Trade Theory & Matrix Algebra

Graham Joncas (戈雷)

1 Introduction to Matrices

In economics, we frequently deal with systems of equations. In undergraduate economics, these are usually limited to two variables: say, an economy with two goods, a game with two strategies, or a portfolio with a 'stock market' asset and a risk-free asset. These are simple enough to solve by hand. However, to analyze economies with many goods, games with many strategies, or portfolios with many assets — that is, to do *real* economics — we must use matrices.

We can represent a system of equations in matrix form as follows.

$$\begin{aligned} 1x + 2y + 3z &= a \\ 4x + 5y + 6z &= b \\ 7x + 8y + 9z &= c \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Here, we take the coefficients of the variables and put them in a box, called a *matrix*. Call it A. This matrix has 3 rows and 3 columns, but in principle the number of rows and columns can be any positive integer. If a matrix has only one row or one column, we call it a *vector*. Since the variables are the same for all three equations, we can put them in a vector $\boldsymbol{x}$. Last, each equation is equal to a number—here called a *scalar*. We can put each of these scalars together to get another vector $\boldsymbol{v}$. Having defined these matrices and vectors, we can express this whole system of equations in a single equation: $A\boldsymbol{x} = \boldsymbol{v}$.

Suppose we know the values in $\boldsymbol{v}$, but not the ones in $\boldsymbol{x}$. In a normal equation $kx = y$, we can just multiply both sides by the inverse of k, that is, $\frac{1}{k}$ or k^{-1}. Under certain conditions, we can also say that a matrix A has an inverse A^{-1}.

When we multiply $k \times \frac{1}{k}$, the result is 1. These cancel out on the left side, leaving only $\frac{1}{k}y$ on the right side. In a similar way, multiplying $A^{-1}A$ or AA^{-1} gives the identity matrix I. Like multiplying a scalar by 1, multiplying any matrix or vector by the identity matrix makes no change.

This means that if we find an inverse A^{-1}, then we can find the values of $\boldsymbol{x}$.

$$A\boldsymbol{x} = \boldsymbol{v} \quad \Rightarrow \quad \boldsymbol{x} = A^{-1}\boldsymbol{v}$$

A 3×1 vector $\boldsymbol{p}$ has 3 rows (going down), and one column (to the side). A vector with the same values as $\boldsymbol{p}$, but instead with one row and 3 columns, is called the *transpose* of $\boldsymbol{p}$, denoted $\boldsymbol{p}'$.

$$\boldsymbol{p} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{p}' = \begin{bmatrix} p & q & r \end{bmatrix}$$

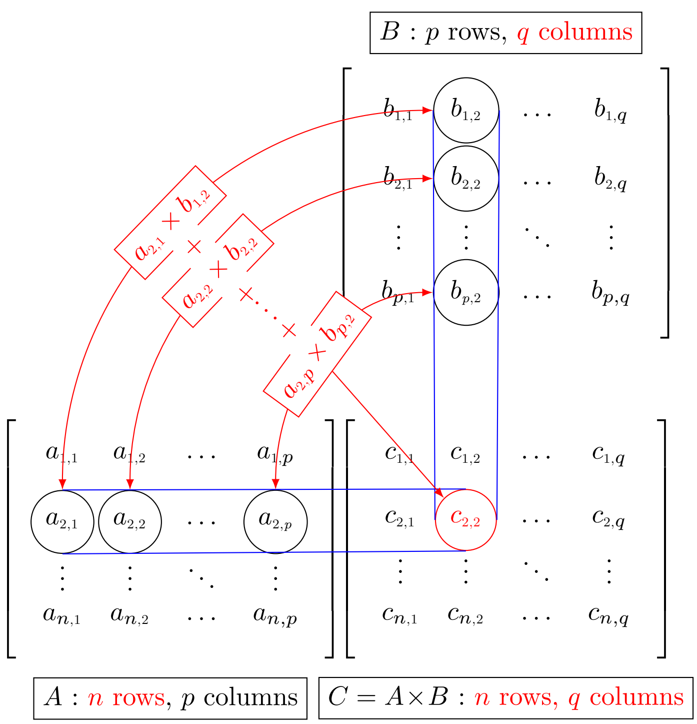
It is standard to represent a matrix as a capital letter, and a vector as a bolded lowercase letter (or sometimes like $\overrightarrow{v}$). Here, we'll add one last bit of notation. A *diagonal matrix* is one whose values are all zeros except for those on the diagonal. If a matrix is diagonal, we'll show this by adding a hat: $\hat{X}$. The identity matrix I is a diagonal matrix as well, but it is so well-known that we will omit the hat.

