2.2 Matrices in Matlab

You can think of a matrix as being made up of 1 or more row vectors of equal length. Equivalently, you can think of a matrix of being made up of 1 or more column vectors of equal length. Consider, for example, the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 5 & -1 & 0 & 0 \\ 3 & -2 & 5 & 0 \end{bmatrix}.$$

One could say that the matrix A is made up of 3 rows of length 4. Equivalently, one could say that matrix A is made up of 4 columns of length 3. In either model, we have 3 rows and 4 columns. We will say that the dimensions of the matrix are 3-by-4, sometimes written 3×4.

We already know how to enter a matrix in Matlab: delimit each item in a row with a space or comma, and start a new row by ending a row with a semicolon.

```
>> A=[1 2 3 0;5 -1 0 0;3 -2 5 0]
A =
     1     2     3     0
     5    -1     0     0
     3    -2     5     0
```

We can use Matlab's **size** command to determine the dimensions of any matrix.

```
>> size(A)
ans =
     3     4
```

That's 3 rows and 4 columns!

Indexing

Indexing matrices in Matlab is similar to the indexing we saw with vectors. The difference is that there is another dimension[2].

To access the element in row 2 column 3 of matrix A, enter this command.

[2] We'll see later that we can have more than two dimensions.

```
>> A(2,3)
ans =
     0
```

This is indeed the element in row 2, column 3 of matrix A.

You can access an entire row with Matlab's colon operator. The command **A(2,:)** essentially means "row 2 every column" of matrix A.

```
>> A(2,:)
ans =
     5    -1     0     0
```

Note that this is the second row of matrix A.

Similarly, you can access any column of matrix A. The notation **A(:,2)** is pronounced "every row column 2" of matrix A.

```
>> A(:,2)
ans =
     2
    -1
    -2
```

Note that this is the second column of matrix A.

You can also extract a submatrix from the matrix A with indexing. Suppose, for example, that you would like to extract a submatrix using rows 1 and 3 and columns 2 and 4.

```
>> A([1,3],[2,4])
ans =
     2     0
    -2     0
```

Study this carefully and determine if we've truly selected rows 1 and 3 and columns 2 and 4 of matrix A. It might help to repeat the contents of matrix A.

```
>> A
A =
     1     2     3     0
     5    -1     0     0
     3    -2     5     0
```

You can assign a new value to an entry of matrix A.

```
>> A(3,4)=12
A =
     1     2     3     0
     5    -1     0     0
     3    -2     5    12
```

When you assign to a row, column, or submatrix of matrix A, you must replace the contents with a row, column, or submatrix of equal dimension. For example, this next command will assign new contents to the first row of matrix A.

```
>> A(1,:)=20:23
A =
    20    21    22    23
     5    -1     0     0
     3    -2     5    12
```

There is an exception to this rule. If the right side contains a single number, then that number will be assigned to every entry of the submatrix on the left. For example, to make every entry in column 2 of matrix A equal to 11, try the following code.

```
>> A(:,2)=11
A =
    20    11    22    23
     5    11     0     0
     3    11     5    12
```

It's interesting what happens (and very powerful) when you try to assign a value to an entry that has a row or column index larger than the corresponding dimension of the matrix. For example, try this command.

```
>> A(5,5)=777
A =
    20    11    22    23     0
     5    11     0     0     0
     3    11     5    12     0
     0     0     0     0     0
     0     0     0     0   777
```

Note that Matlab happily assigns 777 to row 5, column 5, expanding the dimensions of the matrix and padding the missing entries with zeros.

```
>> size(A)
ans =
     5     5
```

The Transpose of a Matrix

You can take the transpose of a matrix in exactly the same way that you took the transpose of a row or column vector. For example, form a "magic" matrix with the following command.

```
>> A=magic(4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

You can compute A^T with the following command.

```
>> A.'
ans =
    16     5     9     4
     2    11     7    14
     3    10     6    15
    13     8    12     1
```

Note that the first row of matrix A^T was previously the first column of matrix A. The second row of matrix A^T was previously the second column of matrix A, and so on for the third and fourth columns of matrix A^T. In essence, taking the transpose reflects the matrix A across its main diagonal (upper left corner to lower right corner), so the rows of A become columns of A^T and the columns of A become rows of A^T.

Building Matrices

Matlab has some powerful capabilities for building new matrices out of one or more matrices and/or vectors. For example, start by building a 2×3 matrix of ones.

```
>> A=ones(2,3)
A =
     1     1     1
     1     1     1
```

Now, build a new matrix with A as the first column and A as the second column. As we are not starting a new row, we can use either space or commas to delimit the row entries.

```
>> C=[A A]
C =
     1     1     1     1     1     1
     1     1     1     1     1     1
```

On the other hand, suppose that we want to build a new matrix with A as the first row and A as the second row. To start a new row we must end the first row with a semicolon.

