
Octave

June 5, 2026

Contents

1 Introduction

2 Using Octave

Why Octave?

In writing *Linear Algebra, An Interactive Introduction*, we drew upon many years of experience teaching elementary linear algebra. We have found that when students experiment with scientific computing software such as [MATLAB](#), the subject comes alive! Students can quickly recognize patterns, form conjectures, and test their theories.

We recognize that not every learning institution has the funding to purchase software for student use. For this reason we chose to use Octave as our computational tool. [GNU Octave](#) is a programming language compatible with MATLAB. [GNU Octave](#) can be downloaded for free at <https://www.octave.org/download> or executed in [SageCells](#) online. Online SageCells and a desktop version of Octave can be equally effective in enhancing the experience in linear algebra.

The purpose of this Octave tutorial is to gather commands and examples useful for this linear algebra course into one place. Students and instructors may find this to be a useful reference as they proceed through the course. Please refer to the [Octave manual](#) for additional support.

How to Use SageCells

We present code in two different ways. First, the code appears as text within the narrative, like this:

```
disp('Hello, world!');
```

This code can be copied and pasted into MATLAB or Octave. Second, we provide permalinks to our code in SageCells, like this:

[Link to code.](#)

Users should feel free to modify our code inside SageCells. To save their work, users should generate their own permalink. Permalinks should be stored to access the code at a later time and to share the code with others. Note that every time the code is modified a new permalink must be generated to reflect the changes.

When writing Octave code in SageCells from scratch, make sure to choose *Octave* as your language from the drop down menu below the cell.

This 3:00 video explains how to use permalinks. The key step is just past the 2:15 mark.

YouTube link: <https://www.youtube.com/watch?v=NhujjT-p2D0>

Getting Help

The following link provides documentation for linear algebra functions used in Octave. [Link to linear algebra functions documentation](#).

To get immediate help inside of a SageCell or in Octave Desktop, you can use the help command, as follows:

```
help eig
```

3 Tutorials and Exercises

4 Octave for Chapter 1

Octave for Chapter 1

The templates in this section provide sample Octave code for vector operations. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

Basic Operations

Template 1. Here are some basic vector operations.

```
% Define vectors v and w
v=[3;1;1;2;4];
w=[-2;1;0;3;-1];

% Find v+w
sum=v+w

% Find 5v
five_v=5*v

% Find the dot product of v and w
dot_v_w=dot(v,w)

% Find the length of v
length_v=norm(v)
```

[Link to code.](#)

Remark 1. • In Octave, we use % to add comments to our code. Octave does not execute anything that appears after % on the same line. It is always a good idea to make notes about your code for yourself and others.

- Note that lines 2 and 3 of our code have a semi-column at the end while lines 6, 9, 12, and 15 do not. Remove the semi-column from lines 2 and 3, add semi-columns to the other lines. Run the code. What happens?

Template 2. We can find the cross-product of two vectors in $\mathbb{R}^3$ as follows.

```
% Define vectors v and w
v=[3;1;1];
w=[-2;1;0];

% Find the cross product v x w
cross_v_w=cross(v,w)
```

[Link to code.](#)

Example 1. Find the angle between vectors $\mathbf{v} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 2 \\ 4 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} -2 \\ 1 \\ 0 \\ 3 \\ -1 \end{bmatrix}$.

Explanation. % Define vectors v and w
v=[3;1;1;2;4];
w=[-2;1;0;3;-1];

```
% Find the angle between v and w in radians and degrees.
% To do so, use the arccosine function acos()
angle_radians = acos(dot(v,w)/(norm(v)*norm(w)))
angle_degrees = rad2deg(angle_radians)
```

[Link to code.](#)

Sometimes it is useful to reference a single component of a vector.

Example 2. Given vector $\mathbf{v} = \begin{bmatrix} 1 \\ -2 \\ -5 \end{bmatrix}$, find the third component of $\mathbf{v}$.

Explanation. % Define vector v
v=[1;-2;-5];

```
% Find the third component of v
z=v(3)
```

[Link to code.](#)

Loops

In this section we will introduce one of the most fundamental concepts in programming, a *loop*. Loops are used to repeat a procedure multiple times. We will start with a simple *for* loop. Every time you perform the steps inside the

loop, you come back to the beginning, and increase the index i by 1 until you reach the desired number of iterations.

Example 3. Here is an example of a loop. Try to guess the outcome of the loop, then run the code to verify your guess.

```
% Enter the starting value
n=10;

% At every iteration, n gets replaced with n+1
for i=1:10
    n=n+1
end
```

[Link to code.](#)

Example 4. Here is another loop. Try to guess the outcome of the loop, then run the code to verify your guess.

```
% Enter the first component of vector v
v(1)=1;

% Enter the second component of vector v
v(2)=1;

% we continue to assign values to vector components
for i=1:10
    v(i+2)=v(i)+v(i+1);
end
v
```

[Link to code.](#)

Do you recognize this famous sequence?

Octave Exercises

Problem 1 Use Octave to find the angle (in degrees and radians) between vectors

$$\mathbf{v} = \begin{bmatrix} 2 \\ -4 \\ 10 \\ -25 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} 5 \\ 3 \\ -20 \\ 11 \end{bmatrix}$$

Problem 2 Write Octave code to show that the points $A(1, -2, 3, 5)$, $B(-2, 0, 1, 1)$, $C(2, 4, -3, 2)$ are the vertices of a right triangle in $\mathbb{R}^4$.

Problem 3 Use vectors $\mathbf{v}$ and $\mathbf{w}$ from Template ?? to illustrate that the cross product of two vectors is orthogonal to both vectors. Pick another pair of vectors $\mathbf{v}$ and $\mathbf{w}$ and repeat the demonstration.

