Linear Algebra Lecture Notes

For MATH 1554 at the Georgia Institute of Technology

Greg Mayer

Edition 0.1

Preface

These lecture notes are intended for use in a Georgia Tech undergraduate level linear algebra course, MATH 1554. In this first edition of the notes, the focus is on some of the topics not already covered in the Interactive Linear Algebra text.

Contents

Chapter 1

Applications of Matrix Algebra

1.1 Block Matrices

A **block matrix** is a matrix that is interpreted as having been broken into sections called **blocks**, or **submatrices**. Intuitively, a block matrix can be interpreted as the original matrix that is partitioned into a collection of smaller matrices. For example, the matrix

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

can also be written as a 2×2 partitioned (or block) matrix:

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

where the entries of A are the blocks

$$A_{1,1} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad A_{1,2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A_{2,1} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \quad A_{2,2} = \begin{pmatrix} 2 & 2 & 2 \end{pmatrix}$$

We partitioned our matrix into four blocks, each of which have different dimensions. But the matrix could also, for example, be partitioned into five 4×1 blocks, or four 1×5 blocks. Indeed, matrices can be partitioned into blocks in many different ways, and depending on the application at hand, there can be a partitioning that is useful or needed.

For example, when solving a linear system $A\vec{x} = \vec{b}$ to determine $\vec{x}$, we can construct and row reduce an augmented matrix of the form

$$X = \begin{pmatrix} A & \vec{b} \end{pmatrix}$$

The augmented matrix X consists of two sub-matrices, A and $\vec{b}$, meaning that it can be viewed as a block matrix. Another application of a block matrix arises when using the SVD, which is a popular tool used in data science. The SVD uses a matrix, Σ, of the form

$$\Sigma = \begin{pmatrix} D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Matrix D is a diagonal matrix, and each $\mathbf{0}$ is a zero matrix. Representing Σ in terms of sub-matrices helps us see what the structure of Σ is. Another block matrix arises when introducing a procedure for computing the inverse of an $n \times n$ matrix. To compute the inverse of matrix A, we construct and row reduce the matrix

$$X = \begin{pmatrix} A & I \end{pmatrix}$$

This is an example of a block matrix used in an algorithm. In order to use block matrices in other applications we need to define matrix addition and multiplication with partitioned matrices.

Block Matrix Addition

If $m \times n$ matrices A and B are partitioned in exactly the same way, then the entries of their sum is the sum of their blocks. For example, if A and B are the block matrices

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix}, \quad B = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix}$$

then their sum is the matrix

$$A + B = \begin{pmatrix} A_{1,1} + B_{1,1} & A_{1,2} + B_{1,2} \\ A_{2,1} + B_{2,1} & A_{2,2} + B_{2,2} \end{pmatrix}$$

As long as A and B are partitioned in the same way the addition is calculated block by block.

Block Matrix Multiplication

Recall the **row column** method for matrix multiplication.

Theorem

Let A be $m \times n$ and B be $n \times p$ matrix. Then, the (i, j) entry of AB is

$$\text{row}_i\, A \cdot \text{col}_j\, B.$$

This is the Row Column Method for matrix multiplication.

Partitioned matrices can be multiplied using this method, as if each block were a scalar provided each block has appropriate dimensions so that products are defined.

Example 1: Computing A^2

Block matrices can be useful in cases where a matrix has a particular structure. For example, suppose A is the $n \times n$ block matrix

$$A = \begin{pmatrix} X & \mathbf{0} \\ \mathbf{0} & Y \end{pmatrix}$$

where X and Y are $p \times p$ matrices, $\mathbf{0}$ is a $p \times p$ zero matrix, and $2p = n$. Then

$$A^2 = AA = \begin{pmatrix} X & \mathbf{0} \\ \mathbf{0} & Y \end{pmatrix} \begin{pmatrix} X & \mathbf{0} \\ \mathbf{0} & Y \end{pmatrix} = \begin{pmatrix} X^2 & \mathbf{0} \\ \mathbf{0} & Y^2 \end{pmatrix}$$

Computation of A^2 only requires computing X^2 and Y^2. Taking advantage of the block structure A leads to a more efficient computation than it otherwise would have been with a naive row-column method that does not take advantage of the structure of the matrix.

Example 2: Computing AB

A and B are the matrices

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \end{pmatrix}$$

$$B = \begin{pmatrix} 2 & -1 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} B_{11} \\ B_{21} \end{pmatrix}$$

where

$$A_{11} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_{12} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad B_{11} = \begin{pmatrix} 2 & -1 \\ 0 & -1 \end{pmatrix}, \quad B_{21} = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

If we compute the matrix product using the given partitioning we obtain

$$AB = \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} B_{11} \\ B_{21} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} \end{pmatrix}$$

where

$$A_{11}B_{11} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ 0 & -1 \end{pmatrix}$$

$$A_{12}B_{21} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Therefore

$$AB = A_{11}B_{11} + A_{12}B_{21} = \begin{pmatrix} 2 & -1 \\ 0 & -1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

Computing AB with the row column method confirms our result.

$$AB = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2+0+0 & -1+0+1 \\ 0+0+0 & 0-1+1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

Block Matrix Inversion

In some cases, matrix partitioning can be used to give us convenient expressions for the inverse of a matrix. Recall that the inverse of $n \times n$ matrix A is a matrix B, that has the same dimensions as A and satisfies

$$AB = BA = I$$

where I is the $n \times n$ identity matrix. As we will see in the next example, we can use this equation to construct expressions for the inverse of a matrix.

Example 3: Expression for Inverse of a Block Matrix

Recall, using our formula for a 2×2 matrix,

$$\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}^{-1} = \frac{1}{ac} \begin{pmatrix} c & -b \\ 0 & a \end{pmatrix} = \begin{pmatrix} 1/a & -b/(ac) \\ 0 & 1/c \end{pmatrix} \tag{1.1}$$

provided that $ac \neq 0$. Suppose A, B, and C are invertible $n \times n$ matrices. Suppose we wish to construct an expression for the inverse of the matrix

$$P = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix}$$

To construct the inverse of P, we can write

$$PP^{-1} = P^{-1}P = I_n$$

where P^{-1} is the matrix we seek. If we let P^{-1} be the block matrix

$$P^{-1} = \begin{pmatrix} W & X \\ Y & Z \end{pmatrix}$$

we can determine P^{-1} by solving $PP^{-1} = I$ or $P^{-1}P = I$. Solving $PP^{-1} = I$ gives us:

$$\begin{aligned} I_n &= PP^{-1} \\ \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} &= \begin{pmatrix} A & B \\ 0 & C \end{pmatrix} \begin{pmatrix} W & X \\ Y & Z \end{pmatrix} \\ \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} &= \begin{pmatrix} AW + BY & AX + BZ \\ CY & CZ \end{pmatrix} \end{aligned}$$