```
>> C=[A; A]
C =
     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

Let's create a 2×3 matrix of all zeros.

```
>> D=zeros(2,3)
D =
     0     0     0
     0     0     0
```

Now, let's build a matrix out of the matrices A and D.

```
>> E=[A D;D A]
E =
     1     1     1     0     0     0
     1     1     1     0     0     0
     0     0     0     1     1     1
     0     0     0     1     1     1
```

The possibilities are endless, with one caveat. The dimensions must be correct or Matlab will report an error. For example, create a 2×2 matrix of ones.

```
>> A=ones(2,2)
A =
     1     1
     1     1
```

And a 2×3 matrix of zeros.

```
>> B=zeros(2,3)
B =
     0     0     0
     0     0     0
```

It's possible to build a new matrix with A and B as row elements.

```
>> C=[A B]
C =
     1     1     0     0     0
     1     1     0     0     0
```

But it's not possible to build a new matrix with A and B as column elements.

```
>> C=[A;B]
??? Error using ==> vertcat
CAT arguments dimensions are not consistent.
```

This happens because A has 2 columns, but B has 3 columns, so the columns don't line up.

We'll see in later work that the matrix building capabilities of Matlab are a powerful ally.

Scalar-Matrix Multiplication

If asked to multiply a matrix by a scalar, one would hope that the operation of scalar-matrix multiplication would be carried out in exactly the same manner as scalar-vector multiplication. That is, simply multiply each entry of the matrix by the scalar.

► **Example 1.** *If A is the matrix*

$$A = 3\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},$$

perform the scalar-matrix multiplication $3A$.

Simply multiply 3 times each entry of the matrix.

$$3A = 3\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{bmatrix}$$

Matlab understands scalar-matrix multiplication. First, enter matrix A.

```
>> A=[1 2 3;4 5 6;7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
```

Now compute $3A$.

```
>> 3*A
ans =
     3     6     9
    12    15    18
    21    24    27
```

Matrix Addition

If two matrices have the same dimension, then add the matrices by adding the corresponding entries in each matrix.

▶ **Example 2.** *If A and B are the matrices*

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

find the sum $A + B$.

Simply add the corresponding entries.

$$A + B = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{bmatrix}.$$

Matlab understands matrix addition.

```
>> A=[1 1 1;2 2 2;3 3 3]; B=[1 1 1;1 1 1;1 1 1];
>> A+B
ans =
     2     2     2
     3     3     3
     4     4     4
```

This is identical to the hand-calculated sum above.

Let's look what happens when the dimensions are not the same.

▶ **Example 3.** *If A and B are the matrices*

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

then find the sum $A + B$.

Note the dimensions of each matrix.

```
>> A=[1 1 1;2 2 2;3 3 3]; B=[1 1 1;1 1 1];
>> size(A)
ans =
     3     3
>> size(B)
ans =
     2     3
```

The matrices A and B do not have the same dimensions. Therfore, it is not possible to sum the two matrices.

```
>> A+B
??? Error using ==> plus
Matrix dimensions must agree.
```

This error message is completely expected.

One final example is in order.

▶ **Example 4.** *If matrix* A *is*

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix},$$

compute $A + 1$.

Note that this addition of a matrix and a scalar makes no sense.

$$A + 1 = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} + 1$$

The dimensions are all wrong. However, this is such a common occurrence in algebraic calculations (as we will see throughout the course), Matlab allows this matrix-scalar addition.

```
>> A=[1 1 1;2 2 2;3 3 3];
>> A+1
ans =
     2     2     2
     3     3     3
     4     4     4
```

Matlab simply adds 1 to each entry of the matrix A. That is, Matlab interprets $A + 1$ as if it were the matrix addition of **Example 2**.

Matrix addition enjoys several properties, which we will ask you to explore in the exercises.

1. Addition is *commutative*. That is, $A + B = B + A$ for all matrices A and B having the same dimension.

2. Addition is *associative*. That is, $(A + B) + C = A + (B + C)$, for all matrices A, B, and C having the same dimension.

3. The zero matrix is the *additive identity*. That is, if A is $m \times n$ and 0 is an $m \times n$ matrix of all zeros, then $A + 0 = A$.

4. Each matrix A has an *additive inverse*. Form the matrix $-A$ by negating each entry of the matrix A. Then, $A + (-A) = 0$.

Matrix-Vector Multiplication

Consider the linear system of three equations in three unknowns.

$$\begin{aligned} 2x + 3y + 4z &= 6 \\ 3x + 2y + 4z &= 8 \\ 5x - 3y + 8x &= 1. \end{aligned} \tag{2.1}$$

Because each of the corresponding entries are equal, the following 3×1 vectors are also equal.

$$\begin{bmatrix} 2x + 3y + 4z \\ 3x + 2y + 4z \\ 5x - 3y + 8x \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix}$$

The left-hand vector can be written as a vector sum.

$$\begin{bmatrix} 2x \\ 3x \\ 5x \end{bmatrix} + \begin{bmatrix} 3y \\ 2y \\ -3y \end{bmatrix} + \begin{bmatrix} 4z \\ 4z \\ 8z \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix}$$

Scalar multiplication can be used to factor the variable out of each vector on the left-hand side.

$$x \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} + y \begin{bmatrix} 3 \\ 2 \\ -3 \end{bmatrix} + z \begin{bmatrix} 4 \\ 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} \tag{2.2}$$

The construct on the left-hand side of this result is so important that we will pause to make a definition.