Problem 4 Write an Octave routine that allows the user to input three points and returns a normal vector to the plane containing these points. Your code might start as follows:

```
% Enter coordinates of point A
A=[1; 4; 6];

% Enter coordinates of point B
B=[-2; 3; 7]

% Enter coordinates of point C
C=[1; -2; 4]

% Normal vector to the plane containing A, B, C is given by
```

What type of user inputs (other than all zeros) would result in the computed normal vector of $\mathbf{n} = \mathbf{0}$? Explain this phenomenon geometrically.

Use your code to find an equation of a plane containing points $(-2, 4, 0)$, $(-1, 8, 1)$, $(3, -1, 2)$.

Problem 5 In this problem you will use the following definition of the dot product.

Definition 1. (??) Let $\mathbf{u}$ and $\mathbf{v}$ be vectors in $\mathbb{R}^n$. The dot product of $\mathbf{u}$ and $\mathbf{v}$, denoted by $\mathbf{u} \cdot \mathbf{v}$, is given by

$$\mathbf{u} \cdot \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = u_1 v_1 + u_2 v_2 + \dots + u_n v_n.$$

Question 5.1 Due to a false fire alarm, a programmer left line 10 of the following code unfinished. Finish the code so that it computes the dot product (dp) of two vectors. Note that this code uses Definition ?? directly, instead of relying on the built-in dot function.

```
% Define vectors v and w
v=[3; 6; -1; 7; 12];
w=[-2; 4; -3; 5; 11];

% Initialize the dot product (dp)
dp=0;

% Find the dot product v*w directly from the definition
for i=1:5
    dp=dp+;
```

```
end

% Print your answer
dp

% Check your answer
correct_answer=dot(v,w)
```

[*Link to code.*](#)

Why did we have to initialize dp? What would happen if we didn't?

5 Octave for Chapter 2

Octave for Chapter 2

The templates in this section provide sample Octave code for solving systems of equations. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

When solving systems by hand, you utilized Gauss-Jordan elimination to find rref. Having the reduced row echelon form available made it easy to write out the solution(s) or determine that the system is inconsistent. We can find rref and the rank of a matrix using Octave as follows.

Template 3. Consider the system

$$\begin{array}{cccccccl}
 x & - & y & & & & = & 0 \\
 2x & - & 2y & + & z & + & 2w & = & 4 \\
 & & y & & & + & w & = & 0 \\
 & & & & 2z & + & w & = & 5
 \end{array} \tag{1}$$

(You may remember this system from [Augmented Matrix Notation and Elementary Row Operations](#).) We will solve this system by finding the reduced row echelon form of the corresponding augmented matrix in Octave.

```
% Define the augmented matrix
% corresponding to the coefficient matrix A and vector b
A_b = [1 -1 0 0 0;
       2 -2 1 2 4;
       0 1 0 1 0;
       0 0 2 1 5];

% Find rref of A_b
rref(A_b)

% Find the rank of A_b
rank(A_b)
```

[Link to code.](#)

If you start with a matrix A and a separate vector $\mathbf{b}$, you can create an augmented matrix without re-typing as follows:

```
% Define the coefficient matrix A
A = [1 -1 0 0;
     2 -2 1 2;
     0 1 0 1;
     0 0 2 1];

% Define vector b
b = [0;4;0;5];

% Create an augmented matrix
A_b=[A b]
```

[Link to code.](#)

Template 4. We can access individual rows, columns, and entries of a matrix. Let

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

```
% Define a matrix
A=[2 -1 3 -1; 1 0 2 1; 1 -1 1 -2]

% Third row of A
row_3=A(3,:)

% Second column of A
col_2=A(:,2)

% Second row third entry of A
a_23=A(2,3)
```

[Link to code.](#)

Example 5. Use Octave code to show one step of Gaussian elimination by subtracting half of the first row from the second row of the given matrix A .

Explanation. % Define matrix A

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

```
% Replace the second row of A with second row minus half the first row.
A(2,:)=-1/2*A(1,:)+A(2,:)
```

[Link to code.](#)

Octave Exercises

Problem 6 Given the code below, write out the system of equations and the corresponding solution(s).

```
% Coefficient matrix A
A=[2 -4 1 1 3;
  0 2 -1 4 -1;
  2 -2 1 1 4];

% Column vector b
b=[-2; 1; 4];

% Augmented matrix A_b
A_b=[A b];

% Reduced row-echelon form
rref_A_b=rref(A_b)
```

[Link to code.](#)

Problem 7 Solve the following system using Octave.

$$\begin{array}{rrrrrrrrcl}
 x_1 & - & x_2 & + & 2x_3 & & + & x_5 & = & -1 \\
 -x_1 & & & & - & x_3 & + & x_4 & & = 1 \\
 2x_1 & + & 2x_2 & & & - & x_4 & + & 2x_5 & = 1 \\
 x_1 & & & & & + & 2x_4 & - & x_5 & = 0
 \end{array} \tag{2}$$

Problem 8 Solve the following system using Octave.

$$\begin{array}{rrrrrrrrcl}
 3x_1 & + & x_2 & + & 4x_3 & - & x_4 & + & 2x_5 & = 10 \\
 x_1 & & & + & 2x_3 & - & x_4 & + & x_5 & = 3 \\
 & & x_2 & + & x_3 & + & x_4 & + & 3x_5 & = 2 \\
 2x_1 & - & x_2 & - & x_3 & & & + & 3x_5 & = -2 \\
 4x_1 & + & x_2 & - & 5x_3 & + & 2x_4 & + & x_5 & = 1
 \end{array} \tag{3}$$

Problem 9 Suppose the graph of function f of the form $f(x) = ax^3 + bx^2 + cx + d$ passes through points $(-2, 1)$, $(0, 4)$, $(1, -2)$ and $(4, 5)$. Set up a system of linear equations to find the coefficients a , b , c and d . Update the coefficients in the GeoGebra interactive below to sketch the graph.