The above matrix equation gives us a set of four equations that can be solved to determine W, X, Y, and Z. The block in the second row and first column gives us

$CY = 0$. It was given that C is an invertible matrix, so Y is a zero matrix because

$$\begin{aligned} CY &= 0 \\ C^{-1}CY &= C^{-1}0 \\ IY &= 0 \\ Y &= 0 \end{aligned}$$

Likewise the block in the second row and second column yields $CZ = I$, so

$$\begin{aligned} CZ &= I \\ C^{-1}CZ &= C^{-1}I \\ Z &= C^{-1} \end{aligned}$$

Now that we have expressions for Y and Z we can solve the remaining two equations for W and X. Solving for X gives us the following expression.

$$\begin{aligned} AX + BZ &= 0 \\ AX + BC^{-1} &= 0 \\ AX &= -BC^{-1} \\ A^{-1}AX &= -A^{-1}BC^{-1} \\ X &= -A^{-1}BC^{-1} \end{aligned}$$

Solving for W:

$$\begin{aligned} AW + BY &= I \\ AW + B0 &= I \\ A^{-1}AW &= A^{-1}I \\ W &= A^{-1} \end{aligned}$$

We now have our expression for P^{-1}:

$$P^{-1} = \begin{pmatrix} W & X \\ Y & Z \end{pmatrix} = \begin{pmatrix} A^{-1} & -A^{-1}BC^{-1} \\ 0 & C^{-1} \end{pmatrix}$$

Note that in the special case where $n = 2$ that each of the blocks are scalars and our expression is equivalent to Equation (1.1).

Summary

In this section we used partitioned matrices to solve problems regarding matrix invertibility and matrix multiplication. Partitioned matrices can be multiplied using this method, as if each block were a scalar provided each block has appropriate dimensions so that products are defined. They can be used for example when dealing with large matrices that have a known structure where it is more convenient to describe the structure of a matrix in terms of its blocks. Although not part of this text, matrix partitioning can be used to help derive new algorithms because they give a more concise representation of a matrix and of operations on matrices.

Exercises

1. Suppose $A = \begin{pmatrix} Y & X \end{pmatrix} \begin{pmatrix} X & 0 \\ Y & Z \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$. Which of the following could A be equal to?

   a) $A = YX^2 + XYX + XZY$

   b) $A = 2X + XZY$

   c) $A = YX^2 + X + Z$

2. A, B, and C are $n \times n$ invertible matrices. Construct expressions for X and Y in terms of A, B, and C.

$$\begin{pmatrix} 0 & X & 0 \\ A & 0 & Y \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & A \\ A & 0 \end{pmatrix} = \begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix}$$

3. Suppose A, B and C are invertible $n \times n$ matrices, and

$$P = \begin{pmatrix} A & 0 \\ B & C \end{pmatrix}$$

   Give an expression for P^{-1} in terms of A, B, and C.

1.2 The LU Factorization

To solve a linear system of the form $A\vec{x} = \vec{b}$ we could use row reduction or, in theory, calculate A^{-1} and use it to determine $\vec{x}$ with the equation

$$\vec{x} = A^{-1}\vec{b}$$

But computing A^{-1} requires the computation of the inverse of an $n \times n$ matrix, which is especially difficult for large n. It is more practical to solve $A\vec{x} = \vec{b}$ with row reductions (i.e. - Gaussian Elimination). But it turns out that there are more efficient methods, especially when n is large.

One method for solving linear systems that relies on what is referred to as a matrix factorizations. A **matrix factorization**, or **matrix decomposition** is a factorization of a matrix into a product of matrices. Factorizations can be useful for solving $A\vec{x} = \vec{b}$, or for understanding the properties of a matrix.

In this section, we factor a matrix into **lower** and into **upper** triangular matrices to construct what is known as the LU factorization that is used to solve linear systems in a systematic and efficient method. Before we introduce the LU factorization, we will first need to introduce lower and upper triangular matrices.

Triangular Matrices

Before we introduce the LU factorization, we need to first define upper and lower triangular matrices.

Upper and Lower Triangular Matrices

Suppose that the entries of $m \times n$ matrix A are $a_{i,j}$. Then A is **upper triangular** if $a_{i,j} = 0$ for $i > j$. Matrix A is **lower triangular** if $a_{i,j} = 0$ for $i < j$.

As an example, all of the matrices below are in upper triangular form.

$$\begin{pmatrix} 1 & 5 & 0 \\ 0 & 2 & 4 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 2 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Notice how all of the entries below the main diagonal are zero, and the entries on and above the main diagonal can be anything. Likewise, examples of lower triangular matrices are below.

$$\begin{pmatrix} 1 & 0 & 0 \\ 3 & 2 & 0 \end{pmatrix}, \quad \begin{pmatrix} 3 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 1 & 4 \\ 0 & 1 \\ 2 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Again, note that our definition for an upper triangular matrix does not specify what the entries on or above the main diagonal need to be. Some or all of the entries above the main diagonal can, for example, be zero. Likewise the entries on and below the main diagonal of a lower triangular matrix do not have to have specific values.

The LU Factorization

After stating a theorem that gives the LU decomposition, we will give an algorithm for constructing the LU factorization. We will then see how we can use the factorization to solve a linear system.

Theorem: The LU Factorization

If A is an $m \times n$ matrix that can be row reduced to echelon form without row exchanges, then $A = LU$, where L is a lower triangular $m \times m$ matrix with 1's on the diagonal, and U is an echelon form of A.

Proof

To prove the theorem above we will first show that we can write $A = LU$ where L is an invertible matrix, and U is an echelon form of A.

Suppose that $m \times n$ matrix A can be reduced to echelon form U with p elementary row operations that only add a multiple of a row to another row that is below it. Then each row operation can be performed by multiplying A with p elementary matrices.

$$E_p E_{p-1} \cdots E_3 E_2 E_1 A = U \tag{1.2}$$

If we let $L^{-1} = E_p E_{p-1} \cdots E_3 E_2 E_1$, then

$$L^{-1} A = U \tag{1.3}$$

Note that $L^{-1} = E_p E_{p-1} \cdots E_3 E_2 E_1$ is invertible because elementary matrices are invertible. Therefore L^{-1} can be reduced to the identity with a sequence of row operations. Moreover, if we multiply Equation (1.3) by L we obtain:

$$LL^{-1}A = LU \quad \Rightarrow \quad A = LU$$

Therefore A has the decomposition $A = LU$ where U is an echelon form of A and L is an invertible $m \times m$ matrix. To show that L is lower triangular, recall from equations (1.2) and (1.3) that

$$L^{-1} = E_p E_{p-1} \cdots E_3 E_2 E_1$$

Each elementary matrix E_i is lower triangular because to reduce A to U we only used one type of row operation: adding a multiple of a row to a row below it, so each E_i is a lower triangular matrix. It can also be shown that the product of two lower-triangular matrices is a lower triangular matrix, and the inverse of a lower triangular matrix is lower triangular. This implies that both L^{-1} and L will be lower-triangular.