Definition 5. *Let α_1, α_2, ..., and α_n be scalars and let $\mathbf{v}_1$, $\mathbf{v}_2$, ..., and $\mathbf{v}_n$ be vectors. Then the construction*

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n$$

is called a **linear combination** *of the vectors $\mathbf{v}_1$, $\mathbf{v}_2$, ..., and $\mathbf{v}_n$. The scalars α_1, α_2, ..., and α_n are called the* **weights** *of the linear combination.*

For example, we say that

$$x \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} + y \begin{bmatrix} 3 \\ 2 \\ -3 \end{bmatrix} + z \begin{bmatrix} 4 \\ 4 \\ 8 \end{bmatrix}$$

is a linear combination of the vectors $[2, 3, 5]^T$, $[3, 2, -3]^T$, and $[4, 4, 8]^T$.[3]

Finally, we take one last additional step and write the system (**2.2**) in the form

$$\begin{bmatrix} 2 & 3 & 4 \\ 3 & 2 & 4 \\ 5 & -3 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix}. \tag{2.3}$$

Note that the system (**2.3**) has the form

$$A\mathbf{x} = \mathbf{b},$$

where

[3] Here we use the transpose operator to save a bit of space in the document.

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 2 & 4 \\ 5 & -3 & 8 \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \text{and} \qquad \mathbf{b} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix}.$$

The matrix A in (**2.3**) is called the *coefficient matrix*. If you compare the coefficient matrix in (**2.3**) with the original system (**2.1**), you see that the entries of the coefficient matrix are simply the coefficients of x, y, and z in (**2.1**). On right-hand side of system (**2.3**), the vector $\mathbf{b} = [6, 8, 1]^T$ contains the numbers on the right-hand side of the original system (**2.1**). Thus, it is a simple matter to transform a system of equations into a matrix equation.

However, it is even more important to compare the left-hand sides of system (**2.2**) and system (**2.3**), noting that

$$\begin{bmatrix} 2 & 3 & 4 \\ 3 & 2 & 4 \\ 5 & -3 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = x \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} + y \begin{bmatrix} 3 \\ 2 \\ -3 \end{bmatrix} + z \begin{bmatrix} 4 \\ 4 \\ 8 \end{bmatrix}.$$

This tells us how to multiply a matrix and a vector. One takes a linear combination of the columns of the matrix, using the entries in the vector as weights for the linear combination.

Let's look at an example of matrix-vector multiplication

▶ **Example 6.** *Multiply the matrix and vector*

$$\begin{bmatrix} 1 & 2 & -3 \\ 3 & 0 & 4 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}.$$

To perform the multiplication, take a linear combination of the columns of the matrix, using the entries in the vector as weights.

$$\begin{aligned} \begin{bmatrix} 1 & 2 & -3 \\ 3 & 0 & 4 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} &= 1 \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix} - 2 \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix} + 3 \begin{bmatrix} -3 \\ 4 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix} + \begin{bmatrix} -4 \\ 0 \\ 4 \end{bmatrix} + \begin{bmatrix} -9 \\ 12 \\ 6 \end{bmatrix} \\ &= \begin{bmatrix} -12 \\ 15 \\ 10 \end{bmatrix} \end{aligned}$$

It's important to note that this answer has the same number of entries as does each column of the matrix.

Let's see if Matlab understands this form of matrix-vector multiplication. First, load the matrix and the vector.

```
>> A=[1 2 -3;3 0 4;0 -2 2]; x=[1; -2; 3];
```

Now perform the multiplication.

```
>> A*x
ans =
   -12
    15
    10
```

Note this is identical to our hand calculated result.

Let's look at another example.

▶ **Example 7.** *Multiply* $A\mathbf{x}$, *where*

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \end{bmatrix} \quad \textit{and} \quad \mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

If you try to perform the matrix-vector by taking a linear combination using the entries of the vectors as weights,

$$A\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} +? \begin{bmatrix} 1 \\ -2 \end{bmatrix}. \tag{2.4}$$

The problem is clear. There are not enough weights in the vector to perform the linear combination.

