercise 1
    f.ReadA("ch1s3x1.dat");
f.Display("The new final demand vector.");
q = LIINV*f;
q.Display("The new outputs");
Vector primary(m);
primary = R*q;
primary.Display("The primary resource vector.");
writevec(q,"newout.dat",8,2);
writevec(primary,"primary.dat",8,2);
tap();
The file ch1s3x1.dat is simply
40 6 100 600 400 170 700 148
The new outputs are
166.14 55.21 222.30 763.57 426.48 206.41 812.58 148.00
and the primary resources required to produce them are
317.24 1351.84 268.34 226.58.
3.3 The net export of depreciation is -5.73.
3.4 The new price vector is (.92 .96 .89 .90 .71 .93 .96 1.00).
3.5 Net export of greenhouse gas production is -31.87.
3.6 A program to solve this problem is found in the file ch1s3x6.cpp
A Historical Note

All of us tend to presume that the world was made the way we found it; if there were input-output tables in it when we arrived, then they must have always been there. Of course, that is not the case. In fact, they are so much connected with the work of one man, Wassily W. Leontief, that without his remarkable contribution they would probably not have been developed until decades later. Born in St. Petersburg in 1906, he was already a university student when the Bolsheviks began taking over the educational program. He joined a group protesting this process, was caught pasting up a poster, spent a while in jail and was periodically jailed and interrogated thereafter. Though deeply interested in the economy of his country and in the efforts at economic planning, he clearly had little to hope for from the Bolshevik government. Even as an undergraduate, however, his paper on "The Balance of the Economy of the USSR" describing efforts in Russia to investigate interindustry relations came to the attention of professors in Germany. When he graduated from the University of Leningrad in 1926, he was offered the possibility of graduate study in Germany, but it was already difficult to get out of the Soviet Union. By an extraordinary turn of fate, he developed a bone tumor on his jaw. It was removed, but the surgeon warned him that he would surely soon die. Armed with the surgeon's written statement, he argued to the officials that he should be allowed to leave the country since he would certainly be useless and possibly expensive to the government. The argument worked, and he arrived in Germany with the tumor in a bottle. It was there re-examined and found ... benign! His work in Germany led, via Nanjing, to an appointment at the National Bureau of Economic Research in New York. His theoretical writings came to the attention of the Harvard faculty which offered him an instructorship. He accepted the Harvard offer on the condition that he be given a research assistant to help him build what we would now call an input-output table. The reply informed him that the entire faculty had discussed his request and had unanimously agreed that what he proposed to do was impossible and, furthermore, that even if it were done, it would be useless. Nonetheless, they were so eager to have him come that they would grant the request and hope that he would use the resources for better purposes. He didn’t. In 1936, his first results were published; in 1939 a book *The Structure of the American Economy* appeared. It had input-output tables for the United States for 1919 and 1929. The theoretical parts of the book had the major ideas of input-output analysis: coefficients, simultaneous solution, and price equations. During World War II, Leontief constructed, with support of the U.S. Bureau of Labor Statistics (BLS), a 96-sector table for 1939 and, by 1944 was able to study changes in employment patterns which could be expected after the end of the war. In 1947, a second edition of the book appeared with the addition of a 1939 matrix and a comparison of input-output and single-equation projections. In 1973, he was awarded the Nobel prize in economics for this work. Leontief remained active until shortly before his death in 1999 at the age of 93.

In 1949, a group at the BLS began work on a 400-sector table for 1947. A 190-sector table was published in 1952, but financing -- which had come through the Defense budget -- for the more than fifty people working on the project was discontinued early in the Eisenhower administration, so that neither the full table nor the extensive documentation of the details of its production were ever published.

In other countries, making of tables spread rapidly. They were incorporated in the United Nation's standard System of National Accounts prepared by Richard Stone. In 1950, the first international conference on input-output methods was sponsored by the United Nations; the eleventh (without U.N. support) was held in 1995.

---

2 The spelling of Leontief's name in Latin letters was for German speakers; English speakers almost invariably mispronounce it, though I never heard him correct anyone. In *Wassily*, the *W* is pronounced *V*, the *a* is long as in "father," and the accent is on the *si* which is pronounced "see". In *Leontief* the accent is on *on* and the *ie* is pronounced like the *ye* in "yet". The final *f* is a soft *v*.
In the late 1950's, Soviet authors, eager to make input-output acceptable in their country, put together a table for the Soviet Union in 1924 and argued that all the essential ideas had originated in the Soviet Union. The difference, however, between what they could find in the literature of that period and Leontief's comprehensive treatment only heightens an appreciation of his contribution.

Gradually, it has come to be recognized that an input-output table is not only useful for economic analysis and forecasting but is also an essential step in making reliable national accounts. The statistical offices of most major industrial countries, therefore, prepare input-output tables, often on a regular basis. Annual tables for France, the Netherlands, Norway, and Japan are prepared as a part of annual national accounting. In the USA, a comprehensive table is made every five years in the years of economic censuses (years ending in 2 and 7) and is used in revising and "benchmarking" the national accounts.

In 1988, the International Input-Output Association was organized as a group of individuals interested in using input-output techniques. In 1989, it began publishing its own journal, *Economic Systems Research.*
Chapter 12

Matrix Balancing and Updating - the RAS Method

1. The RAS algorithm

Making an input-output matrix from scratch for a country is a major undertaking often involving a group of ten or more persons for a number of years. By the time the project is finished, the matrix refers to a year that is apt to seem part of ancient history. Hence the question arises, Given an input-output table for a base year, is there a way to update it to a more recent year with less work than making the table from scratch? In this updating, one usually has some data for the more recent year. One wants the matrix for this year, which we may call the target year, to conform to all those data.

Usually those data include at least industry outputs, major GDP components, and value-added by industry. The value-added by each industry can then be subtracted from its output to give the total intermediate inputs by each industry. Thus, we would know the row total for each industry and the column total for each final demand column and for the intermediate use of each industry. An obvious check on the accuracy of this information is that the sum of the row totals equals the sum of the column totals. We will assume that this condition has been met, although meeting it is not always easy except by a rough scaling. Thus, we have the margins or frame for the table for the target year.

An initial guess of the inside of the table for the target year can then be made by assuming constant coefficients for the input-output coefficients and for the shares in each of the final demand vectors. More sophisticated initial estimates could also be made. One could use, for example, consumption functions to “forecast” the purchases of households. However the initial inside elements of the table are estimated, it is almost certain that they will not have the right row and column sums. Adjusting them to make them conform to these control totals is generally done by what has come to be called the RAS procedure, a name derived from notation in Richard Stone’s description of the method in *A Computable Model of Economic Growth* (Chapman and Hall, London, 1962). The idea had been mentioned by Leontief in the 1941 edition of *The Structure of the American Economy*, but the idea seemed to pass unnoticed until applied by Stone.

The method is extremely simple in practice. First scale all of the rows so that each has the correct total. Then scale all the columns so that each has the correct total. The row sums are then probably no longer correct, so scale them again, and then scale the columns again, and so on until the scaling factors have converged to 1.0. The matrix at that point has the desired row and column sums. If $A'$ denotes the flow matrix at stage $t$ of the operation, $R'$ denotes the row scaling factors at step $t$ arrayed as the diagonal elements of an otherwise zero matrix, and $S'$ denotes the column scaling factors similarly arrayed, then the flow matrix at the beginning of stage $t+1$ is

$$A^{t+1} = R' A' S'. \quad (1)$$

The expression on the right gave rise to the name RAS, which should be pronounced as the three letters, though foreign speakers of English often turn it into one syllable, “ras”.

2. Convergence of the algorithm

The practice is simple, but will the process converge? To answer that question, we will need some notation. Let the original matrix be $A$, whose elements we will denote by $a_{ij}$, let $b$ be the positive vector of required row
s and $c$ be the positive vector of required column sums. The first condition is that $A$ be non-negative. The second is simply that there must exist at least one matrix with zeroes everywhere that $A$ has zeroes and positive numbers everywhere that $A$ has positive numbers which has row sums equal to $b$ and column sums equal to $c$.

Notice that this second condition did not assume a solution of the form we are seeking, that is, derived from $A$ by scaling the rows and columns. It does, however, have some important implications. The first is that the sum of the elements of $b$ must be the same as the sum of the elements of $c$. A further implication is that, if it is possible rearrange the rows and columns of $A$ so that an all-zero block appears, then the corresponding subtotals of $b$ and $c$ must be consistent with those blocks remaining zero while the other cells are positive. For example, if

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$$

then we must also have $b_1 < c_1$. In practice, one insures that the first implied condition (the equality of the sum of row sums and column sums) is met before beginning the RAS calculations. If they fail to converge, then one looks for inconsistencies along the lines of the second implication.

The proof of the convergence of the RAS procedure under these general conditions is requires a complicated notation. The essence of the proof, however, can be seen in the special case in which $A$ is all positive, and we will limit ourselves to that case. (For the general case, see M. Bacharach, Biproporitonal Matrices. Cambridge University Press.)

We will start the process by scaling the rows, then the columns, and so on. In the first row scaling, we choose the first-round row-scaling factors by

$$r_i^{(1)} = b_i / \sum_j a_{ij}.$$  

where the superscript on the $r$ refers to the iteration number. Then we compute the first-round column scaling factors by

$$s_j^{(1)} = c_j / \sum_i r_i^{(1)} a_{ij}.$$ 

Then we come back to compute the second-round row scaling factors,
Thus, we can see that the second-round row factors are reciprocals of convex combinations of the first-round column factors, that is, they are reciprocals of a weighted average of those first-round column factors with positive weights which sum to 1. Thus,

$$\text{Max}_{i} r_{i}^{(2)} \leq 1/(\text{Min}_{j} s_{j}^{(1)}) \quad \text{and} \quad \text{Min}_{i} r_{i}^{(2)} \geq 1/(\text{Max}_{j} s_{j}^{(1)}) .$$

By similar reasoning,

$$\text{Max}_{j} s_{j}^{(1)} \leq 1/(\text{Min}_{i} r_{i}^{(1)}) \quad \text{and} \quad \text{Min}_{j} s_{j}^{(1)} \geq 1/(\text{Max}_{i} r_{i}^{(1)}) .$$

The inequalities in (6) imply

$$\frac{1}{\text{Max}_{j} s_{j}^{(1)}} \geq (\text{Min}_{i} r_{i}^{(1)}) \quad \text{and} \quad \frac{1}{\text{Min}_{j} s_{j}^{(1)}} \leq (\text{Max}_{i} r_{i}^{(1)}) .$$

Then combining the first inequality of (5) with the second of (7) and the second of (5) with the first of (7) gives

$$\text{Max}_{i} r_{i}^{(2)} \leq 1/(\text{Min}_{j} s_{j}^{(1)}) \leq \text{Max}_{i} r_{i}^{(1)} \quad \text{and} \quad \text{Min}_{i} r_{i}^{(2)} \geq 1/(\text{Max}_{j} s_{j}^{(1)}) \geq \text{Min}_{i} r_{i}^{(1)}. $$

In other words, the biggest element of $r$ diminishes from iteration to iteration while the smallest rises. Since $A$ is all positive, all of the inequalities in (5) through (8) will be strict inequalities unless all the elements of $r$ are equal or all the elements of $s$ are equal. But if they are all equal, they must be all be equal to 1, for otherwise the scaling would increase or decrease the total of all elements in the matrix, contrary to the fact that, after the first row scaling, the sum of all elements remains equal to the common sum of the vectors $r$ and $s$. Since the sequences of $r^{(k)}$ and $s^{(k)}$ vectors both lie in closed, bounded sets, they have limit points. Can these limit points be other than the vectors that are all 1’s? No, because at any such point, one more iteration of the process would bring a finite reduction of the maximum element (and a finite increase in the minimum element) of each vector. (This is where we use the all positive assumption to have strict inequalities in (8).) Thus, for points sufficiently close to these limit points, the next iteration must also bring lower maximal and
higher minimal elements than those of the limit point, contrary to the limit point being a limit point. Therefore the unique limit of each sequence of vectors is a vector of ones.

Thus the convergence is proven for the case of all positive \( A \). The proof is similar for \( A \) with some 0 elements, but in this case, it may require several iterations to get a finite reduction in the maximal elements of \( r \) and \( s \).

In practice, the condition that the sum of \( b \) equals the sum of \( c \) is checked and assured before the iterative process begins. The initial \( r \) and \( s \) vectors should be reported by the program because they often indicate discrepancies between \( b \) and \( c \) vectors and the initial \( A \) matrix. Once the iterations start, the largest and smallest elements of the \( r \) and \( s \) vectors should be reported every five or ten iterations. It is common to observe “wars” between a row control and a column control when one element looms large in both its row and column but the control totals for the two are quite different. Such “wars” are symptomatic of a failure of the second assumption and an indication that the \( b \) and \( c \) vectors should be revised.

It should be noted that the RAS procedure works for rectangular matrices just as well as for square ones. It is also useful in making input-output tables and the bridge matrices used to convert investment by investor to investment by product bought or consumption by consumer categories to consumption by product categories used for productive categories.

3. Preliminary adjustments before RAS

It often happens in updating or making tables that one has better information about some cells than about others. For example, in updating the a table with a Glass row, we may have quite good information on the sales of glass products to Beer, because we have information on the production of glass beer bottles. In this case, we can simply remove the “relatively well-known flow” from both its row and column control, perform the RAS balancing on the remaining flows, and then put back in the known flow.

The problem with this procedure is that the “relatively well-known flows” tend to be big flows. If they are not quite consistent with the row or column controls, then removing them requires that all of this inconsistency should be attributed to changes in the remaining small flows. Thus, the small flows can be pushed about rather considerably. This problem can be reduced by a preliminary scaling of the relatively well-know flows before removing them from the process. To describe this adjustment, let \( R_i \) be the sum (in the base year) of the relatively well-known flows in row \( i \); \( S_i \), the sum of the other flows; and \( B_i \), the row control. Then let

\[
\alpha_i = \frac{R_i}{R_i + S_i}
\]

and define \( z_i \) as the solution of

\[
S_i z_i + R_i z_i^{\alpha_i} = B_i.
\]

The value of \( z \) which satisfies (10) is readily found by Newton’s method. We then scale all the relatively well-known flows by \( z_i^{\alpha_i} \) and all the other flow by \( z_i \). By (10), the row will then have the correct sum.
By (9), \(0 \leq \alpha_i \leq 1\). If \(\alpha < \alpha_i < 1\), then \(z^{\alpha_i}\) is closer to 1.0 than is \(z_i\); that is to say, the relatively well-known flows are scaled less than the other flows. They are however, scaled somewhat. If they account for a small fraction of the total of all flows in the row, they will be scaled but little; if they account for much of the row, they will be scaled almost as much as the other flows.

After this preliminary scaling, the known flows can be removed for the rest of the RAS process. While this scaling may seem a bit arbitrary, in practice it has given plausible results in many applications. In fact, it worked so well that the first person working with it, Thomas Reimbold, felt that the \(z\) must stand for Zauber, “magic” in his native language, and the procedure is therefore often referred to as the Zauber process.
Chapter 12

Trade and Transportation Margins and Indirect Taxes

1. Trade and Transportation Margins

A perennial problem in applied input-output analysis is the treatment of trade and transportation margins and of indirect taxes. The problem is nicely illustrated with transportation costs. If output is valued at the producer’s price — the price at the factory gate, so to speak — then the cost of transporting the goods to the user must be considered to be paid separately by the purchasing industry. Thus, the cost of the rail services used in hauling the coal used by electric power plants shows up as an input of rail transportation into electric generation. The cost of hauling generation equipment to and from the utilities’ repair facilities would appear in the same cell. Similarly, the cost of hauling coal to a steel mill and of hauling iron ore to the same mill will appear in the same cell.