■

Constructing the LU Factorization

To construct the LU factorization of a matrix we must first apply a sequence of row operations to A in order to reduce A to U. Equation (1.3) gives us that

$$E_p E_{p-1} \cdots E_3 E_2 E_1 A = L^{-1} A = U, \quad \text{where } L^{-1} = E_p E_{p-1} \cdots E_3 E_2 E_1$$

But if $L^{-1} L = I$, then then the sequence of row operations that reduce A to U will reduce A to I. This gives us an algorithm for constructing the LU factorization.

Algorithm: Constructing the LU Factorization of a Matrix

Suppose A is an $m \times n$ matrix that can be row reduced to echelon form without row exchanges. To construct the LU factorization:

1. reduce A to an echelon form U by a sequence of row replacement operations, if possible

2. place entries in L such that the sequence of row operations that reduces A to U will reduce L to I

Note that the above procedure will work for any $m \times n$ matrix that can be reduced to echelon form without row exchanges. Meaning that we do not need A to be square or invertible to construct its LU factorization.

Example 1: LU of a 3×2 Matrix

In this example we construct LU factorizations of the following matrix.

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 10 \\ 0 & 12 \end{pmatrix}$$

Because A is a 3×2 matrix, the LU factorization has the form

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{pmatrix} \begin{pmatrix} * & * \\ 0 & * \\ 0 & 0 \end{pmatrix} \tag{1.4}$$

Each $*$ represents an entry that we need to compute the value of. To reduce A to U we apply a sequence of row replacement operations as shown below.

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 10 \\ 0 & 12 \end{pmatrix} \sim \begin{pmatrix} 1 & 3 \\ 0 & 4 \\ 0 & 12 \end{pmatrix} \sim \begin{pmatrix} 1 & 3 \\ 0 & 4 \\ 0 & 0 \end{pmatrix} = U$$

Matrix U is the echelon form of A that we need for the LU factorization. We next construct L so that the row operations that reduced A to U will reduce L to I. Our row operations were:

$$R_2 - 2R_1 \to R_2 \quad \text{and} \quad R_3 - 3R_2 \to R_3$$

With these two row operations, we see that L must be the matrix:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

Note that the row operations $R_2 - 2R_1 \to R_2$ and $R_3 - 3R_2 \to R_3$ applied to L will give us the identity. The LU factorization of A is

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 0 & 4 \\ 0 & 0 \end{pmatrix}$$

Solving Linear Systems with the LU Factorization

Our motivation for introducing the LU factorization was to introduce an efficient method for solving linear systems. Given rectangular matrix A and vector $\vec{b}$, we wish to use the LU factorization of A to solve $A\vec{x} = \vec{b}$ for $\vec{x}$. A procedure for doing so is below.

Algorithm

To solve $A\vec{x} = \vec{b}$ for $\vec{x}$:

1. Construct the LU decomposition of A to obtain L and U.
2. Set $U\vec{x} = \vec{y}$. Forward solve for $\vec{y}$ in $L\vec{y} = \vec{b}$.
3. Backwards solve for $\vec{x}$ in $U\vec{x} = \vec{y}$.

Example 2: Solving a Linear System With LU

In this example we will solve the linear system $A\vec{x} = \vec{b}$ given the LU decomposition of A.

$$A = LU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 0 \end{pmatrix}$$

We first set $U\vec{x} = \vec{y}$ and solve $L\vec{y} = \vec{b}$. Reducing the augmented matrix $(L\,|\,\vec{b})$ gives us:

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 2 \\ 1 & 1 & 0 & 0 & 3 \\ 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 3 & 1 & 0 \end{array}\right) \sim \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 3 & 1 & 0 \end{array}\right) \sim \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 \end{array}\right) \sim \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right)$$

Therefore, $\vec{y}$ is the vector

$$\vec{y} = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

We now solve $U\vec{x} = \vec{y}$.

$$\left(\begin{array}{ccc|c} 1 & 4 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 4 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

The solution to the linear system, $\vec{x}$, is the vector

$$\vec{x} = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}$$

Final Notes on The LU Factorization

In our treatment of the LU factorization we constructed the LU decomposition using the following process.

1. reduce A to an echelon form U by a sequence of row replacement operations, if possible
2. place entries in L such that the same sequence of row operations reduces L to I

Certainly there is much more to the LU factorization than what was presented in this section. There are for example other methods for constructing $A = LU$

that you may encounter in future courses or project you are working on. In our approach, the only row operation we use to construct L and U is to replace a row with a multiple of a row above it. Multiplying a row by a non-zero scalar is not needed, but more importantly, we cannot swap rows. More advanced linear algebra and numerical analysis courses would address this significant limitation.

Exercises

1. Construct the LU Factorizations for the following matrices.

   a) $A = \begin{pmatrix} -1 & 5 & 3 \\ 1 & -10 & -3 \end{pmatrix}$

   b) $A = \begin{pmatrix} 1 & 5 \\ 2 & 10 \\ 0 & 60 \end{pmatrix}$

   c) $A = \begin{pmatrix} 2 & 1 & 0 \\ 4 & 3 & 1 \\ 0 & -1 & 2 \end{pmatrix}$

2. Show that the product of two $n \times n$ lower triangular matrices is lower triangular.

3. Show that the inverse of an $n \times n$ lower triangular matrix is also $n \times n$ and lower triangular.

1.3 The Leontif Input-Output Model

Input–output models are used in economics to model the inter-dependencies between different sectors of an economy. Wassily Leontief (1906–1999) is credited with developing the type of analysis that we explore in this chapter. His work on this model earned a Nobel Prize in Economics.

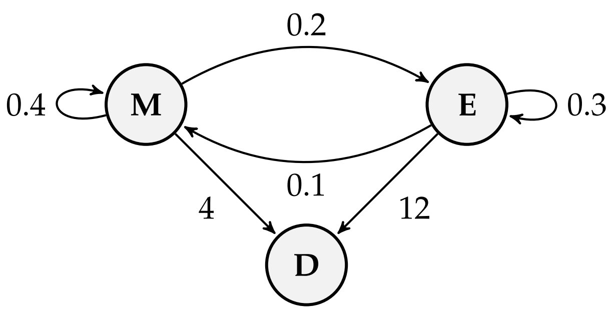
The input-output model assumes that there are sectors in an economy that produce a set of desired products to meet an external demand. The model also assumes that the sectors themselves will also demand a portion of the output that the sectors produce. If the sectors produce exactly the number of units to meet the external demand, then we have the equation

$$(\text{sector output}) - (\text{internal consumption}) = (\text{external demand})$$

In this section we will see that this equation is a linear system that can be solved to determine the output the economy needs to produce to meet the external demand.