Let's see if Matlab understands this "weighty" problem.

```
>> A=[1 1 1;2 0 -2]; x=[1; 1];
>> A*x
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

Inner dimensions? Let's see if we can intuit what that means. In our example, matrix A has dimensions 2×3 and vector x has dimensions 2×1. If we juxtapose these dimensions in the form $(2\times 3)(2\times 1)$, then the inner dimensions don't match.

Dimension Requirement. If matrix A has dimensions $m \times n$ and vector $\mathbf{x}$ has dimensions $n \times 1$, then we say "the innner dimensions match," and the matrix-vector product $A\mathbf{x}$ is possible. In words, the number of columns of matrix A must equal the number of rows of vector $\mathbf{x}$.

Matrix-Matrix Multiplication

We would like to extend our definition of matrix-vector multiplication in order to find the product of matrices. Here is the needed definition.

Definition 8. *Let A and B be matrices and let $\mathbf{b}_1$, $\mathbf{b}_2$, ..., and $\mathbf{b}_n$ represent the columns of matrix B. Then,*

$$AB = A\big[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\big] = \big[A\mathbf{b}_1, A\mathbf{b}_2, \ldots, A\mathbf{b}_n\big].$$

Thus, to take the product of matrices A and B, simply multiply matrix A times each vector column of matrix B. Let's look at an example.

▶ **Example 9.** *Multiply*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}.$$

We multiply the first matrix times each column of the second matrix, then use linear combinations to perform the matrix-vector multiplications.

$$\begin{aligned} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix} &= \left[\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix} \right] \\ &= \left[1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 2 \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \quad -2 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 1 \begin{bmatrix} 2 \\ 4 \end{bmatrix} \right] \\ &= \begin{bmatrix} 5 & 0 \\ 11 & -2 \end{bmatrix} \end{aligned}$$

Let's see if Matlab understands this form of matrix-matrix multiplication. First, load the matrices A and B.

```
>> A=[1 2;3 4]; B=[1 -2;2 1];
```

Now, multiply.

```
>> A*B
ans =
     5     0
    11    -2
```

Note that this result is indentical to our hand calculation.

Again, the inner dimensions must match or the matrix-matrix multiplication is not possible. Let's look at an example where things go wrong.

▶ **Example 10.** *Multiply*

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}.$$

When we multiply the first matrix times each column of the second matrix, we immediately see difficulty with the dimensions.

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \left[\begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} \right] \tag{2.5}$$

In the first column of the matrix product, the matrix-vector multiplication is not possible. The number of columns of the matrix does not match the number of entries in the vector. Therefore, it is not possible to form the product of these two matrices.

Let's see if Matlab understands this dimension difficulty.

```
>> A=[1 1 1;2 0 -2]; B=[1 2;3 4];
>> A*B
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

The error message is precisely the one we would expect.

Dimension Requirement. If matrix A has dimensions $m \times n$ and matrix B has dimensions $n \times p$, then we say "the inner dimensions match," and the matrix-matrix product AB is possible. In words, the number of columns of matrix A must equal the number of rows of matrix B.

Let's look at another example.

▶ **Example 11.** *Multiply*

$$AB = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & -2 \\ 2 & 0 \end{bmatrix}.$$

Load the matrices A nd B into Matlab and check their dimensions.

```
>> A=[1 1 1;2 0 -2]; B=[1 2;1 -2;2 0];
>> size(A)
ans =
     2     3
>> size(B)
ans =
     3     2
```

Thus, matrix A has dimensions 2×3 and B has dimensions 3×2. Therefore, the inner dimensions match (they both equal 3) and it is possible to form the product of A and B.

```
>> C=A*B
C =
     4     0
    -2     4
```

Note the dimensions of the answer.

```
>> size(C)
ans =
     2     2
```

Recall that A was 2×3 and B was 3×2. Note that the "outer dimensions" are 2×2, which give the dimensions of the product.

Dimensions of the Product. If matrix A is $m \times n$ and matrix B is $n \times p$, then the dimensions of AB will be $m \times p$. We say that the "outer dimensions give the dimension of the product."

Properties of Matrix Multiplication

Matrix multiplication is *associative*. That is, for any matrices A, B, and C, providing the dimensions are right,

$$(AB)C = A(BC).$$

Let's look at an example.

▶ **Example 12.** *Given*

$$A = \begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 2 & 5 \end{bmatrix}, \quad \textit{and} \quad C = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix},$$

use Matlab to demonstrate that $(AB)C = A(BC)$.

Load the matrices A, B, and C into Matlab, then calculate the left-hand side of $(AB)C = A(BC)$.

```
>> A=[2 2;3 3]; B=[1 1;2 5]; C=[3 3;2 5];
>> (A*B)*C
ans =
    42    78
    63   117
```

Next, calculate the right-hand side of $(AB)C = A(BC)$.

```
>> A*(B*C)
ans =
    42    78
    63   117
```

Hence, $(AB)C = A(BC)$.

Matrix Multiplication is Associative. In general, if A, B, and C have dimensions so that the multiplications are possible, matrix multiplication is associative. That is, it is always the case that

$$(AB)C = A(BC).$$

Unfortunately, matrix multiplication is **not** *commutative*. That is, even if A and B are of correct dimensions, it is possible that $AB \neq BA$. Let's look at an example.

