

5.2 Matrix Equations

Linear equations. An $m \times n$ system of linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m, \end{aligned}$$

can be written as a matrix equation $A\vec{X} = \vec{b}$. Let A be the matrix of coefficients a_{ij} , let \vec{X} be the column vector of variable names x_1, \dots, x_n and let \vec{b} be the column vector with components b_1, \dots, b_n . Then

$$\begin{aligned} A\vec{X} &= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \vdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\ &= \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{pmatrix} \\ &= \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \end{aligned}$$

Therefore, $A\vec{X} = \vec{b}$. Conversely, a matrix equation $A\vec{X} = \vec{b}$ corresponds to a set of linear algebraic equations, because of reversible steps above.

A system of linear equations can be represented by its **variable list** x_1, x_2, \dots, x_n and its **augmented matrix**

$$(1) \quad \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ & & \vdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_n \end{pmatrix}.$$

The augmented matrix is denoted $\mathbf{aug}(A, \vec{b})$ when the system is given as $A\vec{X} = \vec{b}$. Given an augmented matrix and a variable list, the conversion back to a linear system of equations is made by expanding $\mathbf{aug}(A, \vec{b})\vec{Y} = \mathbf{0}$, where \vec{Y} has components $x_1, \dots, x_n, -1$. Because this expansion involves just dot products, it is possible to rapidly write out the linear system for a given augmented matrix. Hand work often contains the

following kind of exposition:

$$(2) \quad \left(\begin{array}{cccc|c} x_1 & x_2 & \cdots & x_n & \\ a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ & & \vdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_n \end{array} \right)$$

In (2), a dot product is applied to the first n elements of each row, using the variable list written above the columns. The symbolic answer is set equal to the rightmost column's entry, in order to recover the equations.

It is usual in homogeneous systems to *not write* the column of zeros, but to deal directly with A instead of $\mathbf{aug}(A, \mathbf{0})$. This convention is justified by arguing that the rightmost column of zeros is unchanged by swap, multiply and combination rules, which are defined the next paragraph.

Elementary row operations. The three operations on equations which produce equivalent systems can be translated directly to row operations on the augmented matrix for the system. The rules produce **equivalent systems**, that is, the three rules neither create nor destroy solutions.

- Swap** Two rows can be interchanged.
- Multiply** A row can be multiplied by multiplier $m \neq 0$.
- Combination** A multiple of one row can be added to a different row.

Documentation of row operations. Throughout the display below, symbol **s** stands for **source**, symbol **t** for **target**, symbol **m** for **multiplier** and symbol **c** for **constant**.

- Swap** **swap(s,t)** \equiv swap rows **s** and **t**.
- Multiply** **mult(t,m)** \equiv multiply row **t** by **m** $\neq 0$.
- Combination** **combo(s,t,c)** \equiv add **c** times row **s** to row **t** \neq **s**.

The standard for documentation is to write the notation next to the target row, which is the row to be changed. For swap operations, the notation is written next to the first row that was swapped, and optionally next to both rows. The notation was developed from early `maple` notation for the corresponding operations `swaprow`, `mulrow` and `addrow`, appearing in the `maple` package `linalg`. In early versions of `maple`, an additional argument A is used to reference the matrix for which the elementary row operation is to be applied. For instance, `addrow(A,1,3,-5)`

assumes matrix A is the object of the combination rule, which is documented in written work as `combo(1,3,-5)`. In written work, symbol A is omitted, because A is the matrix appearing on the previous line of the sequence of steps.

Later versions of `maple` use the `LinearAlgebra` package, and a single function `RowOperation` to accomplish the same thing. A short conversion table appears below.

Hand-written	<code>maple linalg</code>	<code>maple LinearAlgebra</code>
<code>swap(s,t)</code>	<code>swaprow(A,s,t)</code>	<code>RowOperation(A,[t,s])</code>
<code>mult(t,c)</code>	<code>mulrow(A,t,c)</code>	<code>RowOperation(A,t,c)</code>
<code>combo(s,t,c)</code>	<code>addrow(A,s,t,c)</code>	<code>RowOperation(A,[t,s],c)</code>

Conversion between packages can be controlled by the following function definitions, which causes the `maple` code to be the same regardless of which linear algebra package is used.