The problems with this treatment are (1) it puts quite diverse activities into the same cell and (2) the table does not reflect the way the rail industry thinks about its business. It thinks in terms of products hauled — and prepares statistics on products hauled, not on industries to which it delivers. (Despite these problems, this treatment is the one most commonly followed.)

All of the problems apply with equal force to all the other transportation margins and to wholesale and retail trade margins.

One alternative is to change the measure of output of the industry to include the cost of delivering the product to the user. One disadvantage of this treatment is that it removes the numbers in the input-output table one step further from the numbers in terms of which people in the industry think, namely in producer prices. Another problem is that transportation margins may be very different for a dollar’s worth of product delivered to different users. The transportation cost of oil delivered to an electric utility by pipeline from a marine terminal may be very different from delivering by truck or rail to a small industrial user.

A better alternative is to add another dimension to the input-output tables. Thus, corresponding to each cell of the tables we have considered so far there would be a vector. The first entry in the vector would be the transaction in producer prices; the second entry would show the rail margin; the third, the truck margin; the fourth, the air freight; and so on through the wholesale and retail trade margins. In effect, we would have a table with layers, the first layer for the producer price transaction, the second for the rail margins, and so on. In fact, the benchmark tables for the United States are prepared with all this information. It has not been commonly used because the size of the matrices involved has been, until fairly recently, large relative to the power of the computers available. That constraint has now been effectively removed, and we may ask, How would we in fact compute with such a layered table?

If \( A \) represents the coefficient matrix in producer prices and \( T_i \) represents the \( i \)th layer of transportation and trade margin coefficients, then the fundamental input-output equations become

\[
q = Aq + \sum_i S_i T_i q = (A + \sum_i S_i T_i) q = f
\]
where $S_i$ is a matrix with 1's in the row which produces the service distributed by layer $i$ and elsewhere all zero. The matrix $\left( A + \sum_i S_i T_i \right)$ is, in fact, the matrix in producer prices with which it has been traditional to compute. What is gained by distinguishing the layers is not a correction of the traditional computations but rather a better description of what the flows are and a better basis for studying changes in coefficients in the $T_i$ matrices.

2. Indirect taxes, especially Value Added Taxes

Indirect taxes such as property taxes or franchise taxes are always and without problems treated as a component of value added, along with depreciation, profits, interest, and labor compensation. Excise taxes such as those on gasoline, alcohol, and tobacco are usually similarly treated, but with less justification, because some uses of these products are exempt. For example, gasoline used to power agricultural machinery or exported whiskey or cigarettes are exempt. Thus, these taxes should also be treated as a layer of the table, since they are not uniform for all cells. Retail sales taxes are usually treated as a component of value added by Retail trade. This treatment assumes that the tax is proportional to the retail margin in all products in all cells. In fact, there are different tax rates on different products, and some products are sold by retail establishments for intermediate use without retail sales tax.

The greatest problems, however, have probably been created by the value added tax (VAT) in the tables of countries which use this tax, a group that now includes all members of the European Union and numerous other countries. Producers pay VAT on the value of their sales but may deduct the VAT paid on their purchases. VAT is not charged on certain products, such as health services. Nor is it charged on exports. Many European input-output tables have been published in producer prices plus non-deductible VAT. That practice meant that the cell for paper products sold to the hotel industry did not contain VAT, because the VAT on those sales was deductible from the VAT owed by the hotels. The cell for paper products sold to hospitals, however, contained VAT, because the hospitals owed no VAT from which the VAT on the paper products could be deducted. Similarly, since households owe no VAT, they cannot deduct the VAT on the paper products they buy, so the VAT is included in the cell showing the sales of paper products to households. Thus, the cells in the paper products row of such a matrix have very diverse levels of VAT content. That means that the valuation of the product across the row is not homogeneous. It takes more wood pulp to make a dollar’s worth paper towels used by a hotel than to make a dollar’s worth of paper towels used by a hospital or household, because a significant portion of their dollar goes to VAT. This heterogeneity in the pricing in the row is obviously detrimental to the accuracy of the input-output calculations. The solution to the VAT problem is simply to create a VAT layer of the table.
1. The Problem

Makers of input-output tables often find data on inputs not by the product into which they went but by the industry that used them. An industry is a collection of establishments with a common principal product. But besides this principal product, any one of these establishments may produce a number of secondary products, products primary to other industries. Establishments classified in the Cheese industry may also produce ice cream, fluid milk, or even plastic moldings. Consequently, the Cheese industry may have inputs of chocolate, strawberries, sugar, plastic resins, and other ingredients that would appal a connoisseur of cheese. The inputs, however, are designated by what the product was, not by what industry made them. Similarly, data on the final demands, such as exports and personal consumption expenditure, is by product exported or consumed, not by the industry which made it. Thus, input-output matrices usually appear in two parts. The first part, called the Use matrix, has products in its rows but industries in its columns. The entries show the use of each product (in the rows) by each industry (in the columns.) The second, called the Make matrix, has industries in the rows and products in the columns; the entries show how much of each product was made in each industry.

How can we use these two matrices to compute the outputs of the various products necessary to meet a final demand given in product terms?

One way is to consider that each product will be produced in the various industries in the same proportion as in the base year of the table. This assumption is used, for example, in computable general equilibrium models based on social accounting matrices that explicitly show the Make and Use matrices. This assumption, however, can produce anomalous — not to say silly — results. In the above example, an increase in the demand for cheese would automatically and immediately increase demand for chocolate, strawberries, and sugar. That is nonsense. There must be a better way to handle the problem.

This highly unsatisfactory situation has led to efforts to make a product-to-product matrix. Indeed, the problem is so well recognized that the “Transmission programme of data” of the European system of accounts requires that all national statistical offices of the member states of the European Union transmit “symmetric” input output tables to Eurostat every five years. No real advice, however, is offered by Eurostat to the statistical offices on how to make these product-to-product tables. This paper offers a valuable tool for the process. (“Symmetric” is here intended to mean that the same concepts are used in both rows and columns. Its use as applied to these matrices is both highly confusing and not descriptive. Since it is the nature of the rows and columns that is the same, not their measure, symphysic would be both a better characterization and less confusing.)

To make such a matrix, we need to employ an additional assumption. There are basically two alternatives:
10. The product-technology assumption, which supposes that a given product is made with the same inputs no matter which industry it is made in.
11. The industry-technology assumption, which supposes that all products made within an industry are made with the same mix of inputs.

The System of National Accounts 1993 (SNA) reviews the two assumptions and finds (Section 15.146, p. 367) "On theoretical grounds, .... the industry technology assumption performs rather poorly" and is "highly
implausible." (Section 15.146, p 367) "From the same theoretical point of view, the product (commodity) technology model seems to meet the most desirable properties .... It also appeals to common sense as it is found a priori more plausible than the industry technology assumption. While the product technology assumption thus is favoured from a theoretical and common sense viewpoint, it may need some kind of adjustment in practice. The automatic application of this method has often shown results that are unacceptable, insofar as the input-output coefficients appear as extremely improbable or even impossible. There are numerous examples of the method leading to negative coefficients which are clearly nonsensical from an economic point of view." (Section 15.147)

Since 1967, the Inforum group has used a "semi-automatic" method of making "some kind of adjustment" in calculations based on the product-technology assumption, as called for by the SNA. We have used it with satisfactory results -- and without a single negative coefficient -- on every American table since 1958. The method was published in Almon 1970 and in Almon et al. 1974. Despite this long and satisfactory use of the method, it seems not to have come to the attention of the general input-output community. In particular, the authors of the section quoted from the SNA seem to have been unaware of it. The purpose of this note is to record the method where it is more likely to come to the attention of anyone working in input-output. At the same time, it expands the previous exposition with an example, provides a computer program in the C++ language for executing the method, and presents some of the experience of applying the method to the 1992 table for the USA.

2. An Example

An example will help us to visualize the problem. The Table 1 below shows the Use matrix for a 5-sector economy with a strong concentration in dairy products, especially cheese and ice cream.

Table 1. The Use Matrix

<table>
<thead>
<tr>
<th>USE Products</th>
<th>Industries</th>
<th>Cheese</th>
<th>Ice cream</th>
<th>Chocolate</th>
<th>Rennet</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheese</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ice cream</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chocolate</td>
<td>4</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rennet</td>
<td>14</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Other</td>
<td>28</td>
<td>72</td>
<td>30</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

We will call this matrix U. The use of chocolate in making cheese and rennet in making ice cream alerts us to the fact that the columns are industries, not products. (Rennet is a substance used to make milk curdle. It is commonly used in making cheese but never in ice cream.) The Make matrix, shown in Table 2 below, confirms that cheese is being made in the ice cream industry and ice cream in the cheese industry.
Table 2. The Make Matrix

<table>
<thead>
<tr>
<th>Industries</th>
<th>Products</th>
<th>Cheese</th>
<th>Ice cream</th>
<th>Chocolate</th>
<th>Rennet</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheese</td>
<td>70</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ice cream</td>
<td>30</td>
<td>180</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chocolate</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rennet</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>535</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>200</td>
<td>100</td>
<td>20</td>
<td>535</td>
<td></td>
</tr>
</tbody>
</table>

This matrix shows that of the total output of 100 of cheese, 70 was made in the Cheese industry and 30 in the Ice cream industry, while of the total ice cream output of 200, 180 was in the Ice cream industry and 20 in the Cheese industry. It also shows that, of the total output of 90 by the cheese industry, 78 percent (70/90 = .77778) was cheese and 12 percent ice cream. We will need the matrix, M, derived from the Make matrix by dividing each cell by the column total. For our example, the M matrix is shown in Table 3.

Table 3. The M Matrix

<table>
<thead>
<tr>
<th>M Cheese</th>
<th>Ice cream</th>
<th>Chocolate</th>
<th>Rennet</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheese</td>
<td>0.7</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Ice cream</td>
<td>0.3</td>
<td>0.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Chocolate</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Rennet</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Other</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Now let us suppose that, in fact, cheese is made by the same recipe wherever it is made and ice cream likewise. That is, we will make the "product-technology assumption." If it is true and the matrices made well, then there exists a "recipe" matrix, R, in which the first column shows the inputs into cheese regardless of where it is made, the second column shows the inputs into ice cream regardless of where it is made, and so on. Now the first column of U, U₁, must be .70*R₁ + .10*R₂, where R₁ and R₂ are the first and second columns of R, respectively. Why? Because the Cheese plants make 70 percent of the cheese and ten percent of the ice cream. In general,

\[ U = RM' \]  

where M' is the transpose of M. It is then a simple matter to compute R as

\[ R = U (M')^{-1}. \]

For our example, \((M')^{-1}\) is given in Table 4.

Table 4. M' Inverse
and $R$ works out to be

Table 5. The $R$ or “Recipe” Matrix

<table>
<thead>
<tr>
<th></th>
<th>Cheese</th>
<th>Ice cream</th>
<th>Chocolate</th>
<th>Rennet</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheese</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ice cream</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chocolate</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rennet</td>
<td>30</td>
<td>70</td>
<td>30</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

This $R$ is very neat. All the rennet goes into cheese and all the chocolate goes into ice cream. Unfortunately, as indicated by the quotation from the SNA, it is rare for the results to turn out so nicely.

Indeed, just a slight change in the $U$ matrix will show us what generally happens. Suppose that the $U$ matrix had been just slightly different, with 1 unit less of chocolate going into cheese as shown below and one less unit of rennet used in ice cream.

Table 6. An Alternative Use Matrix

<table>
<thead>
<tr>
<th>Alternative U</th>
<th>Cheese</th>
<th>Ice cream</th>
<th>Chocolate</th>
<th>Rennet</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheese</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ice cream</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chocolate</td>
<td>3</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rennet</td>
<td>15</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Other</td>
<td>28</td>
<td>72</td>
<td>30</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7 shows what the $R$ matrix would have been:

Table 7. An Impossible $R$ Matrix
Here we find the infamous small negative flows. It is not hard to see how they arise. While it is conceivable that the Cheese industry does not produce chocolate ice cream, it is also very easy for the table makers to forget to put into the Cheese industry the chocolate necessary for the ice cream it produces, or to put in too little. Wherever that happens, negatives will show up in the R matrix.

The negatives have driven at least some statistical offices to the industry-technology assumption. The so-called commodity-to-commodity matrix, C, derived from this assumption is

\[ C = UN', \]

where N is the matrix derived from the Make matrix by dividing each row by the row total. For example, the Cheese column of C is \( C_1 = 0.77778U_1 + 0.14285U_2 \) because 77.778 percent of the product of the first industry is cheese and 14.285 percent of the product of the second industry is cheese. The result of applying this assumption to our example is Table 8.

Table 8. The Mess Made by the Industry Technology Assumption

<table>
<thead>
<tr>
<th></th>
<th>Cheese</th>
<th>Ice cream</th>
<th>Chocolate</th>
<th>Rennet</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheese</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Ice cream</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Chocolate</td>
<td>-1.7</td>
<td>41.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Rennet</td>
<td>21.7</td>
<td>-1.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Other</td>
<td>30.0</td>
<td>70.0</td>
<td>30.0</td>
<td>5.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

This "solution" has made matters worse. The original U matrix had 4 units of chocolate going into the Cheese industry, which admittedly made some ice cream. Now this industry-technology product-to-product matrix asserts that 8.25 units of chocolate went into producing pure cheese! Not into the Cheese industry but into the product cheese! And 8.25 units of rennet went into producing curdled ice cream! To call the result a product-to-product table would be little short of scandalous.

Fortunately, we do not have to choose between this sort of massive nonsense and negative flows. It is perfectly easy to rely mainly on the product-technology assumption, yet avoid the negatives, as we will now show.
3. The No-Negatives Product-Technology Algorithm

We wrote the basic equation relating $U$, $M$, and $R$ as equation (1) above. It will prove convenient to rewrite equation (1) as

$$ U' = MR'. \quad (4) $$

Using $U'_i$ to denote the $i$th column of $U'$ and $R'_i$ to denote the $i$th column of $R$, we can write

$$ U'_i = MR'_i. \quad (5) $$

Notice that this is an equation for the distribution of product $i$ in row $i$ of the Use matrix as a function of $M$ and distribution of the same product in row $i$ of the $R$ matrix. We can simplify the notation by writing

$$ u = U'_i \quad \text{and} \quad r = R'_i; \quad (6) $$

then the previous equation becomes

$$ u = Mr \quad (7) $$
or

$$ \theta = -Mr + u \quad (8) $$

and adding $r$ to both sides gives

$$ r = (I - M)r + u. \quad (9) $$

Save in the unusual case in which less than half of the production of a product is in its primary industry, the column sums of the absolute values of the elements of $(I - M)$ are less than 1, and the convergence of the Seidel iterative process for solving this equation is guaranteed by a well-known theorem. (If the share of the total production of a particular product coming from the industry to which it is primary is $x$, then the absolute value of the diagonal of $(I - M)$ for that product is $| 1 - x |$ and the sum of all the absolute values of off-diagonal elements in the column is $|1 - x |$, so the total for the column is $2|1 - x |$, which is less than 1 if $x > .5$.) We start this process with

$$ r^{(0)} = u \quad (10) $$

and then define successive approximations by

$$ r^{(k+1)} = (I - M) r^{(k)} + u. \quad (11) $$

To see the economic interpretation of this equation, let us write out the equation for the use of a product, say chocolate, in producing product $j$, say cheese:

$$ x_j^{(k+1)} = u_j - \sum_{h=1}^{k} m_{j,h} x_h^{(k)} + \left(1 - m_{j,j}\right) x_j^{(k)} \quad (12) $$
The first term on the right tells us to begin with the chocolate purchases by the establishments in the cheese industry. The second term directs us to remove the amounts of chocolate needed for making the secondary products of those establishments by using our present estimate of the technology used for making those products, \( r(k) \). Finally, the last term causes us to add back the chocolate used in making cheese in other industries. The amount of chocolate added by the third term is exactly equal to the amount stolen, via second terms, from other industries on account of their production of product \( j \):

\[
(1 - m_{jj}) r_j^{(k)} = \sum_{h=1}^{n} m_{hj} r_j^{(k)}
\]

because

\[
\sum_{h=1}^{n} m_{hj} = 1.
\]

It is now clear how to keep the negative elements out of \( r \). When the "removal" term, the second on the right of (12), is larger than the entry in the Use matrix from which it is being removed, we just scale down all components of the removal term to leave a zero balance. Then instead of adding back the "total-stolen-from-other-industries" term, \((1 - m_{jj}) r_j \), all at once, we add it back bit-by-bit as it is captured. If a plundered industry, say Cheese, runs out of chocolate with only half of the total chocolate claims on it satisfied, we simply add only half of each plundering product's claim into that product's chocolate cell in the R matrix. We will call the situation where the plundered industry runs out of the product being removed before all claims are satisfied a "stop".