▶ **Example 13.** *Let*

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 3 & 5 \\ 2 & 7 \end{bmatrix}.$$

Do the matrices A and B commute? That is, does $AB = BA$?

Load the matrices into Matlab, then compute AB.

```
>> A=[1 2;3 4]; B=[3 5;2 7];
>> A*B
ans =
     7    19
    17    43
```

Now compute BA.

```
>> B*A
ans =
    18    26
    23    32
```

Thus, $AB \neq BA$.

Matrix Multiplication is not Commutative. In general, even if the dimensions of A and B allow us to reverse the order of multiplication, matrices A and B will not commute. That is,

$$AB \neq BA.$$

Any change in the order of multiplication of matrices will probably change the answer.

Some matrices do commute, making this even more complicated.

```
>> A=[5 3;7 4],B=[-4 3;7 -5];
>> A*B
ans =
     1     0
     0     1
>> B*A
ans =
     1     0
     0     1
```

In this case, $AB = BA$.

However, in general, matrix multiplication is not commutative. The loss of the commutative property is not to be taken lightly. Any time you change the order of multiplication, you are risking an incorrect answer. There are many insidious ways that changes of order can creep into our calculations. For example, if you multiply the left-hand side of equation on the left by a matrix A, then multiply the right-hand side of the equation on the right by the same matrix A, you've changed the order and should expect an incorrect answer. We will explore how the loss of the commutative property can adversely affect other familiar algebraic properties in the exercises.

Here is a list of matrix properties you can depend on working all of the time. Let A and B be matrices of the correct dimension so that the additions and multiplications that follow are possible. Let α and β be scalars.

$A(B+C) = AB + AC$	$\alpha(A+B) = \alpha A + \alpha B.$
$(A+B)C = AC + BC.$	$\alpha(\beta A) = (\alpha\beta)A.$
$(\alpha+\beta)A = \alpha A + \beta A$	$(\alpha A)B = \alpha(AB) = A(\alpha B).$

For example, as stated above, matrix multiplication is distributive over addition. That is, $A(B+C) = AB + AC$.

```
>> A=[2 3;-1 4]; B=[1 2;0 9]; C=[-3 2;4 4];
>> A*(B+C)
ans =
     8    47
    18    48
>> A*B+A*C
ans =
     8    47
    18    48
```

We will explore the remaining properties in the exercises.

2.2 Exercises

1. Given the matrices

$$A = \begin{bmatrix} 3 & 3 \\ 2 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix},$$

and

$$C = \begin{bmatrix} 3 & 1 \\ 5 & 8 \end{bmatrix},$$

use Matlab to verify each of the following properties. Note that 0 represents the zero matrix.

a) $A + B = B + A$

b) $(A + B) + C = A + (B + C)$

c) $A + 0 = A$

d) $A + (-A) = 0$

2. The fact that matrix multiplication is not commutative is a **huge** loss. For example, with real numbers, the following familiar algeraic properties hold.

i. $(ab)^2 = a^2b^2$
ii. $(a + b)^2 = a^2 + 2ab + b^2$
iii. $(a + b)(a - b) = a^2 - b^2$

Use Matlab and the matrices

$$A = \begin{bmatrix} 1 & 1 \\ 4 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & 3 \\ 1 & 6 \end{bmatrix}$$

to show that none of these properties is valid for these choices of A and B. Can you explain why each of properties (i-iii) is not valid for matrix multiplication? *Hint: Try to expand the left-hand side of each property to arrive at the right-hand side.*

3. Given the matrices

$$A = \begin{bmatrix} 1 & 0 \\ 2 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 2 & 7 \end{bmatrix},$$

and

$$C = \begin{bmatrix} 1 & 2 \\ 0 & 9 \end{bmatrix},$$

use Matlab to verify each of the following forms of the distributive property.

a) $A(B + C) = AB + AC$

b) $(A + B)C = AC + BC$

4. Given the matrices

$$A = \begin{bmatrix} 2 & 2 \\ 4 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 1 \\ 8 & 9 \end{bmatrix},$$

and the scalars $\alpha = 2$ and $\beta = -3$, use Matlab to verify each of the following properties.

a) $(\alpha + \beta)A = \alpha A + \beta A$

b) $\alpha(A + B) = \alpha A + \alpha B$

c) $\alpha(\beta A) = (\alpha\beta)A$

d) $(\alpha A)B = \alpha(AB) = A(\alpha B)$

5. Enter the matrices **A=pascal(3)** and **B=magic(3)**.

a) Use Matlab to compute $(A+B)^T$.

b) Use Matlab to compute $A^T + B^T$ and compare your result with the result from part (a). Explain what your learned in this exercise.

6. Enter the matrix **A=pascal(4)** and the scalar $\alpha = 5$.

a) Use Matlab to compute $(\alpha A)^T$.

b) Use Matlab to compute αA and compare your result with the result from part (a). Explain what your learned in this exercise.