Example 1: The Internal Consumption Matrix

Suppose an economy that has two sectors: manufacturing (M) and energy (E). Both of the sectors produce an output to meet an external demand (D) for their products. Sectors M and E also require output from each other to produce their output. The way in which they do so is described in the diagram below.



The numbers in the above diagram can be interpreted as follows.

- For every 100 units that sector M creates, M requires 40 units from M and 10 units from E.
- For every 100 units that sector E creates, E requires 20 units from M and 30 units from E.
- An external demand (D) requires 4 units from M and 12 units from E.

In other words, if M were to create x_M units, then M would consume $0.4x_M$ units from M and $0.1x_M$ units from E. The consumption from sector M could be represented with a vector.

$$\text{consumption from M} = \begin{pmatrix} 0.4x_M \\ 0.1x_M \end{pmatrix} = \frac{x_M}{10}\begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

Likewise, the consumption from sector E would be

$$\text{consumption from E} = \frac{x_E}{10}\begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Adding these vectors together gives us the total internal consumption from both sectors.

$$\begin{aligned} \text{total internal consumption} &= \frac{x_M}{10}\begin{pmatrix} 4 \\ 1 \end{pmatrix} + \frac{x_E}{10}\begin{pmatrix} 3 \\ 2 \end{pmatrix} \\ &= \frac{1}{10}\begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}\begin{pmatrix} x_M \\ x_E \end{pmatrix} \\ &= C\vec{x}, \quad \text{where } \ C = \frac{1}{10}\begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}, \quad \vec{x} = \begin{pmatrix} x_M \\ x_E \end{pmatrix} \end{aligned}$$

Matrix C is called the **consumption matrix**. Typically its entries are between 0 and 1, and the sum of the entries in each column of C will be less than 1. Vector $\vec{x}$ is the output of the sectors. If the sectors produce exactly the number of units to meet the external demand, then we have the equation

$$(\text{sector output}) - (\text{internal consumption}) = (\text{external demand}) \tag{1.5}$$

$$\vec{x} - C\vec{x} = \vec{d} \tag{1.6}$$

In our example, vector $\vec{d} = \begin{pmatrix} 4 \\ 12 \end{pmatrix}$, and $\vec{x} - C\vec{x} = (I - C)\vec{x}$. This simplifies Equation (1.6) to

$$(I - C)\vec{x} = \vec{d} \tag{1.7}$$

$$\left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{1}{10} \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix} \right) \begin{pmatrix} x_M \\ x_E \end{pmatrix} = \begin{pmatrix} 4 \\ 12 \end{pmatrix} \tag{1.8}$$

$$\begin{pmatrix} 0.6 & -0.2 \\ -0.1 & 0.7 \end{pmatrix} \begin{pmatrix} x_M \\ x_E \end{pmatrix} = \begin{pmatrix} 4 \\ 12 \end{pmatrix} \tag{1.9}$$

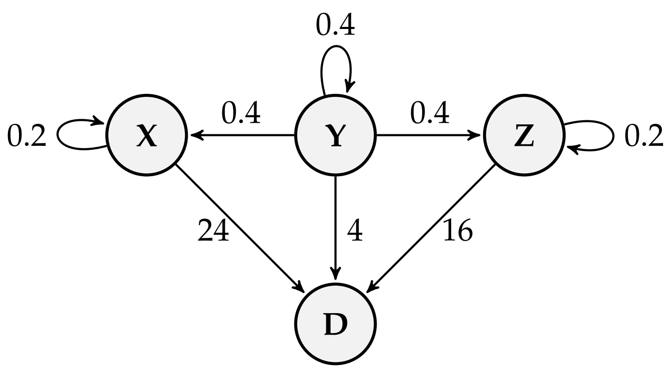
This is a linear system with two equations, whose solution gives us the output vector that balances production with demand. Expressing the system as an augmented matrix and using row operations yields the solution as shown below.

$$\left(\begin{array}{cc|c} .6 & -.2 & 4 \\ -0.1 & 0.7 & 12 \end{array} \right) \sim \left(\begin{array}{cc|c} -1 & 7 & 120 \\ 6 & -2 & 40 \end{array} \right) \sim \left(\begin{array}{cc|c} -1 & 7 & 120 \\ 0 & 40 & 760 \end{array} \right) \sim \left(\begin{array}{cc|c} 1 & 0 & 13 \\ 0 & 1 & 19 \end{array} \right)$$

The unique solution to this linear system is $\vec{x} = \begin{pmatrix} 13 \\ 19 \end{pmatrix}$. This is the output that sectors M and E would need to produce to meet the external demand exactly.

Example 2: An Economy with Three Sectors

Suppose an economy that has three sectors: X, Y, and Z. Each of these sectors produce an output to meet an external demand (D) for their products. The way in which they do so is described in the diagram below.



The external demand, D, is requiring 24 units from X, 4 units from Y, and 16 units from Z. Our goal is to determine how many units the sectors need to produce in order to satisfy this demand, while also accounting for internal consumption.

If Sector X were to create x_X units, then it would consume $0.2x_X$ units from X and $0.4x_X$ units from Y. This consumption could be represented by the vector

$$\text{consumption from Sector X} = \begin{pmatrix} 0.2x_X \\ 0.4x_X \\ 0x_X \end{pmatrix} = \frac{x_X}{10}\begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix}$$

Likewise, the consumption from the other two sectors are

$$\text{consumption from Sector Y} = \frac{x_Y}{10}\begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix}$$

$$\text{consumption from Sector Z} = \frac{x_Z}{10}\begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix}$$

Adding these three vectors together gives us the total internal consumption from all sectors and the consumption matrix C.

$$\begin{aligned} \text{total internal consumption} &= \frac{x_X}{10}\begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix} + \frac{x_Y}{10}\begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix} + \frac{x_Z}{10}\begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix} \\ &= \frac{1}{10}\begin{pmatrix} 2 & 0 & 0 \\ 4 & 4 & 4 \\ 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} x_X \\ x_Y \\ x_Z \end{pmatrix} \\ &= C\vec{x}, \quad \text{where} \quad C = \frac{1}{10}\begin{pmatrix} 2 & 0 & 0 \\ 4 & 4 & 4 \\ 0 & 0 & 2 \end{pmatrix}, \quad \vec{x} = \begin{pmatrix} x_X \\ x_Y \\ x_Z \end{pmatrix} \end{aligned}$$

Each of the sectors in our economy are producing units to satisfy an external demand. The difference between the output and the internal consumption will represent the number of units produced to meet external demand.

$$\begin{aligned} \text{remaining units to meed demand} &= (\text{sector output}) - (\text{internal consumption}) \\ &= \vec{x} - C\vec{x} \\ &= (I - C)\vec{x} \end{aligned}$$

