IO-Snap

Technical Document 2025-01

Verifying Output Multipliers with the Leontief Price Model

David Kay
DavedKay@gmail.com

January 15, 2025

Abstract. The Leontief Price Model provides a convenient way to verify the accounting accuracy of IxI input-output tables and corresponding output multipliers derived from underlying make and use tables. This brief technical document provides an example of how this can be accomplished using IO-Snap's 2021 national use and make tables. Links to supporting data files are included.

1 Common Notation

IO-Snap follows the notational conventions of the commodity-by-industry input-output accounting framework shown in Figure 1, adapted from United Nations (1968).

Figure 1: The Commodity-Industry Framework

	Commodities	Industries	Final Demand	Totals
Commodities		U	e	q
Industries	V			g
Primary Inputs		va		
Totals	q'	g'		

- U = the *Use* table: row commodities used by column industries
- V = the *Make* table: column commodities produced by row industries
- e = column final demand activities use of row commodities
- q = column vector of total commodity use
- g = column vector of total industry output
- va = column industry payments to row value added components, typically compensation, payments to governments, and gross operating surplus
- $\mathbf{i}$ is a summing vector of appropriate dimension
- $'$ indicates transpose
- ˆ indicates diagonalization
- i, j => row and column sector subscripts. When used in combination, subscripts denote a from–to relationship, e.g., $z_{i,j}$ denotes a flow from source i to destination j
- r, s, N => superscripts denoting regions r and s, or national variables, N. When used in combination, superscripts denote origin and destination regions, e.g., z^{rs} denotes a flow from region r to region s

2 Using the Leontief Price Model to verify Output Multipliers derived from I-O SNAP's Input-Output Tables

The remainder of this technical document demonstrates how to use the Leontief Price Model to verify the accounting accuracy of IxI input-output tables and multipliers derived from the IO-Snap modeling system. A Google Scholar search on the terms "Leontief Price Model" identifies many related and relevant articles in the literature. This document is intended only to provide an example of the relative ease of using this model to verify the accounting accuracy of IxI output multipliers derived from the IO-Snap modeling system. Users interested in other applications of this modeling framework will want to dig deeper into the related literature. The recommended citation for this document (Kay, 2025) can be found in the Reference section.

2.1 Procedure

2.1.1 Download I-O SNAP's Use and Make Tables

Begin by using the IO-Snap modeling system software to export the national use and make tables. For this example we will be using the 2021 input-output tables. The use table is given in commodity-by-industry (CxI) format and the make table is given in industry-by-commodity (IxC) format. The value-added rows from the use table are given in the industry (I) dimension and the import row from the make table is given in the commodity (C) dimension. All subsequent derivations will be given in terms of the native table dimensions provided by the IO-Snap modeling system.

2.1.2 Calculate the Direct Commodity Requirements (B) Matrix

From the CxI use table, direct commodity requirements can be derived as each cell's share of its column industry output:

$$b_{ij} = \frac{u_{ij}}{g_j} \tag{1}$$

In matrix format, this is known as the direct commodity requirements (B) matrix, where entries in each column show the amount of a commodity or value-added that used by an industry per dollar of output of that industry. This is a commodity-by-industry matrix that can be calculated as follows:

$$B = U\widehat{g}^{-1} \tag{2}$$

2.1.3 Calculate the Market Share (D) Matrix

From the IxC make table, the market share requirements can be derived as each cell's share of its column total, as given below:

$$d_{ij} = \frac{v_{ij}}{q_j} \tag{3}$$

In matrix form this is known as the market share (D) matrix, where entries in each column show, for a given commodity, the proportion of the total output of that commodity produced in each industry. It is assumed that each commodity is produced by the various industries in fixed proportions (industry-technology assumption). This is an industry-by-commodity matrix that can be calculated as follows:

$$D = V\hat{q}^{-1} \tag{4}$$

2.1.4 Calculate the Direct Industry Requirements (A) Matrix

The direct industry requirements (A) matrix is calculated as the product of the D and B matrix, as shown below:

$$A = DB \tag{5}$$

Due to the conformability rule, the product of the D (IxC) matrix and B (CxI) matrix yield an industry-by-industry (IxI) matrix known as the direct industry requirements (A) matrix. Entries in each column of this matrix show the amount of industry output required as an input into the column industry's production function, as given on a per dollar of output basis.

In scalar format each column entry represents a direct input requirement (z_{ij}) as a share of that industry's output (q_{ij}

$$a_{ij} = \frac{z_{ij}}{q_j} \tag{6}$$

2.1.5 Calculate the Leontief Inverse (M) Matrix

When calculated from the direct industry requirements (A) matrix, the Leontief inverse (M) matrix (also known as the multiplier matrix) represents an IxI matrix calculated as the inverse of the ($I - A$) matrix, where (I) is an identity matrix, as shown below:

$$M = [I - A]^{-1} \tag{7}$$

When the derivation of this multiplier matrix is open with respect to households, which happens to be the case in this example, the column sums of the matrix represent Type 1 industry output multipliers.

2.1.6 Convert Commodity Import Shares to Industry Dimension

The last row of the IO-Snap market share (D) matrix is the import row. Because of conformability conditions, import shares are not typically included in the calculation of the direct industry requirements (A) matrix. However, in the IxI framework import shares need to be accounted for within an industry's total cost structure. As such, a separate calculation is required to convert commodity import shares to industry import shares, as follows:

$$m_i^{'} = m_c^{'} B \tag{8}$$

Where $m_i^{'}$ is a row vector of industry import shares calculated as the product of a row vector of commodity import shares ($m_c^{'}$) and the direct commodity requirements (B) matrix. The calculation converts the imports shares in the market share matrix from commodity to industry dimension, which will make them compatible for use in the industry-by-industry (IxI) accounting framework.