7. Using hand calculations only, calculate the following matrix-vector product, then verify your result in Matlab.

$$\begin{bmatrix} 1 & 1 & 2 \\ 3 & 4 & 0 \\ 0 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix}$$

8. Write the following system of linear equations in matrix-vector form.

$$\begin{aligned} 2x + 2y + 3z &= -3 \\ 4x + 2y - 8z &= 12 \\ 3x + 2y + 5z &= 10 \end{aligned}$$

9. Using hand calculations only, calculate the following matrix-matrix product, then verify your result in Matlab.

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 4 \\ 0 & 0 & 5 \\ 3 & 5 & 2 \end{bmatrix}$$

10. Enter the matrix **magic(8)**. What Matlab command will zero out all of the even rows? Use Matlab to verify your conjecture.

11. Enter the matrix **pascal(8)**. What Matlab command will zero out all of the odd columns? Use Matlab to verify your conjecture.

12. Enter the matrix **A=pascal(4)**.

a) What is the result of the Matlab command **A(:,2)=[ ]**? Note: [ ] is the empty matrix.

b) Refresh matrix A with **A=pascal(4)**. What is the result of the Matlab command **A(3,:)=[ ]**?

13. Enter the matrix **A=pascal(5)**.

a) What command will add a row of all ones to the bottom of matrix A? Use Matlab to verify your conjecture.

b) What command will add a column of all ones to the right end of matrix A? Use Matlab to verify your conjecture.

14. Enter the matrix **A=magic(3)**. Execute the command **A(5,4)=0**. Explain the resulting matrix.

15. Enter the matrix **A=ones(5)**.

a) Explain how you can insert a row of all 5's betwen rows 2 and 3 of matrix A. Use Matlab to verify your conjecure.

b) Explain how you can insert a column of all 5's betwen columns 3 and 4 of matrix A. Use Matlab to verify your conjecure.

16. Enter the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

a) What is the output of the Matlab command **A=A([1,3,2],:)**?

b) Refresh matrix A to its original value. What Matlab command will swap columns 1 and 3 of matrix A? Use Matlab to verify your conjecture.

17. Enter the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

a) Enter the Matlab command **A(2,:)=A(2,:)-4*A(1,:)**? Explain the result of this command.

b) Continue with the resulting matrix A from part (a). What is the output of the Matlab command **A(3,:)=A(3,:)-7*A(1,:)**? Explain the result of this command.

18. Type **format rat** to change the display to rational format. Create a 3×3 Hilbert matrix with the command **H=hilb(3)**.

a) What is the output of the Matlab command **H(1,:)=6*H(:,1)**? Explain the result of this command.

b) Continue with the resulting matrix H from part (a). What command will clear the fractions from row 2 of this result?

19. Enter the matrices **A=magic(3)** and **B=pascal(3)**. Execute the command **C=A+i*B**. *Note: You may have to enter* **clear i** *to return i to its default (the square root of -1).*

a) What is the transpose of the matrix C? Use Matlab to verify your response.

b) What is the conjugate transpose of the matrix C? Use Matlab to verify your response.

20. Use Matlab's **hadamard(n)** command to form Hadarmard matrices of order $n = 2$, 4, 8, and 16. In each case, use Matlab to calculate $H^T H$. Note the pattern. Explain in your own words what would happen if your formed the matrix product $H^T H$, where H is a Hadamard matrix of order 256.

21. Enter the Matlab command **magic(n)** to form a "magic" matrix of order $n = 8$. Use Matlab's **sum** command to sum both the columns and the rows of your "magic" matrix. Type **help sum** to learn how to use the syntax **sum(X,dim)** to accomplish this goal. What is "magic" about this matrix?

22. Enter the Matlab command **A=magic(n)** to form a "magic" matrix of order $n = 8$. Use Matlab's **sum** command to sum the columns of your "magic" matrix. Explain how you can use matrix-vector multilication to sum the columns of matrix A.

23. Set **A=pascal(5)** and then set **I=eye(5)**, then find the matrix product AI. Why is I called the *identity matrix*? Describe what a 256×256 identity matrix would look like.

24. Set **A=pascal(4)** and then set **B=magic(4)**. What operation will produce the second column of the matrix product AB? Can this be done without finding the product AB?

25. Set the vector **v=(1:5).'** and the vector **w=(2:6).'**.

a) The product $\mathbf{v}^T\mathbf{w}$ is called an *inner product* because of the position of the transpose operator. Use Matalb to compute the inner product of the vectors $\mathbf{v}$ and $\mathbf{w}$.

b) The product $\mathbf{v}\mathbf{w}^T$ is called an *outer product* because of the position of the transpose operator. Use Matalb to compute the outer product of the vectors $\mathbf{v}$ and $\mathbf{w}$.

26. Enter **A=[0.2 0.6;0.8 0.4]**. Calculate A^n for $n = 2$, 3, 4, 5, etc. Does this sequence of matrices converge? If so, to what approximate matrix do they converge?