The process can also be applied to the rows of the value added part of the matrix. It is not certain, however, that the column sums of the resulting value-added table will match the value added as calculated from product output minus intermediate input. This value-added matrix will generally require RAS balancing to make it consistent with the product-to-product intermediate table.

4. When is it appropriate to use this algorithm?

This algorithm is appropriate where the product-technology assumption itself is at least approximately true. Essentially, it allows there to have been slightly different technologies in industries where assuming strictly the average product technology would produce negatives. It is appropriate where the negatives arise because of inexactness in making the tables or because of slight differences in technologies in different industries. Applied to the Use matrix of either Table 1 or Table 6, this method gives the "neat" Recipe matrix of Table 5 with no rennet in ice cream and no chocolate in cheese. It never produces negative entries nor positive entries where Use has a zero. The row totals are unaffected by the process. It is, moreover, equivalent to deriving Recipe from equation (1) if no negatives would arise, so that if the product-technology assumption is strictly consistent with the Use and Make tables, the method produces the true matrix. It may even produce a correct Recipe matrix from a faulty Use matrix — as it has perhaps done in our example — so that equation (1) could be used to revise the estimate of the Use matrix.

Certain accounting practices, however, may produce situations which appear to be incompatible with the product-technology assumption, even though the underlying reality is quite compatible. For example, local electric utilities generally buy electricity and distribute it. In the U.S. tables, they are shown as buying electricity (not coal), adding a few intermediate inputs and labor, and producing only a secondary product, electricity, which is transferred, via the Make matrix, back to electricity. Looked at mechanically, this
method of making electricity is radically different from that used in the Electricity industry, which uses coal, oil, and gas to make electricity, not electricity itself. If our algorithm is applied thoughtlessly to this situation, it cannot be expected to give very sensible results.

Fortunately, it is easy to generate signs of this sort of problem. One can compute the new Use matrix implied by equation (1) with the Recipe matrix found by the algorithm and the given Make matrix. This "NewUse" matrix can then be compared with the original Use matrix and the causes of the differences investigated. We will follow this procedure in next section on the experience of using the method on the 1992 tables for the USA.

To fix the problem in the above example about electricity, we have only to consider the output of the State and local utilities as production of their own primary product, which is then sold, via the Use matrix — not transferred via the Make matrix — to the Electricity industry. In essence, we use the industry technology assumption for the local electric utilities — and for all other industries where all of the output is secondary. The industry technology assumption may also be preferable for transfers to some catch-all sectors such as "Miscellaneous food preparations" (SIC2099), which includes such disparate products as vinegar, yeast, Chinese noodles, and peanut butter. It is probably just as reasonable to suppose that a product transferred into this industry is made with the average technology of the industry where it is made as with the average technology of this catch-all sector. Indeed, this sort of industry can produce the reverse of the negatives problem. For example, because of the importance of peanut butter in this industry, it has significant inputs of oil seeds. Now the no-negatives algorithm will not pull oil seeds out of the "Macaroni, spaghetti, vermicelli, and noodles" industry, (SIC2098), (which used no oil seeds) just because it transferred some Chinese noodles to 2099. But neither will it take out an adequate amount of flour for those noodles, because flour is quite unimportant in the 2099 input mix. This problem shows up only indirectly by substantial oil-seed inputs to many food industries in the NewUse which transferred products to 2099 but, in fact, used no oil seeds. That is a signal to switch to the industry technology for these transfers by converting them to sales in the Use matrix.

Thus, in the use of this method, a number of iterations may be necessary. Changes in concepts, in treatments of some transactions, and occasionally in underlying data may be necessary. Although the calculation of the non-negative Recipe matrix is totally automatic, it may be necessary to make several runs to get acceptable results.

In this process, it must be recognized that a nice, clean accounting system may not be operational, that is, it may not provide by itself a simple, automatic way to go from final demand vectors specified by products to total outputs of those products. We may have to change slightly some of the concepts in the accounting system to make it operational. In making the change required for the Electricity example, we have messed up the neat accounting concept of the Electricity column of the Use matrix as a picture of what came into a particular group of establishments. We have, however, taken a step toward creating what might be called an operational Use matrix. I do not say, therefore, that statistical offices should not produce pure accounting Use matrices. But I do feel that they should also prepare the operational use matrix and the final product-to-product matrix, for in the process, they will learn about and deal with the problems which the users of the matrix will certainly encounter. They may even discover and correct errors in their work before they are discovered by their users.

This process is totally inappropriate for handling by-products such as hides produced in the meat packing industry or metal scrap produced in machinery industries. Their treatment is a different subject.

5. A Brief History of the Negatives Problem
The idea to compute $R$ from equation (1) seems to have been first put in print by Van Rijckeghem (1967). He realized that there could be negatives but did not think they would be a serious problem. The idea of using equation (1) in this way, however, must have been in the air, for by early 1967, I had used it, without thinking that it was original, found negatives, and started work on the algorithm presented here.

The problem was encountered by ten Raa, Chakraborty and Small [1984] in the course of work which was primarily concerned with identifying by statistical means true by-products. They note the existence of the method presented here but write,

[Almon] iterates truncated Neumann series in which matrix multiplications are carried out only to a limited extent to avoid negatives. This arithmetic manipulation goes without justification, is arbitrary and depends on the choice of [make matrix]-decomposition as well as the iteration scheme.

I do not believe that any of this comment is correct. The Neumann series is the expansion $(I - A)^{-1} = I + A + A^2 + A^3 + \ldots$. The algorithm used here makes no use of this series; rather it uses the Seidel procedure. There are no matrix multiplications, nor is there any equivalence between a “limited” number of terms in the Neuman series and the Seidel solution. The procedure is carried to convergence. We have seen that the procedure has a perfectly reasonable economic interpretation; indeed, it arose from the economic interpretation of the Seidel procedure. The only thing perhaps "arbitrary" is that 0 is considered a reasonable input flow while negatives are considered unreasonable. I do not know what the "[make matrix]-decomposition" refers to, but I can assure the reader that the solution does not depend on the "iteration scheme." While I could not see how it could, given that it is carried to convergence, I changed the program and ran the "robberies" in the opposite order. The answers were identical.

The ingenious attempt of ten Raa [1988] to modify elements of the matrices in such a way as to find a most probable $U$ matrix consistent with a non-negative $R$ should be mentioned even though it ended, in the author’s view, in frustration.

Rainer and Richter [1992] have documented a number of steps which they took towards making what I have called here the operational Use and Make matrices. Such steps should certainly be considered and applied if need. These authors still ended up with hundreds of negative flows in the $R$ matrix because they were using just equation (1). At that point, the process described here could have been applied.

Steenge and Konijn [1992] point out that if the $R$ matrix computed from equation (1) has any negatives in it, then it is possible to change the levels of output of the various industries in such a way that more of all products is produced without using more of all inputs. They feel that it is implausible that such a rearrangement is possible and observe that perhaps the negatives "should not be regarded as rejecting the commodity technology assumption, but as indicators of flaws in the make and use tables." (p. 130). I feel that there is much merit in that comment. It seems to me that the right time and place to use the algorithm presented here is in the process of making the tables. If there are not good statistical grounds for preferring the original Use matrix, the recomputed NewUse might well be argued -- following the reasoning of Steenge and Konijn -- to be a better estimate.

The caveat here is that there may well be cases where it really would be possible to increase the outputs of all products while using less of some product. For example, if there are shoes made in the Plastics products industry without any use of leather, while the Footwear industry uses leather, then by moving shoe production from Footwear to Plastic products it may be possible to produce more of all products while using less leather. Where such cases arise, a different solution is necessary, for example, moving the shoes made in the Plastics
products industry together with their inputs into the Footware industry or insisting that the two kinds of shoes are separate if substitutable products.

6. Application to the U.S.A Tables for 1992

The method described here has been applied to all of the USA tables since 1958 with experiences broadly similar to those described here for the 1992 table. This table has 534 sectors, counting some construction sectors which have no intermediate sales. Of these 534, 425 have secondary production. Of the 283,156 possible cells in a 534 X 534 matrix, the Use matrix has 44,900 non-zero cells, and the Make matrix has 5,885. The matrix was produced in two versions. In one, certain activities, such as restaurant services of hotels, were removed from the industry where they were produced (Hotels) and put into the sector where these activities were primary (Restaurants). In the other, these activities were left in the industry where they were conducted. The first version was designed to make the product-technology assumption more valid, and it has been used here. The matrix also puts true by-products (such as hides from meat packing) in a separate row, not one of the 534 considered here.

To try to convey a feeling of what it is like to work with the algorithm, we will look at the process midway along, rather than at the very beginning or the somewhat polished end. That is, some adjustments in the Use and Make matrix from which the algorithm starts will have already been made. As a result of this application, further adjustments will be suggested before the next application.

Before this application of the algorithm, the output of industries which had only secondary production had been changed, for reasons explained above, to be primary and the flows moved from the Make to the Use matrix.

In the following rather detailed descriptions, necessary to give a picture of what the process is really like, I will, to avoid confusion, capitalize the first letter of the first word in industry names but not in product names.

The industry Water and sewer systems failed to satisfy the requirement that at least half of the output of a product should be in the industry where it is primary. Indeed, some 85 percent of this product’s output comes from Other state and local enterprises, and the iterative procedure failed to converge for a few rows until this secondary transfer was converted into a primary sale. Production of secondary advertising services, which occurred in many sectors, was also converted to a primary product of the producing industry and "sold" via the Use matrix to the Advertising industry. Secondary production of recreational services in agricultural industries was similarly converted. Much of the output of the several knitting industries had been treated originally as secondary production, and these had been changed to primary sales before the calculations shown here. Finally, the diagonals of many columns of the Use matrix are large, in part because intra-firm services, such as those of the central offices, often appear there. Thus the same sort of service that is on the diagonal of industry i is also on the diagonal of industry j. In this case, the product-technology assumption does not apply, not because it is untrue, but because of the way the table was made. Until we are able to obtain tables without this problem, we have just removed half of the diagonals from the Use table before calculating Recipe, and have then put back this amount in both of these matrices and in the NewUse matrix.

The data in both Use and Make tables were given to the nearest 1 million dollars, and all dollar figures cited here are in millions. The convergence test in the iterative process was set at one tenth of that amount, .1 million dollars. The iterative process converged for most rows of the R matrix in less than five iterations. The most iterations required for any row was 15.
The resulting Recipe matrix looks very similar in most cells to the original Use table. The Recipe matrix contains, of course, only non-negative entries and can have strictly positive entries only where U has positive entries. It may, however, as a result of the "robbing" process, have a zero where U has a positive entry. In all, there were only 95 cells in which Recipe had a zero where Use had a positive entry.

Although it is the Recipe matrix that we need from this process, it is also interesting, as noted above, to compare the original Use matrix with what we may call NewUse, computed by the equation $1$ by $\text{NewUse} = \text{Recipe} \times \text{Make}$. The difference between Use and NewUse shows the changes in the Use matrix necessary to make it strictly compatible with product-technology assumption, the given Make matrix, and the calculated Recipe matrix. If there was no "stop" in a row, the two matrices will be identical in that row. There were 118 such identical rows, 109 of them having no secondary output.

In the other rows, these differences turn out to be mostly small but very numerous. The first and most striking difference is that NewUse has almost twice as many non-zero cells as does Use. Nearly all of these extra non-zeros are very small, exactly the sort of thing to be reasonably ignored in the process of making a table. But it is precisely this "reasonable ignoring" that leads to the problem of many small negatives in the product-to-product tables calculated without the no-negatives algorithm.

To get a closer look at how Use and NewUse compare, we may first divide each column by corresponding industry’s output and then look at the column sums of the absolute values of the differences of individual coefficients in the column. This comparison is shown in Table 9. Clearly the vast majority of industries show only small differences compatible with “reasonable ignoring” of small flows in the Use matrix. They, therefore, cast no serious doubt on the product-technology assumption or the usability of the Recipe matrix obtained by the no-negatives algorithm. If what we are interested in is the R matrix, we can ignore the small differences between Use and NewUse.

### Table 9. Comparison of Use and NewUse

<table>
<thead>
<tr>
<th>Sum of Absolute Differences</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>.050 - .250</td>
<td>17</td>
</tr>
<tr>
<td>.030 - .050</td>
<td>24</td>
</tr>
<tr>
<td>.020 - .030</td>
<td>54</td>
</tr>
<tr>
<td>.010 - .020</td>
<td>117</td>
</tr>
<tr>
<td>.000 - .010</td>
<td>312</td>
</tr>
</tbody>
</table>
There are, however, a few cases that should be looked at more closely. Table 10 shows a list of all of industries which had a sum of absolute differences greater than .050. We will look at the top five.

For Asbestos products, the cause of the difference is quickly found. The fundamental raw material for these products comes from industry 31 Misc. non-metallic minerals. Over forty percent of the output of asbestos products, however, is produced in industry 400 Motor vehicle parts and accessories, but this industry buys neither miscellaneous non-metallic minerals nor asbestos products. In other words, it seems to be making almost half of the asbestos products without any visible source of asbestos. This anomaly seems to me to be an oversight in making the Use matrix which should be simply corrected. If our only interest is the Recipe matrix, the algorithm seems to have computed pretty nearly the right result from the wrong data. On the other hand, if we want to correct the Use table, NewUse, gets us started with the right entry for Misc. non-metallic minerals into both Motor vehicle parts and Asbestos products. To keep the right totals in these two columns of Use will require manual adjustments.