Maple `linalg`

```
combo:=(a,s,t,c)->addrow(a,s,t,c);
swap:=(a,s,t)->swaprow(a,s,t);
mult:=(a,t,m)->mulrow(a,t,m);
```

Maple `LinearAlgebra`

```
combo:=(a,s,t,c)->RowOperation(a,[t,s],c);
swap:=(a,s,t)->RowOperation(a,[t,s]);
mult:=(a,t,m)->RowOperation(a,t,m);
macro(matrix=Matrix);
```

RREF. The reduced row-echelon form of a matrix, or **rref**, is specified by the following requirements.

1. Zero rows appear last. Each nonzero row has first element 1, called a **leading one**. The column in which the leading one appears, called a **pivot column**, has all other entries zero.
2. The pivot columns appear as consecutive initial columns of the identity matrix I . Trailing columns of I might be absent.

The matrix (3) below is a typical **rref** which satisfies the preceding properties. Displayed secondly is the reduced echelon system (4) in the variables x_1, \dots, x_8 represented by the augmented matrix (3).

$$(3) \quad \begin{pmatrix} 1 & 2 & 0 & 3 & 4 & 0 & 5 & 0 & 6 \\ 0 & 0 & 1 & 7 & 8 & 0 & 9 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 1 & 11 & 0 & 12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$(4) \quad \begin{array}{rcccccc} x_1 & +2x_2 & & +3x_4 & +4x_5 & & +5x_7 & & = 6 \\ & & x_3 & +7x_4 & +8x_5 & & +9x_7 & & = 10 \\ & & & & & & x_6 & +11x_7 & = 12 \\ & & & & & & & & x_8 = 13 \end{array}$$

Observe that (3) is an **rref** and (4) is a reduced echelon system. The initial 4 columns of the 7×7 identity matrix I appear in natural order in matrix (3); the trailing 3 columns of I are absent.

If the **rref** of the augmented matrix has a leading one in the last column, then the corresponding system of equations then has an equation “ $0 = 1$ ” displayed, which signals an **inconsistent** system. It must be emphasized that an **rref** always exists, even if the corresponding equations are inconsistent.

Elimination method. The elimination algorithm for equations (see page 178) has an implementation for matrices that parallels the algorithm for equations, except that the algorithm *never terminates* upon inconsistency. Computer algebra systems and computer numerical laboratories use similar algorithms which produce exactly the same result. A row is called **processed** if it has been so marked, in which case either (1) the row is all zeros, or else (2) the row contains a leading one and all other entries in that column are zero. A row is **unprocessed** if it has not been marked as processed.

1. Move each unprocessed row of zeros to the last row using **swap** and mark it *processed*.
2. Identify an unprocessed nonzero row having the least leading zeros. Apply the **multiply** rule to insure a leading one. Apply the **combination** rule to make all other entries zero in that column. The number of leading ones (lead variables) has been increased by one and the current column is a column of the identity matrix.
3. All other unprocessed rows have leading zeros up to and including the leading one column. Apply the **swap** rule to move the row into the lowest position, so that all unprocessed rows and zero rows follow, and mark the row as *processed*, e.g., replace the leading one by $\boxed{1}$.

4. Repeat steps 1–3, until all rows have been processed. Then all leading ones have been defined and the resulting matrix is in reduced row-echelon form.

The reduced row-echelon form of a matrix A is denoted $\mathbf{rref}(A)$. The algorithm to obtain it is called **Gauss-Jordan elimination**. Two examples:

- $\mathbf{rref}(\mathbf{0}) = \mathbf{0}$ In **step 2**, all rows of the zero matrix $\mathbf{0}$ are zero. No changes are made to the zero matrix.
- $\mathbf{rref}(I) = I$ In **step 2**, each row has a leading one. No changes are made to the identity matrix I .