If the sectors are to meet the needs of the external demand exactly, the demand would need to equal the number of units produced after internal consumption is taken into account. That is, we need that

$$(I - C)\vec{x} = \vec{d}$$

This is a linear system that can be solved for the output vector, $\vec{x}$. This could be computed using an augmented matrix.

$$\begin{aligned}
\left(I - C \,\middle|\, \vec{d} \right) &= \left(\begin{array}{ccc|c} 0.8 & 0 & 0 & 24 \\ -0.4 & 0.6 & -0.4 & 4 \\ 0 & 0 & 0.8 & 16 \end{array}\right) \\
&\sim \left(\begin{array}{ccc|c} 8 & 0 & 0 & 240 \\ -4 & 6 & -4 & 40 \\ 0 & 0 & 8 & 160 \end{array}\right) \\
&\sim \left(\begin{array}{ccc|c} 1 & 0 & 0 & 30 \\ -4 & 6 & -4 & 40 \\ 0 & 0 & 1 & 20 \end{array}\right) \\
&\sim \left(\begin{array}{ccc|c} 1 & 0 & 0 & 30 \\ 0 & 1 & 0 & 40 \\ 0 & 0 & 1 & 20 \end{array}\right)
\end{aligned}$$

A helpful trick when reducing these matrices by hand is to multiply each row by 10 to make the algebra a bit less tedious. The above augmented matrix is in row reduced echelon form, and indicates that the desired output is

$$\vec{x} = \begin{pmatrix} 30 \\ 40 \\ 20 \end{pmatrix}$$

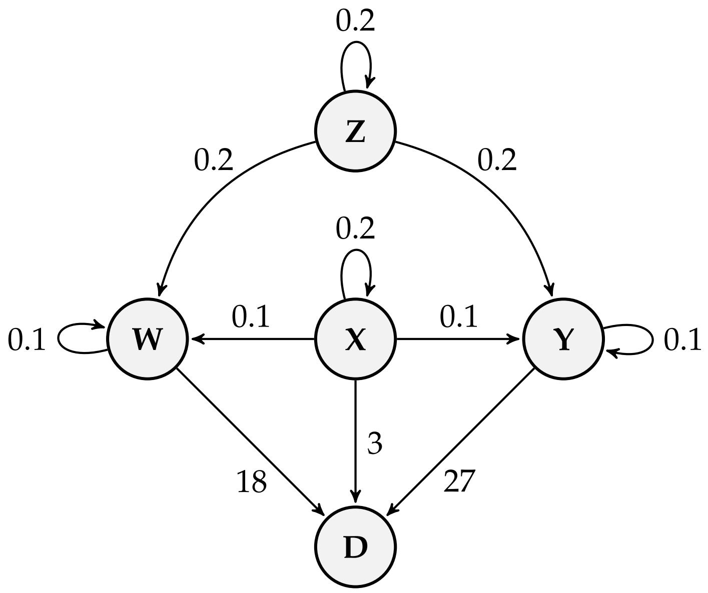
Exercises

1. Consider the production model $\vec{x} = C\vec{x} + \vec{d}$ for an economy with two sectors, where $C = \begin{pmatrix} .0 & .5 \\ .6 & .2 \end{pmatrix}$, and $\vec{d} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$.

   a) Construct the augmented matrix that can be used to calculate $\vec{x}$.

b) Solve your linear system for $\vec{x}$.

2. A model for an economy consists of four sectors, W, X, Y, and Z, and an external demand, D. The relationships between them are given in the diagram below.



Sector Z provides resources to the other sectors internally. There is no external demand from D for the output from Z.

a) Construct the augmented matrix which can be used to solve the system for the output that would meet the external demand exactly while accounting for internal consumption between the four sectors.

b) Solve your augmented matrix to determine the desired output vector.

1.4 2D Computer Graphics

Linear transformations are often used in computer graphics to simulate the motion of an object. They can be modeled with a matrix-vector product of the form

$$T(\vec{x}) = A\vec{x}$$

where $\vec{x}$ is a vector that represents a point that is transformed to the vector $A\vec{x}$. The matrix-vector product $A\vec{x}$ is a transformation that acts on the vector $\vec{x}$ to produce a new vector, $\vec{b} = A\vec{x}$, and if we set the function $T(\vec{x})$ to be

$$T(\vec{x}) = A\vec{x} = \vec{b}$$

then T maps the vector $\vec{x}$ to vector $\vec{b}$. The nature of the transform is described by matrix A.

Translations are a type of transformation needed in computer graphics. But translations are not a linear transformation because they do not leave the origin fixed. How might we use matrix multiplication in order to perform such transformations? In this section we answer this question by introducing homogeneous coordinates, which allow for more general transformations to be computed with linear algebra.

Homogeneous Coordinates

Homogeneous coordinates are a tool that can be used to model translations.

Definition: Homogeneous Coordinates in $\mathbb{R}^2$

Each point (x, y) in $\mathbb{R}^2$ can be identified with the point $(x, y, 1)$, on the plane in $\mathbb{R}^3$ that lies 1 unit above the xy-plane.

For example, a translation of the form $(x, y) \to (x + h, y + k)$ is a transformation. The parameters h and k adjust the location of the point (x, y) after the transformation. This transform can be represented as a matrix multiplication with homogeneous coordinates in the following way.

$$\begin{pmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x+h \\ y+k \\ 1 \end{pmatrix}$$

The first two entries can be extracted from the output of the transform to obtain the coordinate of the translated point. The following examples demonstrate how homogeneous coordinates can be used to create more general transforms.

Example 1: A Composite Transform with Translation

Suppose the transformation $T(\vec{x})$ reflects points in $\mathbb{R}^2$ across the line $x_2 = x_1$ and then translates them by 2 units in the x_1 direction and 3 units in the x_2 direction. In this example we will use homogeneous coordinates to construct a matrix A so that $T = A\vec{x}$.

With homogeneous coordinates the point (x, y) may be represented by the vector

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Points in $\mathbb{R}^2$ can be reflected across the line $x_2 = x_1$ using the standard matrix

$$A_r = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

With homogeneous coordinates our point is represented with a vector in $\mathbb{R}^3$, so we use the block matrix

$$A_1 = \begin{pmatrix} A_r & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The symbol **0** denotes a matrix of zeroes. In this case, either a 1×2 matrix or a 2×1 matrix. Then the matrix-vector product below produces the needed transformation.

$$A_1\vec{x} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} y \\ x \\ 1 \end{pmatrix}$$

Note that the x_1 and x_2 coordinates have been swapped, as required for the reflection through the line $x_2 = x_1$. The matrix below will perform the translation we need.

$$A_2 = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

The product below will apply the translation, of 2 units in the x_1 direction and 3 units in the x_2 direction, to the reflected point.

$$T(\vec{x}) = A_2(A_1\vec{x}) = A_2A_1\vec{x} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y \\ x \\ 1 \end{pmatrix} = \begin{pmatrix} y+2 \\ x+3 \\ 1 \end{pmatrix}$$

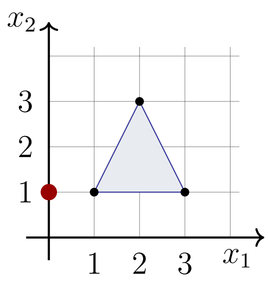
Therfore, our standard matrix is

$$A = A_2A_1 = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

Example 2: Rotation About the Point $(0, 1)$

Triangle S is determined by the points $(1, 1), (2, 3), (3, 1)$. Transform T rotates these points by $\pi/2$ radians counterclockwise about the point $(0, 1)$. Our goal is to use matrix multiplication to determine the image of S under T.