GeoGebra link: <https://www.geogebra.org/m/xry6un6h>

Problem 10 The following code generates a random 20×3 matrix A whose entries are integers -1 , 0 , and 1 , and a random 20×1 vector b whose components are integers -1 , 0 , and 1 .

```

% A is a randomly generated 20 by 3 matrix whose entries are -1, 0, 1.
A=randi([-1,1],20,3);

% b is a randomly generated column vector whose entries are -1, 0, 1.
b=randi([-1,1],20,1);

% Augmented matrix A_b
A_b=[A b];

% Reduced row-echelon form
rref_A_b=rref(A_b)

% Rank of A_b
rank(A_b)

```

[Link to code.](#)

Run the code several times to generate several random combinations of A and $\mathbf{b}$ and find the rank and the reduced row-echelon form of $[A|\mathbf{b}]$. Are you getting the answers you were expecting? Answer the following questions based on your theoretical knowledge.

What are the possibilities for the rank of such $[A|\mathbf{b}]$?

Multiple Choice:

1. The rank can be any positive integer less than or equal to 20.
2. The rank can be any positive integer less than or equal to 3.
3. The rank is always equal to 4.
4. The rank can be any non-negative integer less than or equal to 4.

What are the possibilities for the number of solutions of such $[A|\mathbf{b}]$?

Multiple Choice:

1. This system always has a unique solution.
2. This system may have no solutions, infinitely many solutions or a unique solution.
3. This system will always have infinitely many solutions.
4. This system will always be infeasible.

There is a good chance that your experiments have led you to believe that $\text{rank}[A|\mathbf{b}] = 4$. Can you manually construct $[A|\mathbf{b}]$, satisfying the condition of the problem, with a smaller rank? How many solutions does your system have? Why do you think it is so rare to randomly generate a system $[A|\mathbf{b}]$ of rank less than 4?

Further Considerations

Issues related to computational aspects of linear algebra are outside the scope of this text. However, we want the reader to be aware of some potential problems that may arise.

Warning 1. The `rref` function utilizes a modified version of the Gauss-Jordan elimination algorithm ([Reference Link](#)). For some matrices, implementing this algorithm directly leads to round-off errors and other problems that are outside of the scope of this text. We illustrate what can happen in the next example.

Example 6. We will use a 12×12 Hilbert matrix to illustrate how the `rref` function can fail. You can learn more about Hilbert matrices [here](#).

```
% Use a 12 by 12 Hilbert matrix
A=hilb(12);

% Theoretical rank of A is 12, but the following computation shows a rank of 11
rank(A)

% rref of A should be the identity matrix, but the following result contains a row of zeros
rref(A)
```

[Link to code.](#)

Acknowledgements

The inspiration for Example ?? came from [this Stack Overflow](#) comment.

6 Octave for Chapter 3

Octave for Chapter 3

Examples in this section provide sample Octave code for answering questions about linear combinations, span, and linear independence. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

Example 7. If possible, express each of $\mathbf{b}_1$ and $\mathbf{b}_2$ as a linear combination of vectors in W .

$$\mathbf{b}_1 = \begin{bmatrix} 0 \\ 4 \\ 1 \\ -1 \\ 7 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} -4 \\ 26 \\ -11 \\ -41 \\ 10 \end{bmatrix}, \quad W = \left\{ \begin{bmatrix} 3 \\ 6 \\ -1 \\ 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \\ 3 \\ 5 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 4 \\ -2 \\ 3 \end{bmatrix} \right\}$$

Explanation. For each of $\mathbf{b}_1$ and $\mathbf{b}_2$ we will attempt to solve the system

$$a\mathbf{w}_1 + b\mathbf{w}_2 + c\mathbf{w}_3 = \mathbf{b}_i$$

where $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ are vectors of W .

We will use the following code.

```
% Let A_b1 and A_b2 be augmented matrices
% whose columns are the vectors of W
% together with vectors b1 and b2, respectively.

w1=[3; 6;-1; 0; 4];
w2=[1; -2; 3; 5; 1];
w3=[-1; 0; 4; -2; 3];

b1=[0; 4; 1; -1; 7];
b2=[-4; 26; -11; -41; 10];

A_b1=[w1 w2 w3 b1];
```

```

A_b2=[w1 w2 w3 b2];

% Reduced row-echelon form
rref_A_b1=rref(A_b1)
rref_A_b2=rref(A_b2)

```

[Link to code.](#)

The reduced row-echelon form of $[A|\mathbf{b}_1]$ looks like this:

```

rref_A_b1 =

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
0 0 0 0

```

The row $[000|1]$ indicates that the system is infeasible, so $\mathbf{b}_1$ cannot be written as a linear combination of vectors in W .

The reduced row-echelon form of $[A|\mathbf{b}_2]$ looks like this:

```

rref_A_b2 =

1 0 0 2
0 1 0 -7
0 0 1 3
0 0 0 0
0 0 0 0

```

This shows that the system of equations

$$a\mathbf{w}_1 + b\mathbf{w}_2 + c\mathbf{w}_3 = \mathbf{b}_2$$

has a solution: $a = 2$, $b = -7$, $c = 3$. We write,

$$\mathbf{b}_2 = 2\mathbf{w}_1 - 7\mathbf{w}_2 + 3\mathbf{w}_3$$

Example 8. Determine whether the given vectors are linearly independent.

$$\mathbf{w}_1 = \begin{bmatrix} 38 \\ 15 \\ -11 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} 64 \\ 82 \\ 14 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} -3 \\ 13 \\ 9 \end{bmatrix}$$

Explanation. We will have to determine whether the equation

$$a\mathbf{w}_1 + b\mathbf{w}_2 + c\mathbf{w}_3 = \mathbf{0} \tag{4}$$

has a non-trivial solution. To do this we will consider an augmented matrix $[\mathbf{w}_1 \mathbf{w}_2 \mathbf{w}_3 | \mathbf{0}]$. In the following code, we will skip the zero vector.

```
% Define the vectors individually

w1=[38; 15; -11];
w2=[64; 82; 14];
w3=[-3; 13; 9];

A=[w1 w2 w3];

% Reduced row-echelon form
rref_A=rref(A)
```

[Link to code.](#)

We get the following output

```
rref_A =

1 0 -0.5
0 1 0.25
0 0 0
```

We can see that there are infinitely many solutions $a = 0.5t$, $b = -0.25t$, $c = t$. Letting $t = 4$, we write,

$$2\mathbf{w}_1 - \mathbf{w}_2 + 4\mathbf{w}_3 = \mathbf{0}$$

Since (??) has a non-trivial solution, we conclude that the vectors are linearly dependent.