Following a more programmatic approach, where import shares may not be calculated as part of the market share (D) matrix, the following approach can be employed to calculate industry import shares. First, the following equation can be used to calculate a row vector of commodity import shares ($m_c^{'}$) from the import row ($m^{'}$) of the make (V) matrix. Following this step, equation (8) can then be used to convert the row vector of commodity import shares to a row vector of industry import shares.

$$m_c^{'} = m^{'} \widehat{q}^{-1} \tag{9}$$

2.1.7 Calculate Total Factor Coefficients

Total factor coefficients are calculated as the sum of all industry value-added coefficients, including industry import coefficients, as follows:

$$l_i = ec_i + t_i + s_i + m_i \tag{10}$$

Where l_i is an industry's total factor coefficient calculated as the sum of that industry's employee compensation coefficient (ec_i), production tax coefficient (t_i), and gross operating surplus coefficient (s_i), as derived from the B matrix, as well as that industry's import coefficient (m_i), as derived in the previous step. In matrix format total factor coefficients are represented as a row vector $l^{'}$ across all industries.

Following a more programmatic approach, where all value-added accounts may be imported as a single matrix, the following approach can be employed to calculate value-added shares. First, the following equation can be used to calculate a row vector of total value-added from the value-added portion of the use (U) matrix.

$$va' = i' VA \tag{11}$$

Where (va') is a row vector of total value-added calculated as the product of a 1x3 summation vector (i') and a 3xI matrix of industry value-added accounts (VA), including employee compensation, taxes on production, and gross operating surplus. Once the row vector of total value-added (va') is derived, the following equation can be used to calculate a row vector of value-added shares (va'_i) from the total value-added row (va') of the use (U) matrix:

$$va'_i = va' \widehat{g}^{-1} \tag{12}$$

Finally, the row vector of industry import shares (m'_i) can be added to the row vector of value-added shares (va'_i) to derive a row vector of total factor coefficients (l'), as follows:

$$l' = va'_i + m'_i \tag{13}$$

2.1.8 Calculate the Leontief Price Model

The Leontief Price model is a total cost accounting model that produces a 1xI unity row vector $\mathbf{i}'$ as the product of a row vector of total factor coefficients l' and the Leontief Inverse or multiplier matrix (M), as follows:

$$\mathbf{i}' = l' M \tag{14}$$

2.1.9 Verify Accounting Accuracy of Output Multipliers

If all production costs are accounted for in the multiplier matrix (M) and total factor coefficients (l'), then the product of this operation will be equal to a 1xI unity row vector containing only 1's ($\mathbf{i}'$). In this application, the interpretation of this result is that all costs are appropriately accounted for in the IxI input-output framework, including any output multipliers derived from this framework. It represents a convenient check to ensure that all necessary data is accounted for in the derivation of the IxI input-output framework and its corresponding output multipliers.

3 Implementation Code

3.1 EViews

The following script provides step-by-step examples of how this procedure can be implemented programmatically in EViews using I-O SNAP's national use and make tables.

1. Load make and use matrices to unstructured workfile. Update file path "C:..."

```
wfcreate US u 69
importmat(name=u) "C:...US 2021 National Use.xlsx" range="National Use!b2:bp70"
importmat(name=g) "C:...US 2021 National Use.xlsx" range="National Use!b74:bp74"
importmat(name=e) "C:...US 2021 National Use.xlsx" range="National Use!bq2:bz70"
importmat(name=va) "C:...US 2021 National Use.xlsx" range="National Use!b71:bp73"
importmat(name=v) "C:...US 2021 National Make.xlsx" range="National Make!b2:br68"
importmat(name=q) "C:...US 2021 National Make.xlsx" range="National Make!b70:br70"
importmat(name=import) "C:...US 2021 National Make.xlsx" range="National Make!b69:br69"
```

2. Create row vector, diagonal matrix, and inverse diagonal matrix of industry output (g)

```
vector grow = g
matrix gdiag = @makediagonal (grow)
matrix ginv = @inverse (gdiag)
```

3. Create row vector, diagonal matrix, and inverse diagonal matrix of commodity output (g)

```
vector qrow = q
matrix qdiag = @makediagonal (qrow)
matrix qinv = @inverse (qdiag)
```

4. Calculate direct commodity requirments matrix (b)

```
matrix b = u*ginv
```

5. Calculate market share matrix (d)

```
matrix d = v*qinv
```

6. Calculate direct industry requirments matrix (a)

```
matrix a = d*b
```

7. Create IxI identity matrix based on number of industries in dataset (identity)

```
matrix identity = @identity(67)
```

8. Calculate IxI multipler matrix (m)

matrix m = @inverse(identity - a)

9. Calculate commdoity import share row (importc)

vector importc = import*qinv

10. Calculate industry import share row (importi)

vector importi = importc*b

11. Create 1x3 summation vector based on number of value-added accounts (sum)

rowvector(3) sum = 1

12. Sum value-added accounts into single row vector (varow)

vector varow = sum*va

13. Calculate value-added shares (vacoef)

vector vacoef = varow*ginv

14. Calculate total value-added coefficients, including industry import shares (l)

vector l = vacoef+importi

15. Calculate Leontief Price Model (i)

vector i = l*m

If all production costs are accounted for in the IxI multiplier framework then this calculation should produce a 1x67 unity row vector containing only 1's.

3.2 Matlab

3.3 Python

4 Links to auxiliary files

References

Kay, D. (2025). Verifying output multipliers with the leontief price model. IO-Snap Technical Document 2025 - 01, EconAlyze LLC, Morgantown, WV 26508. https://econalyze.com.

United Nations (1968). *A System of National Accounts*, volume 2 of *Series F*. United Nations, New York, 3 edition.