A sketch of the triangle before and after the rotation is in the diagram below.



We need a way to calculate the locations of the points after the transformation. The rotation can be calculated by first representing each point by a vector in homogeneous coordinates, and then multiplying the vectors by a sequence of matrices that perform the needed transformation. The transformations will first shift the points in a way so that the rotation point is about the origin. We will then rotate about the origin by the desired about. And then we move the rotated points up by one unit to account for the initial translation.

Step 1: Shift Points Down by 1 Unit

In homogeneous coordinates our three points can be represented by the vectors below.

$$\vec{a} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}, \quad \vec{c} = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}$$

Multiplying each vector by the matrix

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

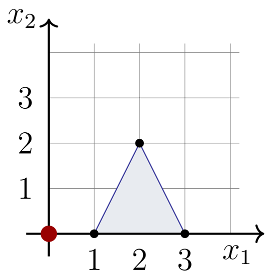
shifts the points down by one unit.

$$\vec{a} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \to A_1\vec{a} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$\vec{b} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \to A_1\vec{b} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

$$\vec{c} = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \to A_1\vec{c} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

Note the difference between the input and output vectors. The second entry of the output vectors is one less than their corresponding entries in the input vectors. Our translated triangle and rotation point is shown below.



With this transform, the rotation point also moves down one unit, from $(0, 1)$ to the origin $(0, 0)$.

Step 2: Rotate About $(0, 0)$
Rotating the translated points by $\pi/2$ radians about the origin can be calulated by multiplying the three vectors by the matrix

$$A_2 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This gives us three new points.

$$\vec{a} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \to A_2 A_1 \vec{a} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$
$$\vec{b} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \to A_2 A_1 \vec{b} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}$$
$$\vec{c} = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \to A_2 A_1 \vec{c} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}$$

Finally, to undo the initial translation that placed the rotation point at the origin, we need to translate our points up by one unit.

Step 3: Translate Points Up One Unit
Translating the data up by one unit can be accomplished by multiplying the three vectors by the matrix

$$A_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

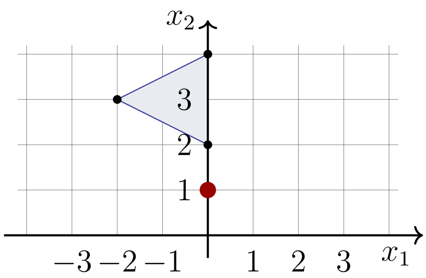
This gives us three new points.

$$\vec{a} = \begin{pmatrix}1\\1\\1\end{pmatrix} \to A_3A_2A_1\vec{a} = \begin{pmatrix}1&0&0\\0&1&1\\0&0&1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix} = \begin{pmatrix}0\\2\\1\end{pmatrix}$$

$$\vec{b} = \begin{pmatrix}2\\3\\1\end{pmatrix} \to A_3A_2A_1\vec{b} = \begin{pmatrix}1&0&0\\0&1&1\\0&0&1\end{pmatrix}\begin{pmatrix}-2\\2\\1\end{pmatrix} = \begin{pmatrix}-2\\3\\1\end{pmatrix}$$

$$\vec{c} = \begin{pmatrix}3\\1\\1\end{pmatrix} \to A_3A_2A_1\vec{c} = \begin{pmatrix}1&0&0\\0&1&1\\0&0&1\end{pmatrix}\begin{pmatrix}0\\3\\1\end{pmatrix} = \begin{pmatrix}0\\4\\1\end{pmatrix}$$

Our rotated and translated triangle is shown below.



Therefore the standard matrix that performs a rotation by $\pi/2$ degrees about $(0,1)$ is the matrix

$$A = A_3A_2A_1 = \begin{pmatrix}1&0&0\\0&1&1\\0&0&1\end{pmatrix}\begin{pmatrix}0&-1&0\\1&0&0\\0&0&1\end{pmatrix}\begin{pmatrix}1&0&0\\0&1&-1\\0&0&1\end{pmatrix} = \begin{pmatrix}0&-1&1\\1&0&1\\0&0&1\end{pmatrix}$$

Our result can be verified by calculating $A\vec{a}$, $A\vec{b}$, or $A\vec{c}$.

Example 3: A Reflection Through The Line $x_2 = x_1 + 3$

In this example we construct the 3×3 standard matrix, A, that uses homogeneous coordinates to reflect points in $\mathbb{R}^2$ across the line $x_2 = x_1 + 3$. We will confirm that our results are correct by calculating $T(\vec{x}) = A\vec{x}$ for any point $\vec{x}$ that uses homogeneous coordinates.

The standard matrix A will be the product of three matrices that translate and reflect points using homogeneous coordinates. The first matrix will translate points in some way so that the line about which we are reflecting will pass through the origin. We can use

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{pmatrix}$$

This matrix will shift points down three units so that the line $x_2 = x_1 + 3$ will pass through the origin. Note that at this point we could have also used a matrix that, for example, shifts to the right by three units. The second matrix will reflect points through the shifted line, which is $x_2 = x_1$. Recall that the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

will reflect vectors in $\mathbb{R}^2$ through the line $x_2 = x_1$. This is because any point with coordinates (x_1, x_2) can be represented with the vector

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

and

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}$$

The point (x_1, x_2) is mapped to (x_2, x_1), which is a reflection through the line $x_2 = x_1$ in $\mathbb{R}^2$. The standard matrix for this transformation in homogeneous coordinates is

$$A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Our final transformation shifts points back up by three units to undo the initial translation.

$$A_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

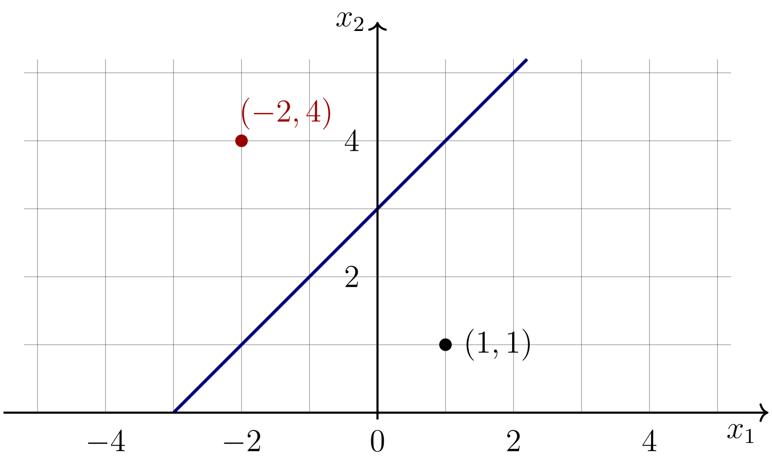
The standard matrix for the transformation that reflects points in $\mathbb{R}^2$ across the line $x_2 = x_1 + 3$ is

$$A = A_3 A_2 A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -3 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