Example 9. From the set of vectors below, construct a linearly independent set that has the same span as W .

$$W = \left\{ \begin{bmatrix} 12 \\ 10 \\ -20 \\ 5 \end{bmatrix}, \begin{bmatrix} 3 \\ -14 \\ 21 \\ 11 \end{bmatrix}, \begin{bmatrix} -6 \\ -38 \\ 62 \\ 17 \end{bmatrix}, \begin{bmatrix} 4 \\ -4 \\ 23 \\ 19 \end{bmatrix}, \begin{bmatrix} -6 \\ 6 \\ 47 \\ 30 \end{bmatrix}, \begin{bmatrix} 2 \\ 11 \\ 17 \\ 8 \end{bmatrix} \right\}$$

Explanation. We will refer to vectors of W as $\mathbf{w}_1 \dots \mathbf{w}_6$, in order of appearance. We will build a linearly independent set from these vectors by adding one vector to our set at a time, and checking for linear independence every time a vector is added. If, after adding one vector, the resulting collection is linearly independent, we will keep the newly added vector, if the collection is not linearly independent we will discard the new vector and move on to the next. The following code accomplishes this.

```
% Define the vectors individually

w1=[12; 10; -20; 5];
w2=[3; -14; 21; 11];
w3=[-6; -38; 62; 17];
w4=[4; -4; 23; 19];
w5=[-6; 6; 47; 30];
```

```

w6=[2; 11; 17; 8];

% Clearly {w1} is linearly independent
% Let's add w2 to the set and check.

rref([w1 w2]) % Looks like w1 and w2 are linearly independent

rref([w1 w2 w3]) % w3 is redundant - this set is linearly dependent. We won't add w3 to the set

rref([w1 w2 w4]) % these are linearly independent

rref([w1 w2 w4 w5]) % w5 is redundant; don't add

rref([w1 w2 w4 w6])

```

[Link to code.](#)

Question 11 Explain why

$$\text{span}(\mathbf{w}_1 \dots \mathbf{w}_6) = \text{span}(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_4, \mathbf{w}_6)$$

See discussion below Definition ?? for guidance.

By starting with $\mathbf{w}_1$, adding vectors to our set one at a time, and checking for linear independence, we determined that $\mathbf{w}_3$ and $\mathbf{w}_5$ are redundant and can be removed from the set without changing the span.

Octave Exercises

Problem 12 If possible, express vector $\mathbf{b}$ as a linear combination of the vectors in W . If you find a linear combination, verify its correctness by hand.

1.

$$\mathbf{b} = \begin{bmatrix} 2 \\ -1 \\ 0 \\ 4 \end{bmatrix}, \quad W = \left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -2 \\ 4 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 4 \\ -1 \end{bmatrix} \right\}$$

Multiple Choice:

- (i) $\mathbf{b}$ is in $\text{span}(W)$.
- (ii) $\mathbf{b}$ is NOT in $\text{span}(W)$.

2.

$$\mathbf{b} = \begin{bmatrix} 4 \\ -6.5 \\ 4.5 \\ 1.5 \end{bmatrix}, \quad W = \left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -2 \\ 4 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 4 \\ -1 \end{bmatrix} \right\}$$

Multiple Choice:

- (i) **b** is in $\text{span}(W)$.
- (ii) **b** is NOT in $\text{span}(W)$.

Problem 13 Determine whether each of the following sets of vectors is linearly independent.

1.

$$\left\{ \begin{bmatrix} -4 \\ -2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 8 \\ 2 \\ 3 \\ 5 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 1 \\ 5 \\ -2 \end{bmatrix}, \begin{bmatrix} 5 \\ 2 \\ 1 \\ 3 \\ 2 \end{bmatrix} \right\}$$

2.

$$\left\{ \begin{bmatrix} -4 \\ -2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 1 \\ 5 \\ -2 \end{bmatrix}, \begin{bmatrix} 5 \\ 2 \\ 1 \\ 3 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 6 \\ 1 \\ 0 \\ 4 \end{bmatrix} \right\}$$

Problem 14 Find the number of redundant vectors in the following set.

$$\left\{ \begin{bmatrix} 2 \\ 6 \\ -1 \\ 4 \\ 3 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 \\ 6 \\ -11 \\ 9 \\ 4 \\ 4 \end{bmatrix}, \begin{bmatrix} -1 \\ 4 \\ 1 \\ 1 \\ -2 \\ 8 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \\ 4 \\ -1 \\ 2 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 32 \\ 3 \\ 13 \\ -4 \\ 50 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \\ 7 \\ -3 \\ -5 \\ 10 \end{bmatrix} \right\}$$

There are redundant vectors in the set.

Problem 15 In Example ??, we constructed a linearly independent set from vectors of W by starting with $\mathbf{w}_1$ and adding one vector at a time to our set moving from left to right and checking for linear independence at every step. Our resulting set $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_4, \mathbf{w}_6\}$ has the same span as the original set W . What if we start this process with $\mathbf{w}_6$ and move from right to left to identify and remove redundant vectors? Would we end up with the same final set? Modify the code of Example ?? to find out. Does our final set have the same span as W ? Prove your claim.