The second largest difference between Use and NewUse shown in Table 10 is in the input of meat animals into Sausage. The Sausage industry is shown in the Use matrix to buy both animals ($655) and slaughtered meat ($9688). It had a primary output of $13458 and a secondary output of $2612 of products primary to Meat packing. Meat packing had a secondary output of $4349 of sausage. Now in Meat packing, the cost of the animals is over eighty percent of the value of the finished product, so the purchases of animals in the Sausage industry is insufficient to cover even the secondary meat output of this industry, not to mention making any sausage. In making Recipe, the input of animals directly into sausage is driven to zero and cut off there rather than being allowed to become negative. Then when NewUse is made, the direct animal input for all the secondary production of meat packing products is put in, thus making a flow some six times as large as the purchase of meat animals by the Sausage industry in the original Use matrix.
What I believe to be really happening here is that Sausage plants are mostly buying halves of slaughtered animals from meat packers, selling off the best cuts as a secondary product, and using the rest to make sausage. Over in the Meat packing plants, the same thing is happening. Fundamentally, there is only one process of sausage making. The question is how to represent it in the input-output framework. The simplest representation of it in the Use matrix would be to have packing houses sell to sausage plants only the meat that would be directly used in sausage. The rest, the choice cuts sold off as meat by Sausage mills, would simply be considered sold by the packers without ever passing through the Sausage mills. The industry output of Sausage mills is reduced but cost of materials (namely, meat) is reduced by exactly the same amount, so there is no need to adjust other flows. Product output of meat is reduced, but not the industry output. Thus, a slight adjustment in the accounting makes it broadly compatible with the product-technology assumption. The seventh item in Table 10, by the way, is just the other side of this problem.

The third largest of the discrepancies lies in row 16, oil-bearing crops, of industry 125 Vegetable oil mills n.e.c (not elsewhere classified). The differences in the underlying flows is not large, $298 in Use and $251 in NewUse, but it turns up in Table 10 because the cost of these oil crops is such a large fraction of the output of the Vegetable oil mills. A comparison of the oil-bearing crops row of Use and NewUse shows that NewUse has a number of small positive entries for industries where, as for Cheese, Use has a zero and where, moreover, it is highly implausible that there was any use of oil seeds. On the other hand, most of the large users of oil seeds, like Vegetable oil mills have had their usage trimmed back. The key to what is going on is found in industry 132 Food preparations n.e.c.. In Use, this industry bought $558 from oil bearing crops, nearly twice the consumption of the vegetable oil mills themselves. Peanut butter, as noted above, is in this catchall industry. That fact, by itself, is not a problem. The problem is that about a quarter of the production of products primary to this industry are made in other industries. In fact, most of the food manufacturing industries have some secondary production of the miscellaneous food preparations. Probably "preparations" made in the Cheese industry are quite different from those made in the Pickles industry. And it certainly makes no sense to spread oil seed inputs all over the food industries. Here we have a clear case of the inapplicability of the product-technology assumption if all these secondary products are considered to be truly the same product. On the other hand, as argued above, the very heterogeneity of the products makes it appropriate to consider each as a primary product of the industry which produces it and then "sell" it, via the Use matrix, to Food preparations for distribution. In the next pass at making Recipe, this change is to be made.

The vegetable oil industries also present another interesting case of apparent but perhaps not real violation of the product-technology assumption, which shows up in the fifth item in Table 10. Industry 125 Vegetable oil mills n.e.c. has inputs of oil-bearing crops, cotton, and tree nuts totaling $437. It uses these oil sources to produce a primary output of $572. Industry 128 “Edible fats and oils” produces $92 of products primary to 125 without a penny of any of these inputs! Surely this is flat violation of the product-technology assumption. But is it really? "Edible fats and oils" buys lots of the products primary to Vegetable oil mills. Thus, it is entirely possible to have two bottles of chemically identical oil made of identical raw materials by identical refining processes but with one bottle made entirely in Vegetable oil mills while the oil in the other bottle was pressed in those mills and then sold to Edible fats and oils for finishing. We might call this situation “trans-market product technology.” Our algorithm gave the right answer for the Vegetable oil mills column of Recipe, that is, it combined output of products primary to the oil mills with the inputs of oil sources which this industry had.

The fourth largest discrepancy in Table 10 is for the gasoline input into Automobile renting and leasing. Use shows $1131; Recipe ups that to $1197.2; but NewUse cuts it back to $565.5. What happened? The problem is that slightly more than half of the output Auto renting is produced in Credit agencies, with a minuscule input of gasoline. When NewUse is made, more than half of the gasoline in Recipe is allocated over to Credit
agencies. Here we are confronted with a failure of the product-technology assumption not because of different processes for producing the same product but because two quite different products have been called one and the same in the accounting system. The output of the Credit agencies, long-term leasing, is quite distinct from the short-term renting, which is were the gasoline was used. The best solution would be to recognize the difference of the two products. Short of that, the worst of the problem can be fixed by turning the secondary transfer from Credit agencies to Automobile rental into a primary flow. The present Recipe matrix, incidentally, is about right in the gasoline row but makes no connection between a final demand for automobile renting and leasing and the output of credit agencies.

From these five or six cases, we see that our algorithm cannot be expected to give usable results on the first try. The problems are likely to lie, however, neither in the fundamental economic reality nor in the algorithm, but in an accounting system which needs a few modifications in Use and Make to make it operational in our sense. Most importantly, the algorithm gives us the means to identify the places that need attention and a way of progressing systematically through the problems. It also provides a way of producing a final, non-negative Recipe matrix that implies a NewUse matrix close enough to the modified Use matrix that the differences can be safely ignored.

Making an input-output table requires fussing over details, and making a good Recipe matrix with the algorithm presented here is no different in this respect from any other part of the process. Use of the algorithm reveals and pinpoints problems. Moreover, the important problems are likely to be small in number. We have covered all of those causing a difference of as much as .100 between columns of Use and NewUse. To get to a Recipe table we would be ready to accept might require another week’s work. But in the total effort which went into making this table, that is minuscule. Most importantly, the use of the algorithm gives us a way to work on the problems rather than just wring our hands over negatives.

In this sense, this algorithm has performed satisfactorily over many years on every U.S. table since 1958. The use of the method seems to me to deserve to become a standard part of making input-output tables and, in particular, for making product-to-product tables.

7. The Computer Program

The C++ code for this algorithm, using functions from BUMP, the Beginner’s Understandable Matrix Package, for handling matrices and vectors, is given below. It is reproduced here because the code shows more clearly than the verbal or formulaic description exactly what is done. The program and the supporting BUMP code made be downloaded from the Inforum Internet site: www.inforum.umd.edu. The main program here reads in the matrices that were used in the examples. The main program for the actual calculations of the full-scale American matrices is significantly larger and has various diagnostic output, such as that shown in Table 10. It is available on request.

In using the algorithm, it is important for documenting what has been done to have a method of input of the original Use and Matrix matrices that preserves the original version at the top of the input file and introduces the modifications as over-rides later in the file. It is also important to have software, such as ViewMat, which will show corresponding columns of several large matrices side-by-side in a scrolling grid. ViewMat is also available on the Inforum Internet site.

```c++
#include <stdio.h> // for printf();
#include <math.h> // for abs()
#include "bump.h"
int purify(Matrix& R, Matrix& U, Matrix& M, float toler);
```
void main()
{
    Matrix Use(5,5), Make(5,5), R(5,5), NewUse(5,5);
    Use.ReadA("Use.dat");
    Make.ReadA("Make.dat");
    purify(R, Use, Make, .000001);
    R.Display("This is R");
    writemat(R,"Recipe");
    NewUse = R*~NewUse;
    writemat(NewUse,"NewUse");
    tap();
    printf("\nEnd of calculations.\n");
}

/* Purification produces a product-to-product (or Recipe) matrix R from a Use matrix U and a Make matrix M. M(i,j) shows the fraction of product j made in industry i. U(i,j) shows the amount of product i used in industry j. The product-technology assumption leads us to expect that there exists a matrix R such that U = RM'. If, however, we compute R = U*Inv(M') we often find many small negative elements in R. This routine avoids those small negatives in an iterative process. */

int purify(Matrix& R, Matrix& U, Matrix& M, float toler){
    int row, i, j, n, iter, imax;
    const maxiter = 20;
    float sum, rob, scale, dismax, dis;
    n = U.rows(); // n = number of rows in U
    m = U.columns(); // m = number of columns in U
    Vector C(m), P(m), Flow(m), Discrep(m);
    // Flow is row of U matrix and remains unchanged.
    // P becomes the row of the purified matrix.
    // C is the change vector at each iteration.
    // At the end of each iteration we set P = Flow + C, to start // the next iteration.
    // Purify one row at a time
    for(row = 1; row <= n; row++){
        C.set(0.); // C, which will receive the changes, is
        // initialized to zero.
        // P = Flow + C will be the new P.
        pulloutrow(Flow, U, row);
        P = Flow;
        iter = 0;
        start: iter++;
        for(j = 1; j<=m; j++){
            sum = 0;
            for(i = 1; i <= m; i++){
                if(i == j) continue;
                rob = P[i]*M(j,i);
                sum += rob;
                C[i] += rob;
            }
            // Did we steal more from j than j had?
            if((sum > Flow[j] & & sum > 0){
                // scale down robbery
                scale = 1. - Flow[j]/sum;
                for(i = 1; i <= m; i++){
                    if(i == j) continue;
                    C[i] -= scale*P[i]*M(j,i);
                }
            }
            sum = Flow[j];
            C[j] = sum;
        }
        // Check for convergence
        imax = 0;
        dismax = 0;
        for(i = 1; i <= m; i++){
            dis = fabs(P[i] - Flow[i] - C[i]);
            Discrep[i] = dis;
            if(dis >= dismax){
                imax = i;
                dismax = dis;
            }
        }
    }
    return 0;
}
P = Flow + C;
C.set(0);
if(dismax > toler){
    if(iter < maxiter) goto start;
    printf("Purify did not converge for row \%d. Dismax = \%7.2f. Imax = \%d.\n", row, dismax, imax);
}
putinrow(P,R,row);
return(OK);
References


The *System of National Accounts 1993* (published by the United Nations, the World Bank, the IMF, the OECD, and the European Union)


Chapter Modeling Personal Consumption Expenditure

Long-term, multisectoral modeling requires calculation of consumer expenditures in some detail by product. Finding a functional form to represent the market demand functions of consumers for this work has proven a surprisingly thorny problem. Clearly, the form must deal with significant growth in real income, the effects of demographic and other trends, and changes in relative prices. Both complementarity and substitution should be possible among the different goods. Increasing income should certainly not necessarily, by the form of the function, force the demand for some good to go negative. Prices should affect the marginal propensity to consume with respect to income, and the extent of that influence should be an empirical question, not one decided by the form of the function.

This paper will present a form which meets these requirements and extends a form suggested twenty years ago by Almon [1979]. Applications of the new form to forty-product demand systems for France, Italy, Spain and the United States are reported and the results compared.

Before presenting this form, however, it may be well to see just how tricky it can be to find a form with these simple requirements by looking at another form, the "Almost Ideal Demand System" (AIDS) suggested by Deaton and Muellbauer [1980]. Its name, the eminence of its authors and its place of publication have led to wide usage. It has, however, a most peculiar property which is likely to sink any growth model in which it is used. Like many others, it is derived from utility maximization; its problems will therefore emphasize the important fact that such derivation does not automatically imply reasonable properties. One of the properties it does imply, however, is Slutsky symmetry in the market demand functions. This property was not mentioned above. Should it have been? What role should this symmetry play in market demand functions? This question also needs to be examined before presenting the new form, for it plays a key role in its formulation.

1. Problems and lessons of the AIDS form.

The AIDS form can be written as an equation for the budget share of good i:

\[ s_i = a_i + \sum_{j=1}^{n} d_{ij} \cdot \log(p_j) + b_i \cdot \log(y/P) \]  \hspace{1cm} (68)

where \( s_i \) is the budget share of product \( i \), \( p_j \) is the price of product \( j \), \( y \) is nominal income and \( P \) is an overall price index, the matrix of \( d \)'s is symmetric and has zero row and column sums, the sum of all the \( a_i \) is 1, and the \( b_i \) sum to zero. Consequently, if any \( b_j \) is positive, then one or more others must be negative. Thus increasing real income must ultimately drive the consumption of one or more goods negative, unless, of course, it has no effect at all on budget shares. This property seems rather less than "ideal". Moreover, the partial derivative of the share with respect to real income is independent of the relative prices, whereas common sense suggests that it should depend on them. Because of these properties, the AIDS form, while possibly "almost ideal" from some point of view, is surely absolutely inadequate for use in any growth model. Since it is derived from utility maximization, it also serves, as already said, as a clear warning that the mere fact of such ancestry is no assurance whatsoever of the adequacy of a form. Perhaps there is also in the AIDS story a lesson for modesty in naming a form, a lesson which has been heeded in naming the "PADS" form proposed here.
A number of other forms derived from utility maximization were reviewed in the Almon cited and found wanting relative to the simple properties set out above. The only study which to my knowledge has estimated these forms, AIDS, and the Almon form all on the same data and compared the results is Gauyacq [1985]. Using French data for 1959-1979, he estimated "the linear expenditure system of Stone; the model with real prices and income of Fourgeaud and Nataf; the additive quadratic model of Houthakker and Taylor; the logarithmically additive model of Houthakker, ... the Rotterdam model of Theil and Barten, the Translog model based on a logarithmic transformation of the utility function; the AIDS model of Deaton and Muellbauer; ... [and] the model proposed by Clopper Almon." The conclusion was not surprising to anyone who had compared the properties of the forms to the simple requirements stated above: "De l'étude que nous avons effectué, il apparaît en définitive que seul le modèle de C. Almon constitue un système que satisfasse approximativement aux attendus théoriques et présente un réel intérêt pour l'étude économétrique de fonctions de demande détaillées." (p. 119). (From the study which we have done, it appears that definitely only the model of C. Almon offers a system which satisfies approximately theoretical expectations and is of real interest for the econometric study of detailed demand functions.) Elegant theoretical derivations, apparently, are of little help in finding adequate forms. Despite this relative success, there is a problem with the Almon suggestion, as we will see in section 3, where we will also see a way to fix it.

2. Slutsky Symmetry and Market Demand Functions

Just about the only non-obvious implication of the theory of the single consumer who maximizes utility subject to a budget constraint is the Slutsky symmetry shown in equation (2).

$$\frac{\partial x_i^k}{\partial p_j} + \frac{\partial x_i^k}{\partial y_k} \cdot x_j^k = \frac{\partial x_j^k}{\partial p_i} + \frac{\partial x_j^k}{\partial y_k} \cdot x_i^k$$

(69)

Here $x_i^k$ is the consumption of product i by individual k, $y_k$ is the nominal income of individual k, and $p_j$ is the price of product j. A comparable relation, however, need not hold for the market demand functions, the sum over all k of individuals' demand functions. Summing the above equation over the individuals gives equation (3),

$$\frac{\partial \sum_k x_i^k}{\partial p_j} + \sum_k \frac{\partial x_i^k}{\partial y_k} \cdot x_j^k = \frac{\partial \sum_k x_j^k}{\partial p_i} + \sum_k \frac{\partial x_j^k}{\partial y_k} \cdot x_i^k$$

(70)

which is in general not the same as -- and does not imply -- equation (4),

$$\frac{\partial \sum_k x_i^k}{\partial p_j} + \frac{\partial \sum_k x_i^k}{\partial \sum_k y_k} \cdot \sum_k x_j^k = \frac{\partial \sum_k x_j^k}{\partial p_i} + \frac{\partial \sum_k x_j^k}{\partial \sum_k y_k} \cdot \sum_k x_i^k$$

(71)

which is what Slutsky symmetry of the market demand functions would imply. Thus, strict micro theory does not imply Slutsky symmetry of market demand functions. Consequently, there is in general no "representative consumer." To suppose that market demand functions derived by maximizing the utility of this non-existent entity have "micro foundations" not enjoyed by functions not so derived is hardly respectful of micro theory. Rather, any market demand functions so derived are on exactly the same theoretical footing as market demand functions made up without any reference to utility maximization. Both kinds of functions must meet the same "adequacy" criteria.
With that point clearly established, we may, however, ask Are there restrictive conditions under which equation (3) would imply equation (4)? One condition is, of course, that all individuals should have not only the same utility function but also the same income, and that the increase in aggregate income is accomplished by giving each the same increase. That condition is hardly interesting for empirical studies. A less restrictive condition is that the marginal propensity to consume a given product with respect to income should be the same for all individuals, or in effect, that the Engel curves for all products should be straight lines. If, for example,

$$\frac{\partial x^k_i}{\partial y^k_j} = a_i$$  (72)

then the second term on each side of equation (3) can be factored to yield

$$\frac{\partial \sum_k x^k_i}{\partial p_j} + a_j \cdot \sum_k x^k_j = \frac{\partial \sum_k x^k_j}{\partial p_j} + a_j \cdot \sum_k x^k_j$$  (73)

This is exactly what equation (4) states, for in this case it makes no difference to whom the "infinitesimal" increase in income is given and

$$\frac{\partial \sum_k x^k_i}{\partial \sum_k y^k_j} = a_i.$$  (74)

Now the assumption that all Engel curves are straight lines is generally contradicted by cross-section budget studies, even when one uses total expenditure in place of income in the Engel curves. (See, for example, Chao [1991] where Figure 2.2 shows Engel curves for 62 products). On the other hand, many products have virtually straight Engel curves over a considerable middle range of total expenditure where most households find themselves. Thus, one gets the impression that while Slutsky symmetry is certainly not a necessary property of market demand curves, it probably does no great violence to reality to impose symmetry to reduce the number of parameters to be estimated.