We can check whether our work is correct by transforming any point (x_1, x_2) with the above standard matrix. For example, the point $(1, 1)$ is transformed by calculating

$$T(\vec{x}) = A\vec{x} = \begin{pmatrix} 0 & 1 & -3 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 4 \\ 1 \end{pmatrix}$$

The reflected point is $(-2, 4)$. The line of reflection, initial point, and the reflected point are shown below.

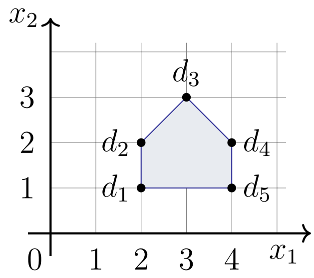


The Data Matrix

The examples in this section have only involved a small number points that need to be transformed. For problems involving many points, it may be more convenient to represent the points in what we refer to as a **data matrix**. For example, the shape in the figure below is determined by five points, or vertices, $d_1, d_2, \ldots, d_5$. Their respective homogeneous coordinates can be stored in the columns of a matrix, D.

$$D = \begin{pmatrix} \vec{d}_1 & \vec{d}_2 & \vec{d}_3 & \vec{d}_4 & \vec{d}_5 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

For our purposes, the order in which the points are placed into D is arbitrary.



In the previous examples we applied a transform with a matrix-vector multiplication. With a data matrix we can use a similar approach. Recall that the product of two matrices A and D, is defined as

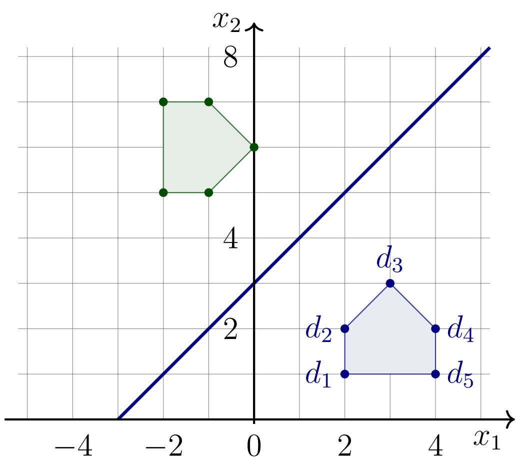
$$AD = A\begin{pmatrix} \vec{d}_1 & \vec{d}_2 & \cdots & \vec{d}_p \end{pmatrix} = \begin{pmatrix} A\vec{d}_1 & A\vec{d}_2 & \cdots & A\vec{d}_p \end{pmatrix}$$

where $\vec{d}_1, \vec{d}_2, \cdots, \vec{d}_p$ are the columns of D. In other words, can perform the transformation on our data by computing AD, which transforms each column independently of the others.

For example, applying the transform in the previous example will reflect our shape through the line $x_2 = x_1 + 3$. The transformation is found by computing

$$AD = \begin{pmatrix} 0 & 1 & -3 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -2 & -1 & 0 & -1 & -2 \\ 5 & 5 & 6 & 7 & 7 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Extracting the first two entries of each column of the result gives us the transformed points (green), as shown in the figure below.



Exercises

1. Construct the standard matrices for the following transforms.

   a) The standard matrix of the transform $\vec{x} \to A\vec{x}$ that reflects points in $\mathbb{R}^2$ across the line $x_1 = k$.

   b) The standard matrix of the transform $\vec{x} \to A\vec{x}$ that rotates points in $\mathbb{R}^2$ about the point $(1, 1)$ and then reflects points through the the line $x_2 = 1$.

1.5 3D Computer Graphics

Results from the previous section on 2D graphics have a natural extension to three dimensions. In this section we extend the data matrix and homogeneous coordinates to three dimensions. This will allow us to model translations and composite transforms involving many points with matrix multiplication.

Rotations in 3D

Rotations about the origin are linear transforms. Because they are linear they can be expressed in the form $T(\vec{x}) = A\vec{x}$ where A is a 3×3 matrix, and we can obtain the columns of matrix A by transforming the standard vectors

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

We will use the convention that a positive rotation is in the counterclockwise direction when looking toward the origin from the positive half of the axis of rotation. For example, rotating $\vec{e}_1$ about the x_3-axis by θ radians results in the vector

$$T(\vec{e}_1) = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix}$$

Transforming the first standard vector $\vec{e}_1$ yields the first column of A. Likewise the remaining columns can be found by transforming the other standard vectors.

$$T(\vec{e}_2) = \begin{pmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{pmatrix}, \quad T(\vec{e}_3) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

The third standard vector does not change under this transformation because it is parallel to the rotation axis. The standard matrix for a rotation about the x_3-axis is

$$A = \begin{pmatrix} T(\vec{e}_1) & T(\vec{e}_2) & T(\vec{e}_3) \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

A similar analysis gives us the standard matrices for rotations about the x_1 and the x_2 axes. Results are summarized in Table 1.1. The standard matrices in the table can be multiplied together to model transforms that perform multiple transformations. The next example demonstrates this application.

rotation axis	**standard matrix**
x_1-axis	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$
x_2-axis	$\begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$
x_3-axis	$\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Table 1.1: Standard matrices for 3D rotations about the coordinate axes.

Example 1: 3D Rotations

Suppose that the transform $\vec{x} \to A\vec{x}$ first rotates points in $\mathbb{R}^3$ about the x_2-axis by $\pi/2$ radians and then rotates points about the x_1-axis by π radians. We can determine the standard matrix, A, for this transform in a few different ways. One approach is to use the standard matrices in Table 1.1. The standard matrix, A, is the product of two rotation matrices.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\pi & -\sin\pi \\ 0 & \sin\pi & \cos\pi \end{pmatrix} \begin{pmatrix} \cos(\pi/2) & 0 & -\sin(\pi/2) \\ 0 & 1 & 0 \\ \sin(\pi/2) & 0 & \cos(\pi/2) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Note that the rotation about the x_2-axis is applied before the rotation about the x_1-axis, which determines the multiplication order. The standard matrix for the first transformation is placed in the rightmost position.

We could also obtain the same result by transforming the standard vectors, because $A = \begin{pmatrix} T(\vec{e}_1) & T(\vec{e}_2) & T(\vec{e}_3) \end{pmatrix}$. The first standard vector gives us the first column of A.