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \hat{X} = \begin{bmatrix} x_1 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{bmatrix}$$

We can multiply a $n \times p$ matrix A by a $p \times q$ matrix B to get a $n \times q$ matrix C. Each element $c_{i,j}$ of C comes from the ith row of A and jth column of B. Multiply the first element of A's ith row $(a_{i,1})$ by the first element of B's jth column $(b_{1,j})$, then multiply $a_{i,2}$ by $b_{2,j}$ and so on until reaching the final element (p) of each row and column. The sum of all these is $c_{i,j}$. Expressing this mathematically:

$$c_{i,j} = \sum_{x=1}^{p} a_{i,x} b_{x,j}$$

This is the hardest part to understand for those first learning to use matrices. Thus, it's helpful to visualize matrix multiplication using the following diagram.



Let's go through a simple numerical example. This will make it clear that matrix multiplication is just tedious arithmetic, which is why we make computers do it.

$$
\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}
\begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}
=
\begin{bmatrix}
1(2){+}2(8){+}3(14) & 1(4){+}2(10){+}3(16) & 1(6){+}2(12){+}3(18) \\
4(2){+}5(8){+}6(14) & 4(4){+}5(10){+}6(16) & 4(6){+}5(12){+}6(18) \\
7(2){+}8(8){+}9(14) & 7(4){+}8(10){+}9(16) & 7(6){+}8(12){+}9(18)
\end{bmatrix}
$$

$$
= \begin{bmatrix} 60 & 72 & 84 \\ 132 & 162 & 192 \\ 204 & 252 & 300 \end{bmatrix}
$$

Last, a crucial detail is that matrix multiplication is non-commutative: in general, $AB \neq BA$. (An exception is $A^{-1}A = AA^{-1} = I$.) That is, multiplying from the left and multiplying from the right gives different results. We must also be careful to only multiply a $n \times p$ matrix or vector by a $p \times q$ matrix or vector. Thus, we can multiply a 3×3 matrix by a 3×1 vector, but not the other way around. Yet, we can multiply the 1×3 transpose of this vector by a 3×3 matrix.

This introduction is only gives the most basic tools for understanding the following derivations. Most other details (such as computing an inverse) can be done by a computer, but knowing how they work will both help you get a deeper understanding of familiar ideas, and help you use matrices in your own research.

2 The Leontief Matrix

Matrices are essential in trade theory because trade involves many commodities traded by many countries. An input-output (IO) table gives a total output vector $\boldsymbol{x}$, final use matrix Y, intermediate use matrix Z, and value-added vector $\boldsymbol{v}'$.

We can use the vector $\boldsymbol{x}$ to construct a diagonal matrix $\hat{X}$. Let the ith element of $\boldsymbol{x}$ be the ith diagonal of $\hat{X}$ (i.e. the element $\hat{x}_{i,i}$), and leave all other elements as zeros. Inverting this gives matrix $\hat{X}^{-1}$.

$$
\hat{X} = \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_n \end{bmatrix}
\qquad \rightarrow \qquad
\hat{X}^{-1} = \begin{bmatrix} \frac{1}{x_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{x_n} \end{bmatrix}
$$

The Leontief model begins with an input coefficient matrix A, each of whose elements $a_{i,j}$ is the value of inputs from sector i per dollar of output from sector j. Thus, A can be decomposed into the product of Z and the inverse of $\hat{X}$:

$$
A = \begin{bmatrix} \frac{z_{1,1}}{x_1} & \frac{z_{1,2}}{x_2} & \cdots & \frac{z_{1,n}}{x_n} \\ \frac{z_{2,1}}{x_1} & \frac{z_{2,2}}{x_2} & \cdots & \frac{z_{2,n}}{x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{z_{n,1}}{x_1} & \frac{z_{n,2}}{x_2} & \cdots & \frac{z_{n,n}}{x_n} \end{bmatrix}
= \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,n} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n,1} & z_{n,2} & \cdots & z_{n,n} \end{bmatrix}
\begin{bmatrix} \frac{1}{x_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{x_n} \end{bmatrix}
= Z\hat{X}^{-1}
$$

It follows from this that $A\hat{X} = Z$, since $\hat{X}^{-1}\hat{X} = I$.