### 3. A Perhaps Adequate Form

The 1979 Almon article introduced a form with a multiplicative relation between the income terms and the price terms. Its general form is:

$$x_i(t) = (a_i(t) + b_i(y/P)) \prod_{k=1}^n p_{ik}^c$$  (75)

where the left side is the consumption *per capita* of product *i* in period *t* and *a_i(t)* is a function of time. The *b_i* is a positive constant. The *y* is nominal income *per capita*; *p_i* is the price index of product *k*; *P* is an overall price index defined by

$$P = \prod_{k=1}^n p_k^s$$  (76)

where *s_k* is the budget share of product *k* in the period in which the price indexes are all 1, and the *c_{ik}* are constants satisfying the constraint
\[ \sum_{k=1}^{n} c_{ik} = 0. \]  

(77)

Any function of this form is homogeneous of degree 0 in all prices and income and satisfies all of the properties set out in the first paragraph. It has three problems:

1. It is not certain that expenditures will add up to income.

2. There is no way to choose the parameters to guarantee Slutsky symmetry at all prices if we want to. We can, however, arrange to have symmetry in some particular base period. As long as the shares of various products in total expenditure do not change very much from those of that base period, we will continue to have approximate symmetry.

3. There are a lot of c's to be estimated.

Problem 1 can be easily fixed by adding on a "spreader," that is, by summing all expenditures, comparing them with \( y \), and allocating the difference in proportion to the marginal propensities to consume with respect to \( y \) at the current prices. The amount to be spread is usually small and the form with a spreader has essentially the same properties as the form without, plus the adding up property. We need not complicate the mathematics here by adding the spreader, but in practice it should be added when the equations are used in modeling.

Problem 2, in view of section 2, is more a cautionary note than a real problem. Symmetry in a base year is probably quite adequate.

Problem 3 -- which occurs in all forms which provide for varying degrees of substitution and complementarity -- can be quite severe. If we have 80 categories of expenditures, we have 6,400 c's less the 80 determined by equation (10). If we have 20 years of annual data, we have 1,600 data points from which to determine these 5,600 parameters, or 3.5 parameters per data point! Clearly, we have to have employ some restrictions. Even if we had only one parameter per data point, we would probably want restrictions to insure reasonableness of the parameters. Indeed, the principal theoretical problem in consumption analysis is to find ways to specify what is "reasonable."

Part of the solution of problem 3 can be found, if we wish, in the point noted in problem 2, namely that we can impose Slutsky symmetry at some prices. The Slutsky condition may be derived either from equation (2) or, more simply, by assuming that the compensating change in income is that which keeps \( y/P \) constant. Either approach gives as the symmetry condition equation (11):

\[ c_{ij}/s_j = c_{ji}/s_i \]  

(78)

Multiplying both sides by \( pp/y \) gives equation (12).

\[ c_{ij}/s_j = c_{ji}/s_i \]  

(79)

If we then define

\[ \lambda_{ij} = c_{ij}/s_j \]  

(80)

then the form can be written as
The problem with this form was that products which had no natural partners with which to form a group all ended up either in very strange groups or, if they were given no group at all, all with nearly the same own price elasticity, namely $\lambda_0$. It is often difficult to find groups for such goods as Telephone service, Medical service, Education, or Religious services. A specification which forces them all to have, for that reason, nearly the same own price elasticity is certainly inadequately flexible.

An adequate form, it now seems, should allow every product to have its own own-price elasticity. We will then have as many price-exponent parameters as there are products plus groups plus subgroups. A simple way to achieve this generalization is to introduce $n$ parameters, $\lambda_1, \ldots, \lambda_n$, and use them to define the $\lambda_{ij}$ as follows. If $i$ and $j$ are not members of the same group or subgroup, then

$$
\lambda_{ij} = \lambda_{ji}.
$$

This restriction cuts the number of parameters by a half. That reduction is a big help but is clearly insufficient.

$$
\mu_{ij} (t) = (a_t (t) + b_t (y/P)) \prod_{k=1}^K \frac{\lambda_{ik} \gamma_k}{y_k}
$$

where

$$
\lambda_{ij} = \lambda_{ji}.
$$

Further help with this problem can be found through the idea of groups and subgroups of commodities. The side box shows an example with fifteen basic commodity categories. These are subdivided into three groups and several categories which are not in any group. The first group is divided into two subgroups; the second, into one subgroup and a category not in the subgroup; the third group has no subgroup.

An adequate form, it now seems, should allow every product to have its own own-price elasticity. We will then have as many price-exponent parameters as there are products plus groups plus subgroups. A simple way to achieve this generalization is to introduce $n$ parameters, $\lambda_1, \ldots, \lambda_n$, and use them to define the $\lambda_{ij}$ as follows. If $i$ and $j$ are not members of the same group or subgroup, then

$$
\lambda_{ij} = \lambda_{ji}.
$$

This restriction cuts the number of parameters by a half. That reduction is a big help but is clearly insufficient.

Further help with this problem can be found through the idea of groups and subgroups of commodities. The side box shows an example with fifteen basic commodity categories. These are subdivided into three groups and several categories which are not in any group. The first group is divided into two subgroups; the second, into one subgroup and a category not in the subgroup; the third group has no subgroup.

Further help with this problem can be found through the idea of groups and subgroups of commodities. The side box shows an example with fifteen basic commodity categories. These are subdivided into three groups and several categories which are not in any group. The first group is divided into two subgroups; the second, into one subgroup and a category not in the subgroup; the third group has no subgroup.
\[ \lambda_{ij} = \lambda_i + \lambda_j \] (83)

while if they are in the same group, G, \( \lambda_{ij} = \lambda_i + \lambda_j + \mu'_G \), and if they are in the same subgroup, g, of the group G, \( \lambda_{ij} = \lambda_i + \lambda_j + \mu'_G + \nu'_g \). The definitions apply only for i not equal to j. The \( \lambda_{ii} \) are each determined by equation (10), the homogeneity requirement.

Using these definitions, for product i, a member of group G and subgroup g, the equation becomes

\[ x_i(t) = (a_i(t) + b_i(y/P)) \prod_{k=1}^n p_k^{(\lambda_k' \lambda_k) s_k} \prod_{k \in G, k \neq i} p_k^{s_k' \mu'_G} \prod_{k \in g, k \neq i} p_k^{s_k' \nu'_g} \cdot p_i^{c_{i1}} \] (84)

Equation (10) requires

\[ \sum_{k=1}^n \lambda_k s_k + \lambda_i \sum_{k \neq i} s_k + \mu'_G \sum_{k \in G, k \neq i} s_k + \nu'_g \sum_{k \in g, k \neq i} s_k + c_{i1} = \] (85)

If we solve this equation for \( c_{i1} \) and substitute in equation (17), we obtain, after a bit of simplification,

\[ x_i(t) = (a_i(t) + b_i(y/P)) (p_i/P) \lambda_i \prod_{k=1}^n (p_k/P)^{\lambda_k s_k} \left( \prod_{k \in G} (p_k/P)^{s_k} \right)^{\mu'_G} \left( \prod_{k \in g} (p_k/P)^{s_k} \right)^{\nu'_g} \] (86)

where we have inserted the terms involving \( p_i/p_k \) into all of the products, because this term is always 1.0 no matter to what power it is raised. We can make the form even simpler by introducing price indexes for the group G and subgroup g defined by

\[ P_G = \left( \prod_{k \in G} p_k^{s_k} \right)^{1/\sum_{k \in G} s_k} \text{ and } P_g = \left( \prod_{k \in g} p_k^{s_k} \right)^{1/\sum_{k \in g} s_k} \] (87)

We then obtain simply equation (21)

\[ x_i(t) = (a_i(t) + b_i(y/P)) \cdot \left( \frac{p_i}{P} \right)^{-\lambda_i} \prod_{k=1}^n \left( \frac{p_k}{p_i} \right)^{-\lambda_k s_k} \cdot \left( \frac{p_i}{P_g} \right)^{-\mu'_G} \left( \frac{p_i}{P_g} \right)^{-\nu'_g} \] (88)

where

\[ \mu = \mu' \sum_{k \in G} s_k \text{ and } \nu = \nu' \sum_{k \in g} s_k \] (89)

This is the form for estimation. Note that it has one parameter, a \( \lambda \), for each good, plus one parameter, a \( \mu \), for each group, plus one parameter, a \( \nu \), for each subgroup. Thus, it appears to have an adequate number of parameters. The Slutsky symmetry of (21) at the initial prices and income may be verified directly by taking partial derivatives of (21).

A special case of historical interest arises when all the \( \lambda_i \) are the same and equal to \( \lambda_o/2 \), for in that case equation (21) simplifies to
\[ x_i(t) = \left( a_i(t) + b_i(y/P) \right) \left( \frac{P_i}{P} \right)^{-\lambda_i} \left( \frac{P_i}{P_g} \right)^{-\mu_i} \left( \frac{P_i}{P_g} \right)^{-\nu_i} \]  

(90)

which is exactly the form suggested in the Almon [1979] article. Thus, the present suggestion is a simple generalization of the earlier one.

In practice, there are apt to be a few commodities, such as Tobacco, Sugar, or Medical care which show so little price sensitivity that they cannot be fit well by this system. For them, we will assume that all the \( \lambda_i \) in their rows and columns are 0. Note that this assumption is perfectly consistent with the symmetry of the lambda's. When there are such "insensitive" commodities in the system, equation (21) is modified in two ways. For these items, there are no price terms at all, while for other items the product term which in (21) is shown with \( k \) running from 1 to \( n \) is modified so that \( k \) runs only over the "sensitive" and not the "insensitive" commodities.

It is useful in judging the reasonableness of regression results to be able to calculate the compensated own and the cross price elasticities. ("Compensated" here means that \( y \) has been increased so as to keep \( y/P \) constant.) Their derivation is straight-forward but complicated enough to make the results worth recording. In addition to the notation already introduced, we need

- \( u_{ij} \) = the share in the base year of product \( j \) in the group which contains product \( i \), or 0 if \( i \) is not in a group with \( j \).
- \( w_{ij} \) = the share in the base year of product \( j \) in the subgroup which contains product \( i \) or 0 if \( i \) is not in a subgroup with \( j \).
- \( \mu_i \) = the \( \mu \) for the group which contains product \( i \), or 0 if \( i \) is not in a group. (Note that \( \mu_i \) is the same for all \( i \) in the same group.)
- \( \nu_i \) = the \( \nu \) for the subgroup which contains product \( i \), or 0 if \( i \) is not in a subgroup. (Similarly, note that \( \nu_i \) is the same for all \( i \) in the same subgroup.)
- \( L = \) The share-weighted average of the \( \lambda_i \):
  \[ L = \sum_{k=1}^{n} \lambda_k s_k \]  

(91)

The compensated own price elasticity of product \( i \) is then

\[ \eta_{ii} = -\lambda_i (1-s_i) - L + \lambda_i s_i - \mu_i (1-u_{ii}) - \nu_i (1-w_{ii}) \]  

(92)

while the cross price elasticity, the elasticity of the demand for good \( i \) with respect to the price of good \( j \), is

\[ \eta_{ij} = \lambda_j s_j + \lambda_j s_j + u_{ij} \mu_j + w_{ij} v_i \]  

(93)

Two tables are produced by the estimation program. One shows, for each product, its share in total expenditure in the base year, the group and subgroup of which it is a member and its share in them, its \( \lambda \) and the \( \mu \) and \( \nu \) of its subgroups, its own price elasticity, and various information on the income parameters. Thus, it contains all the data necessary for calculating any of the cross elasticities. It is small enough to be reasonably reproduced. The other table shows the complete matrix of own and cross elasticities. It is generally too large to be printed except in extract.
It should be noted that the complexity in estimating equation (21) comes from the term indicated by the product sign. Without this term, the equation could be estimated separately for each product or group of products. On the other hand, it is this term which gives Slutsky symmetry at the base point. If one did not care about this symmetry, then this term could be omitted from the equation, with a great reduction in complexity in estimation. Once the programming has been done to estimate with this term, however, it is little trouble to use the program.

So far, we have said little about the "income" term, the term within the first parenthesis of equation (21). In the equations reported below we have used just a constant, real income per capita, the first difference of real income per capita, and a linear time trend. Furthermore, we have used the same population measure, total population, for computing consumption per capita for all items. The estimation program, however, allows much greater diversity. By use of adult-equivalency weights, different weighted populations can be used for computing the per capita consumption of different items. Further, if the size distribution of income is known, it can be used to compute income-based indicators of consumption more appropriate to each item than just average income. Thus, the program allows a different income variable to be used for each consumer category. Finally, instead of just a linear time trend, one can use a "trend" variable appropriate to a particular category. For example, the percentage of the population which smokes could be used in explaining spending on tobacco. The estimation program allows for all these possibilities. On the other hand, in view of this diversity, it seemed pointless to try to place constraints on the parameters of the income terms to make the income terms add up to total income. Instead, in applying the estimated functions, one should calculate the difference between the assumed total expenditure and that implied by the equations and allocate it to the various items.