Frame Sequence. A sequence of swap, multiply and combination steps applied to a system of equations is called a **frame sequence**. The viewpoint is that a camera is pointed over the shoulder of an assistant who writes the mathematics, and after the completion of each step, a photo is taken. The sequence of photo frames is the frame sequence. The **First Frame** is the original system and the **Last Frame** is the reduced row echelon system.

The same terminology applies for systems $A\vec{x} = \vec{b}$ represented by an augmented matrix $C = \mathbf{aug}(A, \vec{b})$. The First Frame is C and the Last Frame is $\mathbf{rref}(C)$.

Steps in a frame sequence can be documented by the notation

$$\mathbf{swap}(s, t), \mathbf{mult}(t, m), \mathbf{combo}(s, t, c),$$

each written next to the target row t . During the sequence, consecutive initial columns of the identity, called **pivot columns**, are created as steps toward the \mathbf{rref} . Trailing columns of the identity might not appear. An illustration:

$$\begin{array}{l} \text{Frame 1: } \left(\begin{array}{cccc|c} 1 & 2 & -1 & 0 & 1 \\ 1 & 4 & -1 & 0 & 2 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \text{Original augmented matrix.} \\ \\ \text{Frame 2: } \left(\begin{array}{cccc|c} 1 & 2 & -1 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \begin{array}{l} \mathbf{combo}(1, 2, -1) \\ \text{Pivot column 1 completed.} \end{array} \\ \\ \text{Frame 3: } \left(\begin{array}{cccc|c} 1 & 2 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \mathbf{swap}(2, 3) \end{array}$$

$$\begin{array}{l}
\text{Frame 4: } \left(\begin{array}{cccc|c} 1 & 2 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & -2 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \text{combo}(2,3,-2) \\
\text{Frame 5: } \left(\begin{array}{cccc|c} 1 & 0 & -3 & 0 & -1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & -2 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \begin{array}{l} \text{combo}(2,1,-2) \\ \text{Pivot column 2 completed.} \end{array} \\
\text{Frame 6: } \left(\begin{array}{cccc|c} 1 & 0 & -3 & 0 & -1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \text{mult}(3,-1/2) \\
\text{Frame 7: } \left(\begin{array}{cccc|c} 1 & 0 & -3 & 0 & -1 \\ 0 & 1 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \text{combo}(3,2,-1) \\
\text{Last Frame: } \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \begin{array}{l} \text{combo}(3,1,3) \\ \text{Pivot column 3 completed.} \\ \text{rref found.} \end{array}
\end{array}$$

Back-substitution and efficiency. The algorithm implemented in the preceding frame sequence is *easy to learn*, because the actual work is organized by creating pivot columns, via swap, combination and multiply. The pivot columns are initial columns of the identity. You are advised to learn the algorithm in this form, but please *change the algorithm* as you become more efficient at doing the steps. See the examples for illustrations.

Back substitution. Computer implementations and also hand computation can be made more efficient by changing step 2 into **step 2a** and adding **step 5**. Step 2a introduces zeros *below* the leading one. Step 5 introduces zeros *above* each leading one, working from the lowest nonzero row back to the first row. Literature refers to **step 5** as **back-substitution**. The process of back-substitution is exactly the original elimination algorithm applied to the variable list in reverse order.

Avoiding fractions. A matrix A containing only integer entries can often be put into reduced row-echelon form without introducing fractions. The **multiply** rule introduces fractions, so its use should be limited. It is advised that leading ones be introduced only when convenient, otherwise make the leading coefficient nonzero and positive. Divisions at the end of the computation will produce the **rref**.