We begin with an output-side accounting identity: total output ($\hat{X}$) equals final use (Y) plus intermediate use (Z). From here, there are two ways to derive the Leontief inverse matrix $L = (I - A)^{-1}$. For the first, we substitute $A\hat{X}$ for Z:

$$\begin{aligned}
\hat{X} &= Y + Z \\
\hat{X} &= Y + A\hat{X} \\
(I - A)\hat{X} &= Y \\
\hat{X} &= (I - A)^{-1}Y = LY
\end{aligned}$$

For the second method of deriving L, we use power series approximation. Total output ($\hat{X}$) equals final use (Y), plus the inputs that went into final use (AY), plus the inputs that went into those inputs (A^2Y), and so on, in an infinite series.

$$\begin{aligned}
\hat{X} &= Y + Z \\
&= Y + AY + A^2Y + A^3Y + \cdots \\
&= (1 + A + A^2 + A^3 + \cdots)Y \\
&= \left(\frac{1}{I - A}\right)Y = (I - A)^{-1}Y = LY
\end{aligned}$$

The step from the third to the fourth line is a matrix analogue of a geometric series, which you may have seen before in a finance class.

$$1 + r + r^2 + r^3 + ... = \sum_{k=0}^{\infty} r^k = \frac{1}{1 - r}, \text{ for } |r| < 1$$

Under similar conditions (all eigenvalues of A have absolute value smaller than 1), the result likewise holds for an infinite series of matrices.

The elements of the Leontief inverse matrix express the total output, direct and indirect, required to produce \$1 of final goods and services. If we define a vector $\boldsymbol{y}$ (of which Y is a disaggregated version), each of whose elements y_i is the value of final goods produced in country i, then $\boldsymbol{v}'L\boldsymbol{y} =$ global GDP, a scalar.

We can also make sense of L's meaning via the value-added coefficient vector $\boldsymbol{p}$, defined as the value-added created per unit of total output for each good:

$$\boldsymbol{p}' = \begin{bmatrix} \frac{v_1}{x_1} & \frac{v_2}{x_2} & \cdots & \frac{v_n}{x_n} \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \frac{1}{x_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{x_n} \end{bmatrix} = \boldsymbol{v}'\hat{X}^{-1}$$

From the fact that $\boldsymbol{p}' = \boldsymbol{v}'\hat{X}^{-1}$, we see $\boldsymbol{v}' = \boldsymbol{p}'\hat{X} = \boldsymbol{p}'LY$.

Here, we refer to $\boldsymbol{p}'L$ as the 'total value-added multiplier'. This measures the additional value added as a result of producing a given good. So if the value-added multiplier is 2.2 and the final use of a good is \$$y$, then \$$1.2y$ in value is added to products in other industries affected by that good.

3 The Ghosh Matrix

Ghosh's model begins with a direct output coefficient (or allocation coefficient) matrix B, each of whose elements $b_{i,j}$ is the distribution of sector i's outputs across sectors j that purchase inter-industry inputs from i.

$$B = \begin{bmatrix} \frac{z_{1,1}}{x_1} & \frac{z_{1,2}}{x_2} & \cdots & \frac{z_{1,n}}{x_n} \\ \frac{z_{2,1}}{x_1} & \frac{z_{2,2}}{x_2} & \cdots & \frac{z_{2,n}}{x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{z_{n,1}}{x_1} & \frac{z_{n,2}}{x_2} & \cdots & \frac{z_{n,n}}{x_n} \end{bmatrix} = \begin{bmatrix} \frac{1}{x_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{x_n} \end{bmatrix} \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,n} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n,1} & z_{n,2} & \cdots & z_{n,n} \end{bmatrix} = \hat{X}^{-1}Z$$

It also follows that $\hat{X}B = Z$, since $\hat{X}\hat{X}^{-1} = I$.

To derive the Ghosh inverse matrix $G = (I - B)^{-1}$, we begin with an input-side accounting identity: total output ($\hat{X}' = \hat{X}$) equals intermediate use (Z) plus value-added (V). Note that V is disaggregated value-added, as opposed to the vector $\boldsymbol{v}$ in the IO table. Our first method is to substitute $\hat{X}$ ($= \hat{X}'$) for Z.

$$\begin{aligned} \hat{X} &= Z + V \\ \hat{X} &= \hat{X}B + V \\ \hat{X}(I - B) &= V \\ \hat{X} &= V(I - B)^{-1} = VG \end{aligned}$$

Our second method for deriving G also involves an infinite series. Here, total output ($\hat{X}$) equals final value-added (V), plus the value added to the inputs (VB), plus the value added to those inputs (VB^2), and so on to infinity.

$$\begin{aligned} \hat{X} &= V + Z \\ &= V + VB + VB^2 + VB^3 + \cdots \\ &= V(1 + B + B^2 + B^3 + \cdots) \\ &= V\left(\frac{1}{I - B}\right) = V(I - A)^{-1} = VG \end{aligned}$$