7 Octave for Chapter 4

Octave for Chapter 4

The templates in this section provide sample Octave code for matrix operations and LU -factorization. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

Basic Operations

Template 5. Here are some matrix operations.

```
% Define matrices A, B, and C
A = [1 -1 0 0;
     2 -2 1 2;
     0 1 0 1];

B = [2 3 -1 4;
     1 -1 2 -2;
     1 -2 0 3];

C = [1 -2 1 4;
     2 -2 0 1;
     -2 1 1 0;
     1 3 -2 -1];

% Find A+B
sum=A+B

% Find the transpose of A
A_trans=transpose(A)

% Find AC
product=A*C

% Find the inverse of C
```

```
inv_C=inv(C)
```

[Link to code.](#)

Template 6. Sometimes it is useful to generate a commonly used matrix or a vector (e.g. identity matrix, zero vector).

```
% 5 by 5 identity matrix
I_5=eye(5)

% 5 by 5 zero matrix
O_5=zeros(5)

% The zero vector in R^5
O_col=zeros(5,1)

% Row of zeros
O_row=zeros(1,5)

% Non-square zero matrix
O_block=zeros(2,3)

% Matrix filled with 1s
One_block=ones(2,3)
```

[Link to code.](#)

Using Loops

We first introduced loops in [Octave for Chapter 1](#). Examples in this section and the corresponding exercises will allow you to practice using loops while also encouraging you to delve deeper into various ways of thinking about matrix multiplication.

Example 10. In [Matrix Multiplication](#) we gave the following definition of matrix-vector multiplication.

$$A\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \dots & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \end{bmatrix}$$

Write a routine that uses this definition to compute the product $A\mathbf{x}$ of a user-defined matrix A and vector $\mathbf{x}$.

Explanation. We will use the following outline for our code.

- The user will enter matrix A and vector $\mathbf{x}$ at the top.
- We will use the `size` function determine the size of the user-defined matrix A .

```
% Find the size of matrix A
[m, n] = size(A); % m is the number of rows, and n is the number of columns of A
```

- We will utilize the *length* function to find the length of **x**.

```
% Find the length of vector x
L=length(x);
```

- We will check that the dimensions of **A** and **x** are compatible. If the number of columns of **A** doesn't match the number of entries in **x**, we will issue a warning.

```
if n ~= L
    warning("Dimensions of A and x are incompatible.");
else
    % insert product code here ...
end
```

Here is the complete code.

```
% User-defined matrix A
A=[4 -1 6 10 -2;
  3 11 8 -3 4;
  -5 2 14 6 -1;
  3 2 -3 0 7];

% User-defined vector x
x=[-2; 4; 8; 2; 3];

% Find the size of matrix A
[m,n]=size(A); % m is the number of rows, and n is the number of columns of A

% Find the length of vector x
L=length(x);

if n ~= L
    warning("Dimensions of A and x are incompatible.");
else
    % if the dimensions are compatible, proceed with the multiplication

    % Initialize the product matrix
    A_x=zeros(m,1);

    % Find the product Ax
    for i=1:m
        for j=1:n
```

```

        A_x(i)=A_x(i)+A(i,j)*x(j);
    end
end

% Print our answer
A_x

% Check our answer
correct_answer=A*x
end

```

[Link to code.](#)

Example 11. In [Matrix Multiplication](#) we noted that a matrix-vector product can also be computed by dotting the rows of the matrix with the vector as follows:

$$A\mathbf{x} = \begin{bmatrix} - & \mathbf{r}_1 & - \\ - & \mathbf{r}_2 & - \\ & \vdots & \\ - & \mathbf{r}_n & - \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{r}_1 \cdot \mathbf{x} \\ \mathbf{r}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{r}_n \cdot \mathbf{x} \end{bmatrix}$$

Demonstrate this method for matrix-vector multiplication in Octave.

Explanation. We will use the following syntax to reference the entire row of a matrix.

```
A(2,:) % this is row 2 of matrix A
```

Alternatively, if we ever need to reference a single column of a matrix, we would use the following:

```
A(:,5) % this is column 5 of matrix A
```

Here is our code that utilizes the dot product to perform matrix-vector multiplication.

```

% Define matrix A
A=[4 -1 6 10 -2;
  3 11 8 -3 4;
 -5 2 14 6 -1;
  3 2 -3 0 7];

% Define vector x
x=[-2; 4; 8; 2; 3];

% Find the product Ax using the dot product method
for i=1:4

```

```

        A_x(i,1)=dot(A(i,:),x);
    end

    % Print our answer
    A_x

    % Check our answer
    correct_answer=A*x

```

[Link to code.](#)

LU-factorization

Before we proceed with the code, recall that if the elementary matrices used to reduce the given matrix to row-echelon form are all lower triangular, then we can find an LU factorization. In general, the following theorem applies.

Theorem 1 (??). Suppose an $m \times n$ matrix A is transformed to a row-echelon matrix U using Gaussian elimination. Let $P_1, P_2, \dots, P_s$ be the elementary matrices corresponding (in order) to the row interchanges used, and write $P = P_s \cdots P_2 P_1$. (If no interchanges are used take $P = I$.) Then:

1. PA is the matrix obtained from A by doing these interchanges (in order) to A .
2. PA has an LU factorization.

Template 7. We can find LU -factorization of a matrix as follows.

```

% Define matrix A
A=[2 2 -1 2;
   -1 2 2 1;
   1 2 0 -2;
   1 1 -2 -4];

% Find LU factorization for A
[L, U, P] = lu (A)
% where L is a lower triangular matrix,
% U is an upper triangular matrix, and
% P is an appropriate permutation matrix

% Verify that this factorization is correct
productLU=L*U
productPA=P*A

```

[Link to code.](#)

Remark 2. • Observe that the permutation matrix P is not the identity matrix. What elementary row operation does multiplication by P induce?