Clever use of the **combination** rule can sometimes create a leading one

without introducing fractions. Consider the two rows

$$\begin{array}{ccccc} 25 & 0 & 1 & 0 & 5 \\ 7 & 0 & 2 & 0 & 2 \end{array}$$

The second row multiplied by -4 and added to the first row effectively replaces the 25 by -3 , whereupon adding the first row twice to the second gives a leading one in the second row. The resulting rows are fraction-free.

$$\begin{array}{ccccc} -3 & 0 & -7 & 0 & -3 \\ 1 & 0 & -12 & 0 & -4 \end{array}$$

Rank and Nullity. What does it mean, if the first column of a **rref** is the zero vector? It means that the corresponding variable x_1 is a **free variable**. In fact, every column that does not contain a leading one corresponds to a free variable in the standard general solution of the system of equations. Symmetrically, each leading one identifies a pivot column and corresponds to a **leading variable**.

The number of leading ones is the **rank** of the matrix, denoted $\mathbf{rank}(A)$. The rank cannot exceed the row dimension nor the column dimension. The column count less the number of leading ones is the **nullity** of the matrix, denoted $\mathbf{nullity}(A)$. It equals the number of free variables.

Regardless of how matrix B arises, augmented or not, we have the relation

$$\text{variable count} = \mathbf{rank}(B) + \mathbf{nullity}(B).$$

If $B = \mathbf{aug}(A, \vec{b})$ for $A\vec{X} = \vec{b}$, then the variable count n comes from \vec{X} and the column count of B is *one more*, or $n + 1$. Replacing the *variable count* by the *column count* can therefore lead to fundamental errors.

Inverses. An efficient method to find the inverse B of a square matrix A , should it happen to exist, is to form the augmented matrix $C = \mathbf{aug}(A, I)$ and then read off B as the package of the last n columns of $\mathbf{rref}(C)$. This method is based upon the equivalence

$$\mathbf{rref}(\mathbf{aug}(A, I)) = \mathbf{aug}(I, B) \quad \text{if and only if} \quad AB = I.$$

The next theorem aids not only in establishing this equivalence but also in the practical matter of testing a candidate solution for the inverse matrix. The proof is delayed to page 277.

Theorem 6 (Inverse Test)

If A and B are square matrices such that $AB = I$, then also $BA = I$. Therefore, only one of the equalities $AB = I$ or $BA = I$ is required to check an inverse.

Theorem 7 (The rref Inversion Method)

Let A and B denote square matrices. Then

- (a) If $\mathbf{rref}(\mathbf{aug}(A, I)) = \mathbf{aug}(I, B)$, then $AB = BA = I$ and B is the inverse of A .
- (b) If $AB = BA = I$, then $\mathbf{rref}(\mathbf{aug}(A, I)) = \mathbf{aug}(I, B)$.
- (c) If $\mathbf{rref}(\mathbf{aug}(A, I)) = \mathbf{aug}(C, B)$ and $C \neq I$, then A is not invertible.

The proof is delayed to page 277.

Finding inverses. The method will be illustrated for the matrix

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

Define the first frame of the sequence to be $C_1 = \mathbf{aug}(C, I)$, then compute the frame sequence to $\mathbf{rref}(C)$ as follows.

$$\begin{array}{ll} C_1 = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right) & \text{First Frame} \\ \\ C_2 = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & -1 & 1 \end{array} \right) & \text{combo}(3, 2, -1) \\ \\ C_3 = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1/2 & 1/2 \end{array} \right) & \text{mult}(3, 1/2) \\ \\ C_4 = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1 & 0 & -1/2 & 1/2 \end{array} \right) & \text{combo}(3, 2, 1) \\ \\ C_5 = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1/2 & -1/2 \\ 0 & 1 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1 & 0 & -1/2 & 1/2 \end{array} \right) & \begin{array}{l} \text{combo}(3, 1, -1) \\ \text{Last Frame} \end{array} \end{array}$$

The theory implies that the inverse of A is the matrix in the right half of the last frame:

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

Elementary Matrices. An elementary matrix E is the result of applying a combination, multiply or swap rule to the identity matrix. It follows from the definition of matrix multiplication that EA is the result of the corresponding combination, multiply or swap rule applied to matrix A . The computer algebra system `maple` easily writes out typical 4×4 elementary matrices (C=Combination, M=Multiply, S=Swap) as follows.