27. Use Matlab **ones** command to create the matrices

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix},$$

and

$$\begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix}.$$

Craft a Matlab command that will build the block diagonal matrix

$$C = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{bmatrix},$$

where the zeros in this matrix represent matrices of zeros of the appropriate size.

28. Enter the Matlab command **hankel(x)** to form a Hankel matrix H, where $\mathbf{x}$ is the vector $[1, 2, 3, 4]$. The help file for the **hankel** commands describes the Hankel matrix as a *symmetric matrix*. Take the transpose of H. Describe what is mean by a symmetric matrix.

29. A Hilbert matrix H is defined by $H(i, j) = 1/(i + j - 1)$, where i ranges from 1 to the number of rows and j ranges from 1 to the number of columns. Use this definition and hand calculations to find a Hilbert matrix of dimension 4×4. Use **format rat** and Matlab's **hilb** command to check your result.

30. The number of ways to choose k objects from a set of n objects is defined and calcualted with the formula

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Define a Pascal matrix P with the formula

$$P(i, j) = \binom{i + j - 2}{i - 1},$$

where i ranges from 1 to the number of rows and j ranges from 1 to the number of columns. Use this definition and hand calculations to find a Pascal matrix of dimension 4×4. Use Matlab's **pascal** command to check your result.

2.2 Answers

1.

a) Enter the matrices.

```
>> A=[3 3;2 1]; B=[1 1;2 3];
```

Calculate $A + B$.

```
>> A+B
ans =
     4     4
     4     4
```

Calculate $B + A$.

```
>> B+A
ans =
     4     4
     4     4
```

b) Enter the matrices

```
>> A=[3 3;2 1]; B=[1 1;2 3];
>> C=[3 1;5 8];
```

Calculate $(A + B)C$.

```
>> (A+B)+C
ans =
     7     5
     9    12
```

Calculate $AC + BC$.

```
>> A+(B+C)
ans =
     7     5
     9    12
```

c) Enter the matrix A and the zero matrix.

```
>> A=[3 3;2 1]; O=zeros(2,2);
```

Calculate $A + 0$.

```
>> A+O
ans =
     3     3
     2     1
```

Calculate A.

```
>> A
A =
     3     3
     2     1
```

d) Enter the matrix A and the zero matrix.

```
\startMatlab
>> A=[3 3;2 1]; O=zeros(2,2);
```

Calculate $A + (-A)$.

```
>> A+(-A)
ans =
     0     0
     0     0
```

Calculate the zero matrix.

```
>> O
O =
     0     0
     0     0
```

3.

a) Enter the matrices A, B, and C.

```
>> A=[1 0;2 5]; B=[0 1;2 7];
>> C=[1 2;0 9];
```

Compare $A(B+C)$ and $AB+AC$.

```
>> A*(B+C)
ans =
     1     3
    12    86
>> A*B+A*C
ans =
     1     3
    12    86
```

b) Enter the matrices A, B, and C.

```
>> A=[1 0;2 5]; B=[0 1;2 7];
>> C=[1 2;0 9];
```

Compare $(A+B)C$ and $AC+BC$.

```
>> (A+B)*C
ans =
     1    11
     4   116
>> A*C+B*C
ans =
     1    11
     4   116
```

5.

a) Ener the matrices A and B.

```
>> A=pascal(3); B=magic(3);
```

Compute $(A+B)^T$.

```
>> (A+B).'
ans =
     9     4     5
     2     7    12
     7    10     8
```

b) Compute $A^T + B^T$.

```
>> A.'+B.'
ans =
     9     4     5
     2     7    12
     7    10     8
```

The transpose of the sum of two matrices is equal to the sum of the transposes of the two matrices.

7. Enter matrix A and vector $\mathbf{x}$.

```
>> A=[1 1 2;3 4 0;0 5 6];
>> x=[1 2 5].';
```

Calculate $A\mathbf{x}$.

```
>> A*x
ans =
    13
    11
    40
```

9. Enter matrices A and B.

```
>> A=[2 3 1;0 1 2;0 0 5];
>> B=[1 1 4;0 0 5;3 5 2];
```

Calculate AB.

```
>> A*B
ans =
     5     7    25
     6    10     9
    15    25    10
```

11. Enter matrix A.

```
>> A=pascal(8);
```

The following command will zero out all the odd columns.

```
>> A(:,1:2:end)=0;
```

13.

a) Enter the matrix A.

```
>> A=pascal(5)
```

To add a row of all ones to the bottom of the matrix, execute the following command.

```
>> A(6,:)=ones(5,1)
```

b) Enter the matrix A.

```
>> A=pascal(5)
```

To add a column of all ones to the right end of the matrix, execute the following command.

```
>> A(:,6)=ones(5,1)
```

15.