4. The Mathematics of Estimation

The function in equation (21) is nonlinear in all its parameters. In a system with 80 consumption categories there will be over 400 parameters involved in the simultaneous non-linear estimation. This size makes it worthwhile to note in this section some simplifying structure in the problem. All non-linear estimation procedures take some guess of the parameters, evaluate the functions with these values to obtain vectors of predicted values, \( \hat{x}_i \), and subtract these from the vectors of observed values, \( x_i \), to obtain vectors of residuals, \( r_i \), thus:

\[
\mathbf{r}_i = \mathbf{x}_i - \hat{\mathbf{x}}_i
\]

They then, in some way, pick changes in the parameters, and re-evaluate the function with the new values. The only difference in the various methods lies in how the changes in the parameters are picked. The Marquardt algorithm, which we use, is very nearly the same as regressing the residuals on the partial derivatives of the predicted values with respect to the parameters. It requires, in particular, these derivatives. For equation (21), they are reasonably easy to calculate if one remembers the formula from the table of derivatives:

\[
\frac{d a^x}{dx} = a^x \ln a
\]  \hspace{1cm} (95)

where \( \ln \) denotes the natural logarithm. Then for the derivative of the demand for the \( i^{th} \) good with respect to its own lambda is
\[
\frac{\partial \hat{x}_i}{\partial \lambda_i} = \hat{x}_i \ln \left( \frac{\prod p_i^{s_k}}{p_i} \right) = \hat{x}_i \left( \sum s_k \ln p_k - \ln p_i \right) \tag{96}
\]

and for j not equal to i
\[
\frac{\partial \hat{x}_i}{\partial \lambda_j} = \hat{x}_i \ln \left( \frac{p_j}{p_i} \right) s_j = \hat{x}_i \left( \ln p_j - \ln p_i \right) s_j \tag{97}
\]

and if i is a member of the group G
\[
\frac{\partial \hat{x}_i}{\partial \mu_G} = \hat{x}_i \ln \left( \frac{p_G}{p_i} \right) = \hat{x}_i \left( \ln p_G - \ln p_i \right) \tag{98}
\]

and if further i is a member of the subgroup g
\[
\frac{\partial \hat{x}_i}{\partial \nu_g} = \hat{x}_i \ln \left( \frac{p_g}{p_i} \right) = \hat{x}_i \left( \ln p_G - \ln p_i \right) \tag{99}
\]

To explain the estimation process, we shall denote the vector of parameters of the "income-and-time term," the term preceding the first dot in equation (21), for product i by \(a_i\) and the vector of parameters of the "price term", the rest of the formula, by \(h\). Thus, \(h\) consists of all values of \(\lambda, \mu,\) and \(\nu\). Note that \(h\) is the same for all products, though a particular \(\mu\) or \(\nu\) may not enter the equation a given commodity. If we let \(A_i\) be the matrix of partial derivatives of the predicted values for product i with respect to the \(a_i\) and similarly let \(B_i\) be the matrix of partial derivatives of the predicted values of product i with respect to \(h\), and finally let \(r_i\) be the residuals, all evaluated at the current value of the parameters, then the regression data matrix, \((X,y)\) in the usual notation, for three commodities is:

\[
(X, y) = \begin{pmatrix}
A_1 & 0 & 0 & B_1 & r_1 \\
0 & A_2 & 0 & B_2 & r_2 \\
0 & 0 & A_3 & B_3 & r_3
\end{pmatrix} \tag{100}
\]

If we now form the normal equations, \(X'Xb = X'y\) in the usual notation, we find

\[
\begin{pmatrix}
A_1' & 0 & 0 & A_1'B_1 \\
0 & A_2' & 0 & A_2'B_2 \\
0 & 0 & A_3' & A_3'B_3 \\
B_1'A_1 & B_2'A_2 & B_3'A_3 & \sum B_1'B_1
\end{pmatrix}
\begin{pmatrix}
da_1 \\
da_2 \\
da_3 \\
dh
\end{pmatrix} =
\begin{pmatrix}
A_1'r_1 \\
A_2'r_2 \\
A_3'r_3 \\
\sum r_1
\end{pmatrix} \tag{101}
\]
After initial values of the parameters have been chosen and the functions evaluated with these values and the sum of squared residuals (SSR) calculated, the Marquardt procedure consists of picking a scalar, which we may call M, and following these steps:

1. Compute the matrices of equation 34, multiply the diagonal elements in the matrix on the left by 1 + M and solve for the changes in the \(a_i\) and \(h\) vectors. Make these changes and evaluate the functions at the new values.

2. If the SSR has decreased, divide M by 10 and repeat step 1.

3. If the SSR has increased, multiply M by 10, go back to the values of the parameters before the last change, evaluate the functions again at these values, and repeat step 1.

The process is stopped when very little reduction in the SSR is being achieved and the changes in the parameters are small. (As M rises, the method turns into the steepest descent method, which can usually find a small improvement if one exists, while as M diminishes, the method turns into Newton's method, which gives rapid convergence when close enough to a solution that the quadratic approximation is good.)

To economize on space in the computer and to speed the calculations, we can take advantage of the structure of the matrix on the left side of equation (34). To do so, let \(Z_i\) be the inverse of \(A_i^tA_i\). Then by Gaussian reduction (34) can be transformed into

\[
\begin{pmatrix}
I & 0 & 0 \\
0 & I & 0 \\
0 & 0 & I \\
0 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
Z_1A_1'B_1 \\
Z_2A_2'B_2 \\
Z_3A_3'B_3 \\
\sum_{i=1}^3 B_i'B_1 - B_1'A_i Z_iA_i'B_1 \\
\end{pmatrix}
\begin{pmatrix}
da_1 \\
da_2 \\
da_3 \\
dh \\
\end{pmatrix}
= 
\begin{pmatrix}
Z_1A_1'r_1 \\
Z_2A_2'r_2 \\
Z_3A_3'r_3 \\
\sum_{i=1}^3 B_i'r_1 - B_1'A_i Z_iA_i'r_1 \\
\end{pmatrix}
\tag{102}
\]

The columns of the matrix on the left which are just columns of the identity matrix do not need to be stored in the computer. Instead, the program computes the terms in the last column of this matrix and in the vector on the right, stores only them, and at the same time builds up the sums in the lower right corner of the matrix and in the bottom row of the vector on the right. Once the matrix and vector of equation (35) are ready, the program solves the equations in the last row for \(dh\) and then substitutes back into the other equations to solve them for \(da_i\).

The estimation program initializes the income parameters by regressing the dependent variables on the just the constant, income, and trend terms. Then all lambda's are started at .25 and all mu and nu at 0. The program was written in Borland C++ 4.5 with DOS extender and with a double-precision version of the BUMP library of matrix and vector objects and operators. The time required to do the estimation seems to be roughly proportional to the fourth power of the number of sectors. The work of evaluating the B matrices and taking \(B'B\) grows roughly with the cube of the number of sectors, so the time required for a single iteration grows with the cube of the number of sectors. The number of iterations, however, seems to grow...
at least linearly with the number of sectors, so the total time required should grow with the fourth power of
the number of sectors. Thus, a 90-sector study can be expected to take about 16 times as long to estimate as
a 45-sector study. This is roughly what we have experienced, with a 93-sector USA system requiring about
100 minutes while a 42-sector Spanish study took only five or six minutes on a 133 MHz pentium. The USA
study required about 120 iterations. The big drops in the objective function started to appear after about 80
iterations.

5. Comparative estimation for France, Italy, Spain, and the USA

To test how adequate this system is for representing the consumer behavior in a variety of countries, it has
been estimated for France, Italy, Spain, and the USA. At the same time, so that the results would tell us
something about the similarities and differences among these countries, the categories have been made
as similar as possible. The categories, the groups, and the sub-groups are shown in the table below.
In using the word “test,” I should make clear that I do not mean any sort of test of statistical “significance,” which I regard as essentially meaningless here. The test is rather to see whether the system is flexible enough to fit the historical data with plausible values of the parameters. Moreover, it is not a test to see whether the program can find those reasonable values from the data alone. Whether or not that is possible depends upon what range of experience history has given us. It is often necessary to tell the program what values are plausible by soft constraints. The details of how that has been done are described in Appendix A on using the program.
The Italian and Spanish data were for forty categories of consumer expenditures, most of them being exactly comparable. The French data were more detailed but were clearly based on the same statistical concepts and could be aggregated to match the Spanish and Italian. The three European datasets showed that the statisticians who had prepared them had been talking to one another and had achieved some degree of comparability. No such fundamental comparability infected the U.S. data. It was, however, available in much more detail than was the European, and in most cases, it was possible to match the European concept -- as I understand it from the words in the definition -- fairly closely. There were a few exceptions among foods. The Europeans had the following sectors:

6 Fruits and vegetables, except potatoes
7 Potatoes
9 Coffee, tea, and cocoa

I could not match these with U.S. data but made up three sectors which at least kept the numbering the same for the other sectors. They were

6 Fresh fruit
7 Fresh vegetables
9 Processed fruits and vegetables

In the U.S. sectors, coffee, tea, and cocoa are in sector 10, Other prepared food.

Other known noncomparabilities included the Italians having no sector for Education but only one for text books, while the Spanish did not attempt to divide "all-included" vacation packages between Transportation and Hotels and restaurants though the others did. Finally, the U.S. has four categories which have no corresponding component in the European accounts. First, and largest, is the imputed space-rental value of owner-occupied housing, which is seemingly not in the Standard National Accounts (SNA) used by the Europeans. Second is Services rendered without payment by financial intermediaries (e.g. free checking accounts). The existence of these services is recognized by the SNA, but the European statistical offices (incorrectly) consider that all of these services are rendered to businesses, and thus appear in the intermediate part of the input-output table and do not enter GDP. Foreign travel shows up elsewhere in the European accounts and was not among the data series I had. Finally, Food furnished employees or eaten on farms seems not to be part of the European system or appears directly in the various food categories. These extra sectors account for about 15 percent of American consumption. Within the forty more or less comparable sectors, the share of the American sectors in total consumption will average about 15 percent below the European.

The regressions were run from 1971 to 1994 (1993 for France.) It quickly became apparent that nearly all of the histories could be fit well, but often one or more of the parameters would have nonsense values. The income elasticity might turn out negative while there was a strong positive time trend. The own price elasticities, which should be negative, frequently turned out positive, perhaps at the same time that the income elasticity was negative. In short, the data were insufficiently varied to identify well the parameters. Fortunately, the program used for the estimation (our creation) allowed for imposing "soft" constraints, which are essentially extra, artificial observations designed to tell the computer, before the estimation, what would be sensible regression coefficients. By using soft constraints, it is often possible to find equations with sensible coefficients which fit almost as well as the unconstrained equation. Except in Spain, where there was a drop in income in the mid 1980's before entry into the Common Market, time and income were very collinear, and it was necessary to softly constrain the time variable to be close to zero, though not exactly
zero. In Spain, there was also a very soft constraint suggesting that the time trend coefficient should be small, but it was softer than in the other countries and consequently stronger time trends appear in the Spanish equations than in the others. In cases of products which evidently had strong time trends in tastes, such as fats and oils or tobacco, the soft constraint on the time trend was removed. Of course, the fact that soft constraints were used which were not identical in the different countries may reduce the comparability of the results. But it also shows that the system can be adapted to the situation in different countries.

Before commenting on the individual products, let us look at the results for the group parameters, as shown below.

<table>
<thead>
<tr>
<th></th>
<th>µ</th>
<th>ν</th>
</tr>
</thead>
<tbody>
<tr>
<td>food</td>
<td>dress</td>
<td>house</td>
</tr>
<tr>
<td>USA</td>
<td>0.25</td>
<td>-0.23</td>
</tr>
<tr>
<td>Italy</td>
<td>-0.02</td>
<td>1.83</td>
</tr>
<tr>
<td>Spain</td>
<td>0.12</td>
<td>-0.34</td>
</tr>
<tr>
<td>France</td>
<td>0.61</td>
<td>0.15</td>
</tr>
</tbody>
</table>

The components of the Food group did indeed turn out to be substitutes in the USA, Spain, and, especially, France. The protein sources were especially strong substitutes with one another in France and less so in Spain and Italy. In the USA, their special interaction was in the direction of complementarity. Buying cars and operating them were decidedly complements in the USA, Spain, and France, but were rather strongly substitutes in Italy. The Italians may not, however, be totally crazy; automobile repair and new cars may indeed be substitutes. Shoes and Clothing turn out to be strongly substitutes in the USA and Italy, weak substitutes in France, and complements in Spain. The household furnishing and operating sectors showed considerable interaction, but were complements in America and substitutes in Europe. The medical sectors were complements in America and France and weakly substitutes in Italy. There is little interaction between public and private transportation in any of the countries.

Examining the individual sectors shows many interesting differences as well as some basic similarities among the countries. For each product, we will show the results of estimation for all four countries. The order of lines in these mini tables is USA, Italy, Spain, France. The sector titles have been left in the original language both to indicate the country and to describe as exactly as possible the content. On each line in the mini-tables for each product, you will find:

- `nsec` the sector number
- `title` the title of the product group in the language of the country
- `G` the number of the group in which the product is included. A 0 indicates that it was not in a group.
- `S` the number of the product’s subgroup. A 0 means that it was not in a subgroup.
- `I` inclusion code: 1 if the product was included in the estimation of the system, otherwise 0.
- `lamb` the value of lambda, λ, for this product.
share the share of this product in total consumption in the base year, the year when all the prices were equal to 1. Unfortunately for purposes of comparison of these shares, the base years were different: 1992 for the USA, 1988 for Italy, 1986 for Spain, and 1980 for France. These differences should have little effect on comparability except on these shares.

IncEl The income elasticity, the percentage by which purchases of this item increase when income increases one percent.

DInc The ratio of the coefficient on the change in income to the coefficient on income.

time% the change in demand for the product due to the passage of one year (without change in income or price) expressed as a percentage of the average purchase.

PrEl the elasticity of demand for the product with respect to its own price.

Err% Standard error of estimate expressed as a percentage of the average value.

rho Autocorrelation of the residuals.

The commentary on each group also reflects looking at the graph of the fit in each country for each product. These graphs are, unfortunately, too space-intensive to print.

We start with the staff of life.

```
nsec title                  G  S I  lamb share IncEl DInc  time%  PrEl  Err%  rho
1 Cereal and bakery produ  1  0 1  0.18 0.013  0.18 -0.60  0.01 -0.55  4.33  0.80
1 Pane e cereali           1  0 1  0.06 0.024  0.13 -1.59  0.00 -0.12  1.56  0.74
1 Pan y cereales           1  0 1 -0.12 0.026  0.18 -0.17 -0.26 -0.02  2.53  0.61
1 Pain et cereales         1  0 1  0.05 0.024  0.45 -0.03  0.00 -0.69  1.44  0.49
```

The Food group holds some striking similarities among the countries as well as big differences. Bread and bakery products (1) have seen virtually no growth in per capita consumption over the years covered here. Note, however, that the share is nearly twice as high in Europe as in America. The income elasticities, however, do not come out at zero but have been offset in the US and France by significant price elasticities. Italy shows both smaller income elasticities and small price elasticity, while in Spain the income elasticity comes out the same as that in the US but is offset by a negative trend of half a percent per year. The higher income elasticity in France may reflect the attractiveness of real croissants, brioche, and the like.

```
nsec title                   G  S I  lamb share IncEl DInc  time%  PrEl  Err%  rho
2 Meat                     1  1 1 -0.16 0.018  0.03  0.47 -0.20 -0.19  3.98  0.74
2 Carne                    1  1 1  0.05 0.056  0.23  2.00  0.00 -0.15  2.78  0.81
2 Viandes                  1  1 1 -0.01 0.066  0.49 -1.00 -0.13 -0.21  2.94  0.50
```

Only Spain has seen any noticeable growth in Meat (2) demand since 1980. It showed an income elasticity of .5 as did France, but Spain has had greater income growth. Both the US and Italy have very low income elasticities, though Italy has a positive “taste” term, while the US and Spain both show negative “taste” trends.

```
nsec title                   G  S I  lamb share IncEl DInc  time%  PrEl  Err%  rho
3 Fish & seafood           1  1 1  1.78 0.002  1.17 -0.07 -0.20 -2.12  8.90  0.52
```
In striking contrast to Bread and Meat, Fish and seafood (3) shows strong income elasticities, above 1.0 in the USA, Italy, and especially France. Fish is definitely the food of the affluent in these countries, while it definitely is not in Spain, where consumption has declined steadily as income rose. Note, however, that the share of fish in the budget of Pedro was twice that of Pietro, three times that of Pierre, and twelve times that of Peter.