<code>with(linalg):</code>	<code>with(LinearAlgebra):</code>
<code>Id:=diag(1,1,1,1);</code>	<code>Id:=IdentityMatrix(4);</code>
<code>C:=addrow(Id,2,3,c);</code>	<code>C:=RowOperation(Id,[3,2],c);</code>
<code>M:=mulrow(Id,3,m);</code>	<code>M:=RowOperation(Id,3,m);</code>
<code>S:=swaprow(Id,1,4);</code>	<code>S:=RowOperation(Id,[4,1]);</code>

The reader is encouraged to write out the above elementary matrices by hand or machine, obtaining

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & m & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Constructing Elementary Matrices E .

- Multiply** Change a one in the identity matrix to symbol $m \neq 0$.
- Combination** Change a zero in the identity matrix to symbol c .
- Swap** Interchange two rows of the identity matrix.

Constructing E^{-1} from elementary matrix E .

- Multiply** Change diagonal multiplier $m \neq 0$ in E to $1/m$.
- Combination** Change multiplier c in E to $-c$.
- Swap** The inverse of E is E itself.

Theorem 8 (The rref and elementary matrices)

Let A be a given matrix of row dimension n . Then there exist $n \times n$ elementary matrices E_1, E_2, \dots, E_k such that

$$\mathbf{rref}(A) = E_k \cdots E_2 E_1 A.$$

The result is the observation that left multiplication of matrix A by elementary matrix E gives the answer EA for the corresponding multiply, combination or swap operation. The matrices E_1, E_2, \dots represent the multiply, combination and swap operations performed in the frame sequence which take the First Frame into the Last Frame, or equivalently, original matrix A into $\mathbf{rref}(A)$.

Illustration. Consider the following 6-frame sequence.

$$\begin{aligned}
 A_1 &= \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 0 \\ 3 & 6 & 3 \end{pmatrix} && \text{Frame 1, original matrix.} \\
 A_2 &= \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & -6 \\ 3 & 6 & 3 \end{pmatrix} && \text{Frame 2, combo(1,2,-2).} \\
 A_3 &= \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 3 & 6 & 3 \end{pmatrix} && \text{Frame 3, mult(2,-1/6).} \\
 A_4 &= \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 0 & 0 & -6 \end{pmatrix} && \text{Frame 4, combo(1,3,-3).} \\
 A_5 &= \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} && \text{Frame 5, combo(2,3,-6).} \\
 A_6 &= \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} && \text{Frame 6, combo(2,1,-3). Found rref.}
 \end{aligned}$$

The corresponding 3×3 elementary matrices are

$$\begin{aligned}
 E_1 &= \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} && \text{Frame 2, combo(1,2,-2) applied to } I. \\
 E_2 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1/6 & 0 \\ 0 & 0 & 1 \end{pmatrix} && \text{Frame 3, mult(2,-1/6) applied to } I. \\
 E_3 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} && \text{Frame 4, combo(1,3,-3) applied to } I. \\
 E_4 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -6 & 1 \end{pmatrix} && \text{Frame 5, combo(2,3,-6) applied to } I. \\
 E_5 &= \begin{pmatrix} 1 & -3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} && \text{Frame 6, combo(2,1,-3) applied to } I.
 \end{aligned}$$

The frame sequence can be written as follows.

$$A_2 = E_1 A_1 \quad \text{Frame 2, } E_1 \text{ equals combo(1,2,-2) on } I.$$

$A_3 = E_2 A_2$	Frame 3, E_2 equals mult(2,-1/6) on I .
$A_4 = E_3 A_3$	Frame 4, E_3 equals combo(1,3,-3) on I .
$A_5 = E_4 A_4$	Frame 5, E_4 equals combo(2,3,-6) on I .
$A_6 = E_5 A_5$	Frame 6, E_5 equals combo(2,1,-3) on I .
$A_6 = E_5 E_4 E_3 E_2 E_1 A_1$	Summary, frames 1-6. This relation is $\mathbf{rref}(A_1) = E_5 E_4 E_3 E_2 E_1 A_1$, which is the result of the Theorem.