- With the last two lines of our code we checked the correctness of our factorization. Is it true that $PA = LU$? Interpret the value $-2.77556e - 17$ in the product LU . What do you think caused this value to occur?

Octave Exercises

Problem 16 Let

$$A = \begin{bmatrix} 2 & 6 & -4 & 1 & 1 \\ -1 & 0 & 2 & 2 & 4 \\ -2 & 1 & 3 & -2 & 1 \\ 1 & 1 & 0 & 0 & 6 \\ -3 & 4 & 1 & -2 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -12 \\ 6 \\ -2 \\ -7 \\ -8 \end{bmatrix}$$

Solve $A\mathbf{x} = \mathbf{b}$ in two different ways. First, use the reduced row-echelon form of $[A|\mathbf{b}]$, second multiply both sides of the equation by A^{-1} . Verify that both methods produce the same answer.

Problem 17 Let $A = \begin{bmatrix} 0 & 1 \\ 5 & 3 \end{bmatrix}$. Find A^{-1} in two different ways. First, use the `inv` function. Second, find `ref([A|I])`.

Problem 18 Generate a 12×14 matrix filled with 3's.

Problem 19 Write a routine that multiplies an $m \times n$ matrix by an $n \times 1$ vector by computing the linear combination of the columns of the matrix, as in the following definition from [Matrix Multiplication](#):

$$A\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \dots & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \dots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

Problem 20 Write a routine that multiplies an $m \times n$ matrix by an $n \times k$ matrix.

Problem 21 A Vandermonde matrix of order n is a square matrix of the form

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix}$$

If you studied section [Curve Fitting](#) this matrix may look familiar to you.

Use loops to create a 5×5 Vandermonde matrix for $x_1 = 2$, $x_2 = -1$, $x_3 = 5$, $x_4 = -2$, $x_5 = 3$.

Problem 22 Use LU-factorization to solve the system

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

Text Source

Problem ?? was adapted from Problem 2.7.3(d) of Keith Nicholson's *Linear Algebra with Applications*. (CC-BY-NC-SA)

W. Keith Nicholson, *Linear Algebra with Applications*, Lyryx 2018, Open Edition, p. 127

8 Octave for Chapters 5-6

Octave for Chapters 5-6

The templates in this section provide sample Octave code for finding subspaces associated with matrices as well as the kernel and image of a linear transformation. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

Column Space of a Matrix (Image of a Linear Transformation)

Recall that the column space of a matrix is the span of the columns of the matrix. In [Subspaces of \$\mathbb{R}^n\$ Associated with Matrices](#) we introduced the following procedure for finding the column space.

Procedure 1 (??). Given a matrix B , a basis for $\text{col}(B)$ can be found as follows:

1. Find $\text{rref}(B)$ (or any row-echelon form B' of B .)
2. Identify the pivot columns of $\text{rref}(B)$ (or B').
3. The columns of B corresponding to the pivot columns of $\text{rref}(B)$ (or B') form a basis for $\text{col}(B)$.

We will implement this procedure in Octave.

Template 8. Let

$$A = \begin{bmatrix} 2 & -1 & 1 & -4 & 1 \\ 1 & 0 & 3 & 3 & 0 \\ -2 & 1 & -1 & 5 & 2 \\ 4 & -1 & 7 & 2 & 1 \end{bmatrix}$$

In Example ?? we found that columns 1, 2 and 4 of A constitute a basis of $\text{col}(A)$. We will now verify this using Octave.

We can find the reduced row-echelon form of A using the *rref* function, and identify the pivot columns visually. We can also take advantage of the expanded version of the *rref* function to identify the pivot columns automatically, as shown below.

```
% Define A
A = [2 -1 1 -4 1;
     1 0 3 3 0;
     -2 1 -1 5 2;
     4 -1 7 2 1];

% Compute rref of A
[rref_A, pivot_cols] = rref(A);

% the pivot column indices are...
pivot_cols

% Extract the pivot columns from A, using pivot column indices
colSpace_basis = A(:, pivot_cols)
```

[Link to code.](#)

When you study linear transformations, you will learn that the column space of a matrix is the image of the linear transformation induced by the matrix. So, Template ?? can be used to find the image of a linear transformation.

Null Space of a Matrix (Kernel of a Linear Transformation)

Recall the following definition of the null space of a matrix from [Subspaces of \$\mathbb{R}^n\$ Associated with Matrices](#).

Definition 2 (??). Let A be an $m \times n$ matrix. The *null space* of A , denoted by $\text{null}(A)$, is the set of all vectors $\mathbf{x}$ in $\mathbb{R}^n$ such that $A\mathbf{x} = \mathbf{0}$.

We can use the reduced row-echelon form (*rref* function) to find the null space, as shown in Example ?? of [Subspaces of \$\mathbb{R}^n\$ Associated with Matrices](#). We will omit the details here.

We can also use the *null* function to find a basis for the null space of a matrix. This function finds an *orthonormal* basis for the null space. An orthonormal set of vectors consists of unit vectors that are pairwise orthogonal. We demonstrate the *null* function below.

Template 9. We will use the matrix from Example ??. Let

$$A = \begin{bmatrix} 2 & -1 & 1 & -4 & 1 \\ 1 & 0 & 3 & 3 & 0 \\ -2 & 1 & -1 & 5 & 2 \\ 4 & -1 & 7 & 2 & 1 \end{bmatrix}$$

Find $\text{null}(A)$.

```
% Define A
A = [2 -1 1 -4 1;
     1 0 3 3 0;
     -2 1 -1 5 2;
     4 -1 7 2 1];
```

```
% Basis for null(A)
% Octave constructs an orthonormal basis, meaning that the basis elements are pairwise orthogonal unit vectors
B=null(A)

% To find the dimension of the null space, find the number of columns in B
[rows, cols]=size(B)
```

[Link to code.](#)

The basis for the null space we found in Template ?? looks different from what you would have found by hand using the *rref* function. In Problem ?? you will verify that *B* in Template ?? spans the null space of *A*.