a) Enter the matrix A.

```
>> A=ones(5);
```

We'll build a new matrix using the first two roes of matrix A, then a row of 5's, then the last three rows of matrix A. Note that we separate new columns with commas.

```
>> B=[A(1:2,:);5*ones(1,5);
A(3:5,:)]
B =
     1     1     1     1     1
     1     1     1     1     1
     5     5     5     5     5
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
```

b) Enter the matrix A.

```
>> A=ones(5);
```

We'll build a new matrix using the first 3 columns of matrix A, then a column of 5's, then the last two columns of matrix A. Note that we separate new rows with semi-colons.

```
>> B=[A(:,1:3),5*ones(5,1),
A(:,4:5)]
B =
    1     1     1     5     1     1
    1     1     1     5     1     1
    1     1     1     5     1     1
    1     1     1     5     1     1
    1     1     1     5     1     1
```

17.

a) Enter the matrix A.

```
>> A=[1 2 3;4 5 6;7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
```

The next command will subtract 4 times row 1 from row 2.

```
>> A(2,:)=A(2,:)-4*A(1,:)
A =
     1     2     3
     0    -3    -6
     7     8     9
```

b) Continuing with the last value of matrix A, the next command will subtract 7 times row 1 from row 3.

```
>> A(3,:)=A(3,:)-7*A(1,:)
A =
     1     2     3
     0    -3    -6
     0    -6   -12
```

19.

a) Enter the matrices A and B and compute C.

```
>> A=magic(3); B=pascal(3);
>> C=A+i*B
```

The transpose of

$$C = \begin{bmatrix} 8+i & 1+i & 6+i \\ 3+1 & 5+2i & 7+3i \\ 4+i & 9+3i & 2+6i \end{bmatrix}$$

is

$$C^T = \begin{bmatrix} 8+i & 3+i & 4+i \\ 1+i & 5+2i & 9+3i \\ 6+i & 7+3i & 2+6i \end{bmatrix}.$$

This result is verified with the following Matlab command.

```
>> C.'
```

b) The conjugate transpose of

$$C = \begin{bmatrix} 8+i & 1+i & 6+i \\ 3+1 & 5+2i & 7+3i \\ 4+i & 9+3i & 2+6i \end{bmatrix}$$

is

$$C^T = \begin{bmatrix} 8-i & 3-i & 4-i \\ 1-i & 5-2i & 9-3i \\ 6-i & 7-3i & 2-6i \end{bmatrix}.$$

This result is verified with the following Matlab command.

```
>> C'
```

21. Enter matrix A.

```
>> A=magic(8)
```

You sum the rows along the first dimension with the following command. You'll note that the sum of each column is 260.

```
>> sum(A,1)
```

You sum the columns along the second dimension with the following command. You'll note that the sum of each row is 260.

```
>> sum(A,2)
ans =
   260
   260
   260
   260
   260
   260
   260
   260
```

23. Store A with the following command.

```
>> A=pascal(5)
A =
   1   1   1   1   1
   1   2   3   4   5
   1   3   6  10  15
   1   4  10  20  35
   1   5  15  35  70
```

You store I with the following command.

```
>> I=eye(5)
I =
   1   0   0   0   0
   0   1   0   0   0
   0   0   1   0   0
   0   0   0   1   0
   0   0   0   0   1
```

Note that AI is identical to matrix A.

```
>> A*I
ans =
    1    1    1    1    1
    1    2    3    4    5
    1    3    6   10   15
    1    4   10   20   35
    1    5   15   35   70
```

A 256×256 identity matrix would have 1's on its main diagonal and zeros in all other entries.

25.

a) Store the vectors $\mathbf{v}$ and $\mathbf{w}$.

```
>> v=(1:5).'; w=(2:6).';
```

The inner product $\mathbf{v}^T\mathbf{w}$ is computed as follows.

```
>> v.'*w
ans =
    70
```

You should be able to compute $\mathbf{v}^T\mathbf{w}$ manually and get the same result.

b) The outer product $\mathbf{v}\mathbf{w}^T$ is computed as follows.

```
>> v*w.'
ans =
    2    3    4    5    6
    4    6    8   10   12
    6    9   12   15   18
    8   12   16   20   24
   10   15   20   25   30
```

You should be able to compute $\mathbf{v}\mathbf{w}^T$ manually and get the same result.

27. Load the matrices A and B.

```
>> A=ones(2); B=2*ones(3);
```

Load the matrix C.

```
>> C=3*ones(2);
```

You can construct the required matrix with the following command.

```
>> D=[A,zeros(2,3),zeros(2,2);
zeros(3,2), B, zeros(3,2);
zeros(2,2), zeros(2,3), C]
```

29. The entry in row 1 column 1 would be $H(1,1) = 1/(1+1-1) = 1$. The entry in row 1 column 2 would be $H(1,2) = 1/(1+2-1) = 1/2$. Continuing in this manner, we arrive at a 4×4 Hilbert matrix.

$$H = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix}$$

This result can be verified by these commands.

```
>> format rat
>> H=hilb(4)
```