When it comes to Milk and dairy products (4), the US is the outlier. The European countries, where consumption runs from 2.5 to 3.3 percent of the total budget, have been increasing consumption steadily, while the USA is cutting back sharply from its already low share of .8 percent. The equation for France attributes the growth to income, the Spanish and Italian equations, more to taste trends. One may say that the American concern about cholesterol has not penetrated the European mind, or one may say that the American cheese industry has never approached the European in placing temptation in front of the consumer.

Fats and oils (5) have uniformly low income elasticities and negative taste trends. Fruit has been declining in the US, while Fruit and vegetables (6), including canned and frozen, have been rising in Italy and stable in Spain and France. Recall that the sectoral definitions are not comparable here. The rest of the story for the US is found in Fresh vegetables (7), also in gentle decline, and in Processed fruits and vegetables (9), which also fails to show any growth. The total share for the US is 1.3 percent of the budget, only a half or a third of that of the European countries. That low share does not necessarily mean that we consume less than they do of these products. There are at least two other factors: larger total consumption and lower prices on agricultural products. The graphs for Potatoes show that the French are rapidly losing their appetite for French fries, as are the Spanish, while the Italians are not.
Sugar (8) proved to be a problem in both Italy and Spain and was removed from the system in these two countries. The problem arose from substantial fluctuations in the price which had little effect. In the US and France, the system had no problem handling the product, and virtually identical price elasticities, -.4, were found. In France, however, there has been a strong trend away from sugar not seen here.

The Other prepared foods (10), the sauces, mixes, and just-run-it-in-the-microwave products have shown strong growth. For the USA, Italy, and France the equations attribute this growth to income, because of the aversion to time trends expressed in the soft constraints. In Spain, however, there were greater fluctuations in income and it was easier for the regression to distinguish time from income. It found that the income elasticity was actually fairly low, .34, and used a strong time trend, .9 percent per year, to account for the growth.

The Soft drink industry (11), stagnant in this country despite an income elasticity of .8 because of sharp price increases, has boomed in Italy and France, with income elasticity estimates of 1.6 and 1.8, respectively. The Spaniards have not been so easily seduced; they show an income elasticity of 1.1 and a negative time trend of .7 percent per year.

Alcoholic beverage (12) sales have been static in the USA country and France, but declining in Italy and Spain. All countries showed very low income elasticities.

The sharp decline in the use of Tobacco (13) in the USA has no parallel in Europe. In France, it was even rising up until 1992, showing an income elasticity of .93.
One of the surprises for me was the sad story of France in the consumption of Clothing (14). I had thought of the French as fashion conscious. Not at all, according to these equations. Clothing accounts for a smaller share in France than in any of the other European countries. The French income elasticity is only .15, against .8 for Spain, 1.4 for the U.S., and .9 for Italy, which has also the highest share of the budget going to clothes. Clearly it is the Italians who are the sartorially conscious nation.

The same story holds for Footwear (15). The U.S., Spain, and Italy all came out with an income elasticity of 1.2, while in France it was only .2.

Rent (16) on living quarters has risen steadily in all four countries; the income elasticities are 1.0 in the U.S. and Italy; 1.7 in France; but only .5 in Spain. Rental payments did not fall during the Spanish slump of the early 1980's, so the equation attributes most of the growth to the time trend rather than to income.

Energy consumption (17) has been virtually constant in the U.S.; the equation found low income and price elasticities. All three European countries, but especially Spain, have seen significant growth. It is interesting that in Spain, where the equation was given less indication to avoid trend terms than in Italy and France, it used that extra freedom to get virtually the same income elasticity as was found in Italy, attributing the extra growth in Spain to the time trend.
The French are again the outlier in demand for Furniture (18). Spanish and Italian income elasticities are high and similar, 1.6 and 1.4 respectively; the US is a respectable 1.1; but France is only .4. Clearly the French have other priorities.

<table>
<thead>
<tr>
<th>nsec title</th>
<th>G</th>
<th>S</th>
<th>I</th>
<th>lamb share</th>
<th>IncEl</th>
<th>IncDnc</th>
<th>time%</th>
<th>PrEl</th>
<th>Err%</th>
<th>rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>19 Floor coverings and tex</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.06</td>
<td>0.004</td>
<td>1.64</td>
<td>-0.02</td>
<td>-0.17</td>
<td>-0.01</td>
<td>8.43</td>
</tr>
<tr>
<td>19 Biancheria e altri arti</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.04</td>
<td>0.011</td>
<td>1.42</td>
<td>-0.30</td>
<td>0.00</td>
<td>-0.68</td>
<td>6.98</td>
</tr>
<tr>
<td>19 Artículos textiles</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.00</td>
<td>0.009</td>
<td>1.19</td>
<td>0.38</td>
<td>-1.08</td>
<td>-0.22</td>
<td>5.69</td>
</tr>
<tr>
<td>19 Art. de ménage en texti</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.00</td>
<td>0.007</td>
<td>0.00</td>
<td>-59.48</td>
<td>-0.03</td>
<td>-0.84</td>
<td>3.59</td>
</tr>
</tbody>
</table>

The story is the same for Carpets, curtains, and household linens (19). The French income elasticity is exactly 0, while it is 1.2 to 1.6 for the other three.

<table>
<thead>
<tr>
<th>nsec title</th>
<th>G</th>
<th>S</th>
<th>I</th>
<th>lamb share</th>
<th>IncEl</th>
<th>IncDnc</th>
<th>time%</th>
<th>PrEl</th>
<th>Err%</th>
<th>rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Kitchen &amp; hh appliances</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.10</td>
<td>0.005</td>
<td>0.53</td>
<td>-0.06</td>
<td>-0.04</td>
<td>-0.05</td>
<td>5.31</td>
</tr>
<tr>
<td>20 Elettrodomestici</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.37</td>
<td>0.012</td>
<td>1.14</td>
<td>0.64</td>
<td>0.00</td>
<td>-0.35</td>
<td>2.55</td>
</tr>
<tr>
<td>20 Electrodomésticos</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.10</td>
<td>0.010</td>
<td>1.64</td>
<td>0.71</td>
<td>-1.27</td>
<td>-0.12</td>
<td>5.30</td>
</tr>
<tr>
<td>20 Ap. de cui., de chauf.</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.02</td>
<td>0.016</td>
<td>0.44</td>
<td>-0.31</td>
<td>0.00</td>
<td>-0.76</td>
<td>4.10</td>
</tr>
</tbody>
</table>

Kitchen and household appliances (20) show a similar pattern, except that both the U.S. and France have income elasticities close to .5, while in Italy and Spain, they are 1.1 and 1.6 respectively.

<table>
<thead>
<tr>
<th>nsec title</th>
<th>G</th>
<th>S</th>
<th>I</th>
<th>lamb share</th>
<th>IncEl</th>
<th>IncDnc</th>
<th>time%</th>
<th>PrEl</th>
<th>Err%</th>
<th>rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 China &amp; glaswr, tablwr</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.32</td>
<td>0.005</td>
<td>0.74</td>
<td>-0.03</td>
<td>0.08</td>
<td>-0.27</td>
<td>3.47</td>
</tr>
<tr>
<td>21 Cristallerie,vasellame</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.66</td>
<td>0.006</td>
<td>1.02</td>
<td>0.21</td>
<td>-0.03</td>
<td>-0.10</td>
<td>3.28</td>
</tr>
<tr>
<td>21 Utensilios domésticos</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.04</td>
<td>0.005</td>
<td>0.17</td>
<td>-0.38</td>
<td>-1.65</td>
<td>-0.03</td>
<td>9.62</td>
</tr>
<tr>
<td>21 Verrerie, vaisselle et</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.01</td>
<td>0.015</td>
<td>0.41</td>
<td>-0.17</td>
<td>0.00</td>
<td>-0.78</td>
<td>2.41</td>
</tr>
</tbody>
</table>

It is Italians and Americans who care about China, glassware, and tableware (21). The French have been particularly sensitive to the rising relative price of these products.

<table>
<thead>
<tr>
<th>nsec title</th>
<th>G</th>
<th>S</th>
<th>I</th>
<th>lamb share</th>
<th>IncEl</th>
<th>IncDnc</th>
<th>time%</th>
<th>PrEl</th>
<th>Err%</th>
<th>rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>22 Other non-durables and</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.59</td>
<td>0.019</td>
<td>0.52</td>
<td>-0.01</td>
<td>0.00</td>
<td>-0.57</td>
<td>2.05</td>
</tr>
<tr>
<td>22 Art.non dur. e servizi</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.24</td>
<td>0.011</td>
<td>1.32</td>
<td>-1.92</td>
<td>0.01</td>
<td>-0.49</td>
<td>7.99</td>
</tr>
<tr>
<td>22 Mantenimiento</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-0.10</td>
<td>0.015</td>
<td>0.95</td>
<td>0.12</td>
<td>-1.07</td>
<td>-0.11</td>
<td>3.82</td>
</tr>
<tr>
<td>22 Art. de ménage non-dur</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.01</td>
<td>0.018</td>
<td>1.15</td>
<td>0.02</td>
<td>0.00</td>
<td>-0.78</td>
<td>2.08</td>
</tr>
</tbody>
</table>

Domestic service (23) fell steadily in the U.S. from 1950 up to 1981, whereupon it suddenly stabilized and began a slow rise. Strikingly similar patterns appear in Spain and France, with low points in 1986 or 1987. Italy, by contrast, has shown strong growth all along, with an income elasticity of 1.4. All equations except the Spanish showed strong price elasticities.

<table>
<thead>
<tr>
<th>nsec title</th>
<th>G</th>
<th>S</th>
<th>I</th>
<th>lamb share</th>
<th>IncEl</th>
<th>IncDnc</th>
<th>time%</th>
<th>PrEl</th>
<th>Err%</th>
<th>rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 Drug preparations and s</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0.08</td>
<td>0.018</td>
<td>1.42</td>
<td>-0.02</td>
<td>-0.04</td>
<td>0.00</td>
<td>3.48</td>
</tr>
<tr>
<td>24 Medicinali e prod. farm</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.022</td>
<td>2.58</td>
<td>-1.01</td>
<td>-0.04</td>
<td>0.00</td>
<td>6.66</td>
</tr>
<tr>
<td>24 Medicamentos</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.016</td>
<td>3.03</td>
<td>-0.96</td>
<td>-1.42</td>
<td>0.00</td>
<td>15.31</td>
</tr>
<tr>
<td>24 Medicamentos et autres p</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.06</td>
<td>0.021</td>
<td>1.50</td>
<td>0.21</td>
<td>0.00</td>
<td>-1.13</td>
<td>7.62</td>
</tr>
</tbody>
</table>

Medicines (24) have shown explosive growth in Europe. Note that all three European graphs ran off the standard scale. In Italy and France, this sector had to be thrown out of the system. The prices
were rising and demand was soaring. Clearly, the problem was that the medicines were being paid for by third parties. The budget constraint had little relevance for the European buying medicine.

Similarly, *Ophthalmic and orthopedic devices* (25) could not be accommodated in the system in Italy and France. No price sensitivity but enormous income sensitivity is found in all countries.

*Services of physicians, dentists, and other medical professionals* (26) could be handled by the system in all countries. Only France, however, showed strong price sensitivity -- the U.S. showed none.

The demand for *Hospitals* (27) has been decidedly more moderate than for the other members of the health care group. Only the U.S. shows an income elasticity greater than 1.0, and the French is down to .3.

Income elasticity for *Automobiles* (28) is strong in all countries: 1.3 in the U.S., Italy, and France, and 1.8 in Spain -- and the Spanish equation has a 1 percent per year time trend on top of that income elasticity. The Spanish are plainly making up for lost time in equipping themselves to congest their streets and highways. Price sensitivity was slight, except in Italy.

*Motor vehicle operation* (29), however, has an income elasticity of only .7 to .9 in three countries and 1.1 in Spain, where there is also a noticeable positive time trend.
Public transportation (30) claims a share of the consumer budget in Europe that is twice as large as the American share. Moreover, the income elasticities in Italy and France are twice what they are in the U.S. The U.S., however, is the most price sensitive, though the elasticity is only -.4.

Communications (31) ran off the standard scale in all the European graphs. In Spain, where the equation was freer to use a time trend, the income elasticity came in at 1.3 with a strong positive trend of 2.8 percent per year added on to the income effect. In France, the equation attributed all the growth to income with an elasticity of 3.7! Since in both countries, much of the growth is attributable to the modernization of a once stodgy telephone monopoly, I suspect the Spanish equation is more appropriate.

The one and only runaway sector in the U.S. is TV, radio, audio, musical instruments, and computers (32). The enormous growth is, of course, in the computer component. We have not used the official “computer deflator” which would have made it grow even faster, but have left computers undeflated. It is not clear to me whether or not computers are in this category in Europe. They probably are, because the category’s the share in total spending is considerably smaller here than in Europe. Even so, the income elasticity in the U.S. 1.5, the same as in Spain and France, and below Italy’s 1.9. The key to the super fast grow is the relatively strong price elasticity, -1.1, coupled with the rapid decline of the relative price of these products.

Despite the onslaught of electronic information and entertainment, Books, magazines, and newspapers (33) have hung on to their absolute level of sales, though they have been losing share in the consumer’s dollar, as appears from the modest income elasticities of .6 or .7, the highest being in Italy, which, with Spain, has the highest share of the consumer’s budget, almost twice that in the USA.
Least that dismal comparison leaves you a bit embarrassed to be American, take heart from Education (34), for which the American budget share is nearly three times that of the European. Alas, however, the difference is much affected by accounting conventions. In the U.S., all of the endowment income of schools and colleges counts as consumption expenditures on education. The income elasticity in the USA is .9, .7 in Spain, and a paltry .4 in France, where the budget share is less than a fifth of that here. The French expect the state to cover all the costs of education.

Recreational services (35), which includes spectator sports, have been a growth industry everywhere except in Spain. The U.S. leads with an income elasticity of 1.8, and a budget share of 3.1 percent. Italy and France also have elasticities above 1 and budget shares of 2.3 and 1.9 percent respectively. In Spain, however, the income elasticity was only .7 and there was a negative time trend of a percent per year. My suspicion is that the great national spectator sport of bull fighting is to some extent losing its hold on the imagination of young, urban Spaniards.

Personal care articles and services (36), covering from tooth paste to hair salons, has been a growth industry in Europe, with income elasticities of 1.4 in France and 1.2 in Italy, in contrast to .7 in America. The Spanish equation used its greater freedom to use a trend term to find almost exactly the American income elasticity but add to it a time trend of .9 percent per year.

The Hotel and restaurant business (37) enjoys fairly good income elasticities (.8 to 1.0) in all four countries.

The Other goods category (38) seems to be totally non-comparable across countries. Just the fact that it accounts for 15 percent of the Spanish budget and 1.6 percent of the French budget indicates that the contents of the sector must be quite different.
Financial services and insurance (39) has been a major growth industry in France and Italy, where it found income elasticities of 3.7 and 1.8, respectively. What looks like a change of definition has dominated the Spanish series. In America, this is a much more mature industry with roughly ten times the budget share it carries in Europe. (The services rendered without payment by financial intermediaries are not included here.)

Other services (40) are also much more important in the US than in Europe, but continue to have a higher elasticity here (1.3) than in France (.9) or Spain (.5). The Italian elasticity is high (1.2) but on a very small base, only 0.8 percent of the budget.