If you are familiar with linear transformations, you may recall that the *kernel* of a linear transformation induced by a matrix is the null space of the matrix. The *null* function gives us a way to compute the kernel.

Octave Exercises

For Problems ?? and ??, use Octave to find $\text{col}(A)$, $\text{null}(A)$, and their respective dimensions. Verify that the Rank-Nullity Theorem (??) holds for *A*.

Problem 23 Let

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

Problem 24 Let

$$A = \begin{bmatrix} -2 & -9 & -33 & -22 \\ 0 & 4 & 15 & 10 \\ 1 & 3 & 12 & 10 \\ -3 & -9 & -36 & -24 \end{bmatrix}$$

Problem 25 Let

$$H = \text{span} \left\{ \begin{bmatrix} 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 8 \\ 15 \\ 6 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ 6 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 4 \\ 6 \\ 6 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 15 \\ 6 \\ 3 \end{bmatrix} \right\}$$

Find the dimension of *H* and determine a basis.

Problem 26 Find the rank of the following matrix. Also find a basis for the row and column spaces.

$$\begin{bmatrix} 1 & 3 & 0 & -2 & 0 & 3 \\ 3 & 9 & 1 & -7 & 0 & 8 \\ 1 & 3 & 1 & -3 & 1 & -1 \\ 1 & 3 & -1 & -1 & -2 & 10 \end{bmatrix}$$

Problem 27 Find the rank of the following matrix. Also find a basis for the row and column spaces.

$$\begin{bmatrix} 1 & 3 & 0 & -2 & 7 & 3 \\ 3 & 9 & 1 & -7 & 23 & 8 \\ 1 & 3 & 1 & -3 & 9 & 2 \\ 1 & 3 & -1 & -1 & 5 & 4 \end{bmatrix}$$

Problem 28 Find the rank of the following matrix. Also find a basis for the row and column spaces.

$$\begin{bmatrix} 1 & 0 & 3 & 0 & 7 & 0 \\ 3 & 1 & 10 & 0 & 23 & 0 \\ 1 & 1 & 4 & 1 & 7 & 0 \\ 1 & -1 & 2 & -2 & 9 & 1 \end{bmatrix}$$

Problem 29 In Template ?? we used the `null` function to find a basis B for the null space of matrix

$$A = \begin{bmatrix} 2 & -1 & 1 & -4 & 1 \\ 1 & 0 & 3 & 3 & 0 \\ -2 & 1 & -1 & 5 & 2 \\ 4 & -1 & 7 & 2 & 1 \end{bmatrix}$$

1. Verify that the product of A with each element of the basis B is $\mathbf{0}$.
2. Use `rref(A)` to find another basis for the null space.
3. The basis we found using the `null` function and the basis you found in part (b) look very different. How do we know that they span the same subspace?

Problem 30 Use some of our earlier matrices, together with their null spaces, for the following experiment.

- Given a matrix, find the product of the matrix with some vector in its null space. What do you observe? Why does this make sense?
- Formulate a conjecture about the relationship between the rows of a matrix and the elements of its null space.
- Prove your conjecture.

Bibliography

Matrices used in Problems ??-?? come from the end of Chapter 4 of Ken Kuttler's *A First Course in Linear Algebra*. (CC-BY)

Ken Kuttler, *A First Course in Linear Algebra*, Lyryx 2017, Open Edition, pp. 227–232.

9 Octave for Chapter 7

Octave for Chapter 7

Examples in this section provide sample Octave code for finding determinants and illustrating their uses and properties. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

Warning 2. In practice, computing determinants using Laplace Expansion is computationally expensive. *LU*-factorization can be used to improve efficiency, but may be numerically unstable. As a result, determinants are rarely used in computational practice. Examples and exercises in this section are designed to illustrate theoretical properties and computational pitfalls associated with determinants and should not be interpreted as templates for large-scale computational problems.

We will use the *det* function to find determinants in Octave.

Example 12. Compute the determinant of each of the following matrices. If a matrix is singular, say so.

$$A = \begin{bmatrix} -1 & 5 & -1 \\ 7 & -4 & -1 \\ 5 & 6 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 6 & -1 & 2 \\ 12 & -3 & 1 \\ -1 & 4 & 11 \end{bmatrix}$$

Explanation. We find the determinant of each matrix.

```
% Define matrices A and B
A=[-1  5  -1;
   7  -4  -1;
   5   6 -3];

B=[6  -1  2;
  12 -3  1;
  -1  4  11];

% Find det(A) and det(B)
det_A=det(A)
det_B=det(B)
```

[Link to code.](#)

We see that $\det(A) = 0$ and $\det(B) \neq 0$. We conclude that A is singular and B is not.

The following example illustrates a common technique for computing determinants numerically.

Example 13. In Exploration ??, we found that

$$A = \begin{bmatrix} 2 & -1 & 2 \\ 6 & 1 & 5 \\ -4 & 10 & -3 \end{bmatrix} = LU = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -2 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 2 \\ 0 & 4 & -1 \\ 0 & 0 & 3 \end{bmatrix}$$

Having a lower triangular/upper triangular factorization makes it easy to compute the determinant. Find the determinant of A by sight.

Explanation.

$$\det(A) = \det(LU) = \det(L)\det(U) = (1)(24) = 24$$

Unfortunately, LU -factorization is prone to round-off error, which may lead to erroneous or ambiguous outcomes, as you will see Problem ??.

Recall the following properties of determinants.

Theorem 2 (??). Let $A = [a_{ij}]$ be an $n \times n$ matrix.