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \to \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

This result agrees with our result obtained above by multiplying rotation matrices together. Note also that our convention is that a positive rotation is in the counterclockwise direction when looking toward the origin from the positive half of the axis of rotation.

The Data Matrix for 3D Transforms

Similar to the 2D case, for problems involving many points it is convenient to represent the points a data matrix. Analogous to our approach in 2D, points in $\mathbb{R}^3$ can be represented in a matrix whose columns are vectors that correspond to the points we wish to transform. We may transform this matrix with a matrix-vector multiplication. Recall that the product of two matrices A and D, is defined as

$$AD = A\begin{pmatrix} \vec{d}_1 & \vec{d}_2 & \cdots & \vec{d}_p \end{pmatrix} = \begin{pmatrix} A\vec{d}_1 & A\vec{d}_2 & \cdots & A\vec{d}_p \end{pmatrix}$$

where $\vec{d}_1, \vec{d}_2, \cdots, \vec{d}_p$ are the columns of D. In other words, can perform the transformation on our data by computing AD, which transforms each column independently of the others. The following example demonstrates this approach.

Example 2: A Projection in 3D with the Data Matrix

x_1	x_2	x_3
1	1	1
1	2	1
2	2	1
2	1	1
1	1	2
1	2	2
2	2	2
2	1	2

Table 1.2: Corners of a cube with side length 1.

Data in Table (1.2) define a cube in $\mathbb{R}^3$ with side length 1. Suppose the linear transform $T(\vec{x})$ projects points in $\mathbb{R}^3$ onto the x_1x_2-plane. In this example we will construct the matrix, A, that is the standard matrix of the transformation $T(\vec{x}) = A\vec{x}$.

The data in Table (1.2) (blue) and its projection (green) are shown Figure (1.2).

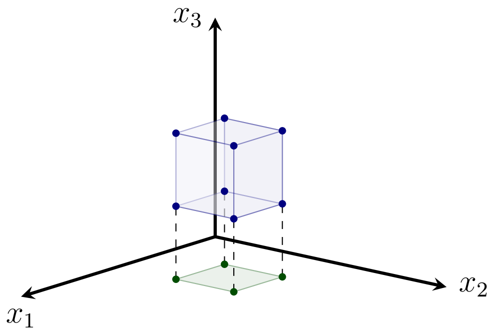


Figure 1.1: Data from Table (1.2) and its projection onto the x_1x_2-plane.

Because the given transform that we are dealing with in this example is linear, we can express the transform in the form of a matrix-vector product

$$T(\vec{x}) = A\vec{x}$$

where A is a 3×3 matrix. Moreover, because we are working with a linear transform, each column of A is equal to the product

$$A\vec{e}_i, \quad i = 1, 2, 3$$

and $\vec{e}_i$ is a standard vector. For example, the first column of A can be found using $\vec{e}_1$, which is the vector

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Projecting $\vec{e}_1$ onto the x_1x_2-plane does not change the vector, because the vector is already in that plane.

$$\vec{e}_1 \to A\vec{e}_1 = \vec{e}_1 = \text{first column of } A$$

The first column of A is $\vec{e}_1$. Likewise, the second column of A is $\vec{e}_2$, becuase $\vec{e}_2$ is also already in the x_1x_2-plane.

$$\vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \to A\vec{e}_2 = \vec{e}_2 = \text{second column of } A$$

The last column of A is the projection of $\vec{e}_3$ onto the plane, which is the zero vector.

$$\vec{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow A\vec{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \text{third column of } A$$

Combining our results for each column of A gives us the standard matrix.

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

Now that we have the standard matrix for this transform, we can use it to transform the data in Table 1. Representing each point as a vector in $\mathbb{R}^3$ and placing the vectors in a data matrix, D, will allow us to compute the projection using a matrix multiplication. Our matrix D is

$$D = \begin{pmatrix} 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{pmatrix}$$

The transformed points can be computed as follows.

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

Extracting the columns of the product gives us the projected points.

3D Homogeneous Coordinates

Homogeneous coordinates in 3D are analogous to the homogeneous 2D coordinates we introduced in the previous section.

Homogeneous Coordinates in $\mathbb{R}^3$

$(X, Y, Z, 1)$ are homogeneous coordinates for (x, y, z) in $\mathbb{R}^3$

A translation of the form $(x, y, z) \to (x + h, y + k, z + l)$ can be represented as a matrix multiplication with homogeneous coordinates:

$$\begin{pmatrix} 1 & 0 & 0 & h \\ 0 & 1 & 0 & k \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + h \\ y + k \\ z + l \\ 1 \end{pmatrix}$$

Example 3: A Translation in 3D

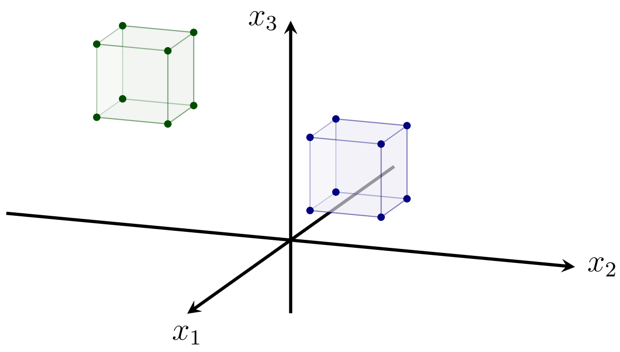
The data in Table (1.2) can be translated using a homogeneous coordinate system. The data matrix D_n in homogeneous coordinates would be

$$D_h = \begin{pmatrix} 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The transform that, for example, shifts the data by -3 units in the x_2 direction and by 1 unit in the x_3-direction is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} D_h = \begin{pmatrix} 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 \\ -2 & -1 & -1 & -2 & -2 & -1 & -1 & -2 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The figure below shows the original data (blue) and its translated version (green).



Exercises

1. Construct the standard matrices for the following transforms.

   a) The 4×4 standard matrix of the transform $\vec{x} \to A\vec{x}$ that uses homogeneous coordinates to reflect points in $\mathbb{R}^3$ across the plane $x_3 = k$, where k is any real number.

   b) The 3×3 standard matrix of the transform $\vec{x} \to A\vec{x}$ that reflects points in $\mathbb{R}^3$ across the plane $x_1 + x_2 = 0$.

   c) The 3×3 standard matrix of the transform $\vec{x} \to A\vec{x}$ that first rotates points in $\mathbb{R}^3$ about the x_3-axis by an angle θ and then projects them onto the x_2x_3-plane.

2. Line L passes through the point $(1, 0, 0)$ and is parallel to the vector $\vec{v}$, where

$$\vec{v} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

   Construct the 4×4 matrix that uses homogeneous coordinates to rotate points in $\mathbb{R}^3$ about line L by an angle θ.