We can make sense of the Ghosh matrix via the final products coefficient vector $\boldsymbol{f}$, defined as the value of final goods produced in country i per unit of total output.

$$\boldsymbol{f} = \begin{bmatrix} \frac{y_1}{x_1} \\ \frac{y_2}{x_2} \\ \vdots \\ \frac{y_n}{x_n} \end{bmatrix} = \begin{bmatrix} \frac{1}{x_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{x_n} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \hat{X}^{-1}\boldsymbol{y}$$

From the fact that $\boldsymbol{f} = \hat{X}^{-1}\boldsymbol{y}$, we see $\boldsymbol{y} = \hat{X}\boldsymbol{f} = VG\boldsymbol{f}$. Thus $G\boldsymbol{f}$ acts as another multiplier, establishing a relationship between value-added V and final use $\boldsymbol{y}$.

It turns out that the Leontief and Ghosh matrices are transformations of each other. We know that $A = Z\hat{X}^{-1} \rightarrow AX = Z$ and $B = \hat{X}^{-1}Z \rightarrow XB = Z$. Plugging each into the other, we see that $A = \hat{X}B\hat{X}^{-1}$ and that $B = \hat{X}^{-1}A\hat{X}$. Using the trick that $\hat{X}I\hat{X}^{-1} = I$, we proceed as follows.

$$\begin{aligned}(I - A) &= I - \hat{X}B\hat{X}^{-1} &&= \hat{X}(I - B)\hat{X}^{-1} \\ (I - A)^{-1} &= \left[\hat{X}(I - B)\hat{X}^{-1}\right]^{-1} &&= \hat{X}(I - B)^{-1}\hat{X}^{-1}\end{aligned}$$

That is, $L = \hat{X}G\hat{X}^{-1}$. This further implies that $G = \hat{X}^{-1}L\hat{X}$.

4 Trade Theory & Matrices in Stata

We can compute the empirical Leontief and Ghosh matrices using publicly available data, such as the International Input-Output tables in the WIOD database. The following code is based on the 2014 dataset, downloadable in Stata format. After downloading the dataset and opening it in Stata, we first have to clean it.

`keep if Country=="CHN"`	Drops all observations except China
`keep vCHN* TOT`	Keeps all variables starting with `vCHN`, plus `TOT`
`drop vCHN57- vCHN61`	Drops some unneeded variables
`replace TOT = TOT + 0.001`	Adds 0.001 to all values in `TOT` to prevent zeros

Next is the hardest part: computing the intermediate input coefficient matrix A.

`mkmat vCHN1- vCHN56, matrix (IT)`	Uses IO table data to make a matrix `IT` $= Z$
`mkmat TOT, matrix(TOT)`	Creates a vector `TOT` $= \boldsymbol{x}$
`matrix TI=diag(TOT)`	Defines a matrix `TI` that's the diagonal of `TOT`
`svmat TI`	$\rightarrow$ Saves this matrix as `TI` $= \hat{X}$
`mkmat TI1- TI56, matrix(TI)`	Creates the matrix `TI`
`matrix TIinv = inv(TI)`	Defines a matrix `TIinv` that's the inverse of `TI`
`svmat TIinv`	$\rightarrow$ Saves this matrix as `TIinv` $= \hat{X}^{-1}$
`matrix A=IT*TIinv`	Defines a matrix $A = Z\hat{X}^{-1}$
`svmat A`	$\rightarrow$ Saves this matrix as A

Now that A is defined, computing the Leontief inverse matrix is straightforward:

`matrix I=I(56)`	Defines a 56×56 identity matrix I
`svmat I`	$\rightarrow$ Saves this matrix as I
`matrix L=I-A`	Defines a matrix L by subtracting A from I
`svmat L`	$\rightarrow$ Saves this matrix as $L = (I - A)$
`matrix B = inv(L)`	Defines a matrix B that's the inverse of L
`svmat B`	$\rightarrow$ Saves this matrix as $B = L^{-1} = (I - A)^{-1}$

Finally, we compute the Ghosh matrix by transforming the Leontief matrix.

`mkmat TIinv1- TIinv56, matrix (TIinv)`	Creates the matrix `TIinv` $= \hat{X}^{-1}$
`mkmat B1- B56, matrix(B)`	Creates the matrix $B = (I - A)^{-1}$
`matrix G=TIinv*B*TI`	Defines a matrix $G = \hat{X}^{-1}(I - A)^{-1}\hat{X}$
`svmat G`	$\rightarrow$ Saves this matrix as G