The summary is the equation

$$\mathbf{rref}(A_1) = \begin{pmatrix} 1 & -3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -6 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{6} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} A_1$$

The inverse relationship $A_1 = E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1} E_5^{-1} \mathbf{rref}(A_1)$ is formed by the rules for constructing E^{-1} from elementary matrix E , page 274, the result being

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 6 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{rref}(A_1)$$

1 Example (Reduced Row–Echelon Form) Find the reduced row–echelon form of the coefficient matrix A using the **rref method**, page 269. Then solve the system.

$$\begin{aligned} x_1 + 2x_2 - x_3 + x_4 &= 0, \\ x_1 + 3x_2 - x_3 + 2x_4 &= 0, \\ x_2 + x_4 &= 0. \end{aligned}$$

Solution: The coefficient matrix A and its **rref** are given by (details below)

$$A = \begin{pmatrix} 1 & 2 & -1 & 1 \\ 1 & 3 & -1 & 2 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad \mathbf{rref}(A) = \begin{pmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Using variable list x_1, x_2, x_2, x_4 , the equivalent reduced echelon system is

$$\begin{aligned} x_1 - x_3 - x_4 &= 0, \\ x_2 + x_4 &= 0, \end{aligned}$$

which has standard general solution

$$\begin{aligned} x_1 &= t_1 + t_2, \\ x_2 &= -t_2, \\ x_3 &= t_1, \\ x_4 &= t_2, \quad -\infty < t_1, t_2 < \infty. \end{aligned}$$

Details of the rref method.

$$\begin{pmatrix} 1^* & 2 & -1 & 1 \\ 1 & 3 & -1 & 2 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

The coefficient matrix A . Leading one identified, marked as 1^* .

$$\begin{pmatrix} \boxed{1} & 2 & -1 & 1 \\ 0 & 1^* & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Apply the **combination** rule to zero the other entries in column 1. Mark the row processed. Identify the next leading one.

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

Apply the **combination** rule to zero the other entries in column 2. Mark the row processed; **rref** found.

Proof of Theorem 6:

Assume $AB = I$. Let $C = BA - I$. We intend to show $C = \mathbf{0}$, then $BA = I$, which completes the proof.

Compute $AC = ABA - A = AI - A = \mathbf{0}$. It follows that the columns \vec{y} of C are solutions of the homogeneous equation $A\vec{y} = \vec{\mathbf{0}}$. The proof is completed by showing that the only solution of $A\vec{y} = \vec{\mathbf{0}}$ is $\vec{y} = \vec{\mathbf{0}}$, because then C has all zero columns, or equivalently, $C = \mathbf{0}$.

First, $B\vec{u} = \vec{\mathbf{0}}$ implies $\vec{u} = AB\vec{u} = A\vec{\mathbf{0}} = \vec{\mathbf{0}}$, hence B has an inverse, and then $B\vec{x} = \vec{y}$ has a unique solution $\vec{x} = B^{-1}\vec{y}$.

Suppose $A\vec{y} = \vec{\mathbf{0}}$. Write $\vec{y} = B\vec{x}$. Then $\vec{x} = I\vec{x} = AB\vec{x} = A\vec{y} = \vec{\mathbf{0}}$. This implies $\vec{y} = B\vec{x} = B\vec{\mathbf{0}} = \vec{\mathbf{0}}$. The proof is complete.