It would be safe and politic to conclude that this comparison has shown that the new functional form is capable of representing a variety of behavior, including significant substitution and complementarity. While that is, from a technical point of view, the most important conclusion, I cannot pass up the temptation to try to picture the national characters as they appear from these estimates. This venture is especially dangerous since citizens of all the four countries may be readers. Please take no personal offence.

The American has enough if not too much to eat, has become diet conscious and is cutting down on cholesterol but is, sad to say, bothering less and less to prepare fresh fruits and vegetables. He would like to eat more fish but is very sensitive to its price. He has no particular interest in more alcohol; soft drinks are sort of a necessity, not a special treat, and smoking is just a way to make yourself into a social outcast. Ms. America is very concerned about how she and her family are dressed. Housing and furnishing and equipment for the house are important, but using more energy for running the home is a matter of no interest. More domestic help for the working woman is becoming important again. The nation has gone bonkers over home computers. Every child of any age must have one. Books? Well, of course, a few books. But sports, concerts, plays, skiing, sailing, any kind of recreation, that’s what America is all about. Relative to the Europeans, the American is not starved for medical care, and growth in this area has been less here than there. Automobiles are important but not much of a class symbol; operating them is just a necessity. Private education and tuition at public universities is a serious matter. Use of communication has grown because of the declines in its price. Public transport is to be avoided.

The Italian is outstanding among Europeans for dressing well. He is proud of his country’s cucina, and would gladly eat more cheese, fish, and, above all, soft drinks; but he is losing interest in vino. Of pasta, he has, thank you, enough. Eating out at a good trattoria, ah, that’s worth the price. He continues to puff away on his cigarette just to show his defiance of statistics. In an energy-poor country, he wants more electricity and fuel. Signora is concerned not only with dressing the family well but is especially concerned to have a well-furnished house, refined furniture, attractive carpets and linens, and a bit of style and elegance in china and crystal. She wants appliances to help her with the house work. But most of all she wants domestic help. Dispensing with domestic servants was a modern idea that never crossed her sensible mind. The family has been upgrading its car, especially because prices have been coming down relative to other goods. The modest motor scooter is giving way to the even noisier motorcycle. But public transport is still a respectable way to get around. Especially if you have a portable telephone -- and who doesn’t? Yes, computers and audio
equipment have caught on fast, just not to the mania level of the Americans. Reading the newspaper is very important, and books still hold more allure than in any of the other three countries. One might suppose that just watching Italian politics would provide spettacoli enough, but no, recreation is high on the list of priorities. Socialized medicine led to explosive growth of spending on medicine but not on doctors, and certainly not on hospitals.

The Spanish are the newly rich of Europe. And the riches come after a period of declining income in the early 1980’s. They have increased meat consumption in the last five years. Eating out is great, but please, no more fish! And less potatoes and wine. But an extra cigarette, por favor. Clothing is a good thing to economize on, as are shoes, though, of course, as income goes up you should look just a little better. Pretty much the same goes for furniture, rugs, and linens. China and glassware are an especially good place to economize when your income rises. One good place to put some of the savings on these goods is into more and better appliances along with the electricity to run them. Indeed, electricity is showing such a growth that one becomes suspicious that air conditioning might be catching on. But top priority for these savings is the car. No other of our countries is close to the Spanish income elasticity for cars. And if you have bought the car, then you have to drive it. But the new high-speed rail lines are making public transportation competitive again. Right behind the car in priority comes “recreational equipment” which seems to correspond to home electronics, including possibly the computer. Never mind, however, about those recreational services that everybody else is so crazy about. Just living in Spain is recreation enough. As in all three European countries, medical expenditures have skyrocketed: medicines, therapeutic devices, and services of doctors. Even hospital services have seen some rise.

Now the French seem utterly indifferent to improving, when income rises, how they are dressed or to how their house is furnished or to what sort of china or glassware they use, but not to what they eat and drink. Increase the French family’s income by one percent, and it will spend .8 percent more on dining out, .5 percent more on meat, 1.6 percent more on fish and those delicious shellfish, .8 percent more on cheese, 1.8 percent more on candy and “other” prepared foods, 1.8 percent more on soft drinks, including, of course, mineral water, and even a bit more, .2 percent, on wine. If the French don’t uphold their reputation as the fashion center of Europe, they are certainly les gourmands of the continent. It must be added that they are cutting back sharply on sugar and potatoes. Alone among the four countries, they are increasing their use of tobacco. They attach less priority to buying household appliances than to increasing meat consumption. A pleasant effect of that indifference is that energy consumption has remained stable. They have relatively little interest in new cars, relative, that is, to their neighbors in Spain and Italy, and expenditures on operating the cars are correspondingly stable. Public transport is more income elastic than is operation of cars. Besides their interest in food, they spend added income on personal care, on home electronics and recreational equipment, on cultural and sporting events, on financial services, and on communications. As in the other European countries, the large shares of increased income have gone to -- or come in the form of -- medicines, therapeutic devices, and services of doctors.

Well, it seems we are not all the same. There do appear to be national differences that go beyond language. For the present purposes, however, the important point is that this new functional form seems to be able to work well in what turns out to be a surprising variety of situations. Perhaps it is adequate.
References


Appendix A. Use of the estimation program

The estimation program has two control input matrices, groups.ttl and softcon.dat, and several data matrices, consum.dat, prices.dat, cstar.dat, popul.dat, and time.dat.

The groups.ttl file, as the name suggests, defines the groups. It also specifies which categories are sensitive and which insensitive to price, which weighted population, which income variable, and which trend variable is to be used by each category. This file for the Spanish study is shown in the box below. Its first column consists of simply the integers from 1 to n, the number of categories of consumption. The second column carries the number of the group in which the category falls, or a zero if it is not assigned to a group, and the third column carries the number of the subgroup to which the category belongs or a zero if it belongs to none. The fourth is the number of the weighted population to be used for the item, the fifth is the number of the "income" (or Cstar) series to be used, the sixth is the number of the "trend" series to be used, and the seventh is a 1 if the category is a regular, price-sensitive commodity or a 0 if it is not. Although conceptually we have thought of neatly defined groups and subgroups strictly within the groups, the computer program makes no effort to enforce this tidy structure. It is possible to form "subgroups" with categories drawn from more than one group.

The second major control file is softcon.dat, which gives soft constraints for the various equations. It is, in fact, hardly to be expected that all parameters would come out with reasonable values when so many of the variables have similar trends. Thus the use of soft constraints on the coefficients is an integral part of the estimation process. The estimation program allows the user to specify the desired value of any parameter except the constant term and to specify a "trade-off parameter" to express the user's trade-off between closeness of fit and conformity with desired values of the parameters. In these studies, I began with constraints saying that I wanted the time trends to be close to zero. I then worked on the income elasticities to get them all positive; for some products, that meant relaxing the soft constraint on the time trend. Then I added soft constraints to make the own price elasticities all negative. Finally, some of the coefficients on the change in income had to be constrained to keep them from being more negative that the income term is positive.

The softcon.dat file for Spain is shown in a box below. For each product, there can be specified desired values of the income elasticity, the change in income in elasticity units, the time trend as a percent of the base year (1988) value, lambda, and the mu and nu of the group and subgroup. The table shows for each of these a pair of numbers, the desired value and the trade-off parameter. If the trade-off parameter is 0, the desired value has no effect on the estimation. The higher the parameter, the stronger the constraint relative to the data. A value of 1.0 for the trade-off parameter gives about equal weight to the constraint and to the data. Constraints on mu and nu values can be specified on the line for any member of the group or subgroup, but I have always placed them on the line of the first item in the group or subgroup. This table is, in fact, precisely the way the constraints are entered into the program; the table shows the contents of the file softcon.dat, which is read by the estimation program.
The Groups.ttl File for Spain

# Groups.ttl. Columns are
# 1 The consumption category number
# 2 The group number
# 3 The subgroup number
# 4 Which weighted population number to be used with this category
# 5 Which Income (Cstar) variable
# 6 Which Trend variable
# 7 Use price terms (1 = yes, 0 = no)
# 8 The title of the category

1 1 1 1 1 1 1  Pan y cereales
2 1 1 1 1 1 1  Carne
3 1 1 1 1 1 1  Pescado
4 1 1 1 1 1 1  Leche, queso y huevos
5 1 0 1 1 1 1  Aceites y grasas
6 1 0 1 1 1 1  Frutas y verduras
7 1 0 1 1 1 1  Patatas y tubérculos
8 0 0 1 1 1 0  Azúcar
9 1 0 1 1 1 1  Café, té y cacao
10 1 0 1 1 1 1  Otros alimentos
11 1 0 1 1 1 1  Bebidas no alcohólicas
12 1 0 1 1 1 1  Bebidas alcohólicas
13 0 0 1 1 1 1  Tabacos
14 2 0 1 1 1 1  Vestido
15 2 0 1 1 1 1  Calzado
16 0 0 1 1 1 1  Alquileres y agua
17 0 0 1 1 1 1  Calefacción y alumbrado
18 3 0 1 1 1 1  Muebles
19 3 0 1 1 1 1  Artículos textiles
20 3 0 1 1 1 1  Electrodomésticos
21 3 0 1 1 1 1  Utensilios domésticos
22 3 0 1 1 1 1  Mantenimiento
23 3 0 1 1 1 1  Servicio doméstico
24 4 0 1 1 1 0  Medicamentos
25 4 0 1 1 1 1 0  Aparatos terapéuticos
26 4 0 1 1 1 1 0  Servicios médicos
27 4 0 1 1 1 1 0  Atención hospitalaria y seguro médico privado
28 5 2 1 1 1 1  Compra de vehículos
29 5 2 1 1 1 1  Gasto de uso de vehículos
30 5 0 1 1 1 1  Servicios de transporte
31 0 0 1 1 1 1  Comunicaciones
32 3 0 1 1 1 1  Artículos de esparcimiento
33 0 0 1 1 1 1  Libros, periódicos y revistas
34 0 0 1 1 1 1  Enseñanza
35 0 0 1 1 1 1  Servicios de esparcimiento
36 0 0 1 1 1 1  Cuidados y efectos personales
37 0 0 1 1 1 1  Restaurantes cafés y hoteles
38 0 0 1 1 1 1  Otros artículos n.c.o.p.
39 0 0 1 1 1 1  Servicios financieros n.c.o.p.
40 0 0 1 1 1 1  Otros servicios n.c.o.p
41 0 0 1 1 1 1  Viajes turísticos todo incluido
The consum.dat file begins with some dimensions and dates and then contains the data on consumption in almost exactly the form in which it would be written by the G command matty. The layout is shown in the above box for the Spanish case; the ... show where material has been cut out of the file to make it fit on the page. Notice the four numbers with which it begins. Each should be on its own line. Then come the data, with 20 series at a time across the "page". Comments may be introduced in the data by beginning the line with a #.
<table>
<thead>
<tr>
<th># Date</th>
<th>kcpi1</th>
<th>kcpi2</th>
<th>kcpi3</th>
<th>kcpi4</th>
<th>...</th>
<th>kcpi20</th>
</tr>
</thead>
<tbody>
<tr>
<td>70.000</td>
<td>499.340</td>
<td>1021.118</td>
<td>549.566</td>
<td>467.982</td>
<td>...</td>
<td>164.161</td>
</tr>
<tr>
<td>71.000</td>
<td>490.641</td>
<td>1022.712</td>
<td>571.055</td>
<td>466.438</td>
<td>...</td>
<td>175.657</td>
</tr>
<tr>
<td>72.000</td>
<td>506.050</td>
<td>1017.367</td>
<td>574.642</td>
<td>472.211</td>
<td>...</td>
<td>201.513</td>
</tr>
<tr>
<td>73.000</td>
<td>548.905</td>
<td>1182.404</td>
<td>582.436</td>
<td>498.514</td>
<td>...</td>
<td>210.624</td>
</tr>
<tr>
<td>74.000</td>
<td>554.821</td>
<td>1338.080</td>
<td>536.756</td>
<td>562.671</td>
<td>...</td>
<td>202.528</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>94.000</td>
<td>599.644</td>
<td>1666.945</td>
<td>606.117</td>
<td>735.611</td>
<td>...</td>
<td>289.384</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Date</th>
<th>kcpi21</th>
<th>kcpi22</th>
<th>kcpi23</th>
<th>kcpi24</th>
<th>...</th>
<th>kcpi40</th>
</tr>
</thead>
<tbody>
<tr>
<td>70.000</td>
<td>109.692</td>
<td>286.789</td>
<td>184.407</td>
<td>272.749</td>
<td>...</td>
<td>68.330</td>
</tr>
<tr>
<td>71.000</td>
<td>117.373</td>
<td>299.460</td>
<td>188.615</td>
<td>313.366</td>
<td>...</td>
<td>74.745</td>
</tr>
<tr>
<td>72.000</td>
<td>134.650</td>
<td>342.775</td>
<td>189.809</td>
<td>337.034</td>
<td>...</td>
<td>79.912</td>
</tr>
<tr>
<td>73.000</td>
<td>140.738</td>
<td>351.768</td>
<td>187.654</td>
<td>340.828</td>
<td>...</td>
<td>85.539</td>
</tr>
<tr>
<td>74.000</td>
<td>135.328</td>
<td>350.958</td>
<td>180.607</td>
<td>370.052</td>
<td>...</td>
<td>86.795</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>94.000</td>
<td>121.689</td>
<td>412.728</td>
<td>183.031</td>
<td>780.092</td>
<td>...</td>
<td>165.796</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Date</th>
<th>kcpi41</th>
<th>kcpi42</th>
</tr>
</thead>
<tbody>
<tr>
<td>70.000</td>
<td>34.073</td>
<td>298.342</td>
</tr>
<tr>
<td>71.000</td>
<td>38.251</td>
<td>334.738</td>
</tr>
<tr>
<td>72.000</td>
<td>45.404</td>
<td>395.237</td>
</tr>
<tr>
<td>73.000</td>
<td>48.923</td>
<td>425.776</td>
</tr>
<tr>
<td>74.000</td>
<td>55.430</td>
<td>479.674</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>94.000</td>
<td>48.288</td>
<td>934.576</td>
</tr>
</tbody>
</table>

The ... indicate where data have been removed to fit the into this box.

Exactly the same format is followed for the prices.dat file, which give the price indexes, except that the four numbers at the top are omitted. The Cstar.dat, which gives the income series, begins with the number of such series. It then has these series arranged in columns. It has one extra year of data at the beginning so that the first difference of income can be calculated. The Popul.dat file is very similar; it begins with an integer giving the number of populations, followed by data in the same format. It also has the extra year at the beginning. Finally the tempi.dat file gives various series which may be used as the time trend. Like the popul.dat file, it has the number of series at the beginning but does not have the extra year of data at the beginning.

Once the files groups.ttl, consum.dat, prices.dat, estar.dat, tempi.dat, and softcon.dat are ready, the program is run by the command "symcon [n]" from the DOS prompt. The optional parameter, n, is the number of iterations to be run before turning over control to the user. Thus "symcon" will run only 1 iteration and then give the user the option of quitting (by tapping y) or continuing the Marquardt process another iteration. If the command given is "symcon 40", then 40 iterations are automatically run without pausing for user input. In this case, when the limit is reached, the program sounds three long notes: low, high, low. A symcon calculation started in this way can be put into the background of a multitasking operating system such as OS2. When it has reached the limit, the notes will sound, and the user can turn his attention to it. To check that data has been read correctly, use "symcon d". (The d is for "debug").