Proof of Theorem 7:

Details for (a). Let $C = \mathbf{aug}(A, I)$ and assume $\mathbf{rref}(C) = \mathbf{aug}(I, B)$. Solving the $n \times 2n$ system $C\vec{X} = \vec{\mathbf{0}}$ is equivalent to solving the system $A\vec{Y} + I\vec{Z} = \vec{\mathbf{0}}$ with n -vector unknowns \vec{Y} and \vec{Z} . This system has exactly the same solutions as $I\vec{Y} + B\vec{Z} = \vec{\mathbf{0}}$, by the equation $\mathbf{rref}(C) = \mathbf{aug}(I, B)$. The latter is a reduced echelon system with lead variables equal to the components of \vec{Y} and free variables equal to the components of \vec{Z} . Multiplying by A gives $A\vec{Y} + AB\vec{Z} = \vec{\mathbf{0}}$, hence $-\vec{Z} + AB\vec{Z} = \vec{\mathbf{0}}$, or equivalently $AB\vec{Z} = \vec{Z}$ for every vector \vec{Z} (because its components are free variables). Letting \vec{Z} be a column of I shows that $AB = I$. Then $AB = BA = I$ by Theorem 6, and B is the inverse of A .

Details for (b). Assume $AB = I$. We prove the identity $\mathbf{rref}(\mathbf{aug}(A, I)) = \mathbf{aug}(I, B)$. Let the system $A\vec{Y} + I\vec{Z} = \vec{\mathbf{0}}$ have a solution \vec{Y}, \vec{Z} . Multiply by B to obtain $BA\vec{Y} + B\vec{Z} = \vec{\mathbf{0}}$. Use $BA = I$ to give $\vec{Y} + B\vec{Z} = \vec{\mathbf{0}}$. The latter system therefore has \vec{Y}, \vec{Z} as a solution. Conversely, a solution \vec{Y}, \vec{Z} of $\vec{Y} + B\vec{Z} = \vec{\mathbf{0}}$ is a solution of the system $A\vec{Y} + I\vec{Z} = \vec{\mathbf{0}}$, because of multiplication by A . Therefore, $A\vec{Y} + I\vec{Z} = \vec{\mathbf{0}}$ and $\vec{Y} + B\vec{Z} = \vec{\mathbf{0}}$ are *equivalent* systems. The latter is in reduced row-echelon form, and therefore $\mathbf{rref}(\mathbf{aug}(A, I)) = \mathbf{aug}(I, B)$.

Details for (c). Suppose not, then A is invertible. Then (b) implies $\mathbf{aug}(C, B) = \mathbf{rref}(\mathbf{aug}(A, I)) = \mathbf{aug}(I, B)$, which gives $C = I$, a contradiction.

Exercises 5.2

Lead and free variables. For each matrix A , assume a homogeneous system $A\vec{X} = \vec{\mathbf{0}}$ with variable list x_1, \dots, x_n . List the lead and free variables.

1. $\begin{pmatrix} 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
2. $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 2 \end{pmatrix}$
3. $\begin{pmatrix} 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
4. $\begin{pmatrix} 1 & 2 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
5. $\begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
6. $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$
7. $\begin{pmatrix} 1 & 1 & 3 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
8. $\begin{pmatrix} 1 & 2 & 0 & 3 & 4 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
9. $\begin{pmatrix} 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
10. $\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

$$11. \begin{pmatrix} 0 & 1 & 0 & 5 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$12. \begin{pmatrix} 1 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Rank and nullity. For each matrix A , assume a homogeneous system $A\vec{X} = \vec{0}$ with variable list x_1, \dots, x_n . List the lead variables and the free variables. Then report the values of **rank**(A) and **nullity**(A).

$$13. \begin{pmatrix} 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Elementary Matrices. Define symbols **combo**, **swap** and **mult** as follows:

(a) The symbol **combo**(i, j, c) documents the addition of c times source row i to target row j of matrix A .

(b) The symbol **swap**(i, j) documents the interchange of row i and row j of matrix A .

(c) The symbol **mult**(i, m) documents the multiplication of row i of matrix A by constant $m \neq 0$.