1. If B is obtained from A by interchanging two different rows, then

$$\det B = -\det A$$

2. If B is obtained from A by multiplying one of the rows of A by a non-zero constant k . Then

$$\det B = k \det A$$

3. If B is obtained from A by adding a multiple of one row of A to another row, then

$$\det B = \det A$$

Example 14. Let

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

Illustrate part (b) of the above theorem using the second row of matrix A , and constant $k = 4$.

Explanation. We will use the following code to illustrate the property.

```
% Define rows of matrix A
r_1=[2 -1 5];
r_2=[0 3 -4];
r_3=[-2 1 -1];
```

```

% Combine the rows to form matrix A
A=[r_1; r_2; r_3];

% Compute the determinant of A
det_A=det(A)

% Form matrix B by multiplying the second row of A by k=4
B=[r_1; 4*r_2; r_3];

% Compute the determinant of B
det_B=det(B)

% Compute 4*det(A)
det_A_times4=4*det(A)

```

[Link to code.](#)

For our next example, recall the following theorem, known as Cramer's Rule.

Theorem 3 (??). Let A be a nonsingular $n \times n$ matrix, and let $\mathbf{b}$ be an $n \times 1$ vector. Then the components of the solution vector $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$ are given by

$$x_i = \frac{\det A_i(\mathbf{b})}{\det A}$$

where $A_i(\mathbf{b})$ is matrix A with its i^{th} column replaced by $\mathbf{b}$.

Example 15. Use Octave to solve the following system for x_1 using Cramer's rule.

$$\begin{bmatrix} 1 & 2 & -1 \\ -1 & 1 & 1 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \\ 1 \end{bmatrix}$$

Explanation. We will use the following code.

```

% Define columns of matrix A
c_1=[1; -1; 0];
c_2=[2; 1; 3];
c_3=[-1; 1; 1];

% Combine the columns to form matrix A
A=[c_1 c_2 c_3];

% Define vector b
b=[-2; 3; 1];

% Form A_1 by replacing the first column of A with b
A_1=[b c_2 c_3];

% Find x_1

```

$$x_1 = \det(A_1) / \det(A)$$

[Link to code.](#)

Octave Exercises

Problem 31 Let

$$A = \begin{bmatrix} 1 & -2 & 1 & 6 & 1 \\ 0 & -1 & 3 & 5 & -1 \\ 2 & -1 & 8 & 4 & 2 \\ 0 & 14 & -11 & -7 & 2 \\ 3 & -3 & 9 & 10 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -2 & 1 & 6 & 1 \\ 0 & -1 & 3 & 5 & 0 \\ 2 & -1 & 8 & 4 & 1 \\ 3 & 12 & -2 & 3 & 0 \\ 2 & 0 & -11 & -9 & 5 \end{bmatrix}$$

Determine whether each of the above matrices is singular in two different ways.

- Use the `det` function
- Use an alternative approach

Are the two methods used above in agreement? Click the arrow on the right to learn more.

Octave uses LU-decomposition to compute the determinant. This method is susceptible to rounding errors ([Reference](#)). Because of this, using the `det` function to determine singularity is generally not a good idea.

If A is non-singular, find A^{-1} , and $\det(A^{-1})$. Illustrate the relationship between $\det(A)$ and $\det(A^{-1})$ computationally.

Problem 32 Modify Example ?? to illustrate parts (a) and (c) of Theorem ?. (Use rows and coefficients of your choice.)

Problem 33 Modify the code of Example ?? to find x_2 and x_3 .

10 Octave for Chapter 8

Octave for Chapter 8

Examples in this section provide sample Octave code for computing eigenvalues. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

Template 10. Find eigenvalues and the corresponding eigenvectors of the given matrix A .

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

% Find eigenvalues of A
eigen_A=eig(A)

% Find eigenvalues and eigenvectors of A
[V,D]=eig(A)
% D is a diagonal matrix whose diagonal entries are the eigenvalues of A
% Columns of V are the eigenvectors of A
```

[Link to code.](#)

Template 11. We can find the characteristic polynomial of the given matrix A as follows.

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

% Find the coefficients of the char poly of A
char_poly = poly(A)
```

The output looks like this:

```
char_poly =  
1 -5 -1 5
```

[Link to code.](#)

The output is a vector of coefficients, starting with the leading coefficient. In this case, we see that the characteristic polynomial is $\lambda^3 - 5\lambda^2 - \lambda + 5$.

Warning 3. In this text, we find the characteristic polynomial by computing $\det(A - \lambda I)$. Octave, as well as some other textbooks, finds the characteristic polynomial based on $\det(\lambda I - A)$. Note that setting these expressions equal to zero to find eigenvalues produces the same result. However, the coefficients of the characteristic polynomials from the two approaches are negatives of each other. If you were to do this problem by hand, using the method in this text, you would get $-\lambda^3 + 5\lambda^2 + \lambda - 5$.

Example 16. In Example ??, you found that the rotation matrix $M = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$ has complex eigenvalues. Use the *eig* command in Octave to compute the eigenvalues and eigenvectors of M .

Explanation. Here is the input code:

```
% Define A  
A=[sqrt(2)/2  -sqrt(2)/2;  
  sqrt(2)/2  sqrt(2)/2];  
  
% Find eigenvalues of A  
eigen=eig(A)  
  
% Find eigenvalues and eigenvectors of A  
[V,D]=eig(A)  
% D is a diagonal matrix whose diagonal entries are the eigenvalues of A  
% Columns of V are the eigenvectors of A
```

This is what we get for the output:

```
eigen =  
  
(0.707107,0.707107)  
(0.707107,-0.707107)  
  
V =  
  
(0,0.707107) (0,-0.707107)
```

(0.707107,0) (0.707107,-0)

D =

(0.707107,0.707107) (0,0)
(0,0) (0.707107,-0.707107)

[Link to code.](#)

Each complex number in the output is presented using its real and imaginary parts as x and y coordinates of a point. For example, the second eigenvalue, written as $(0.707107, -0.707107)$, is the complex number $\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i$. Compare this to the eigenvalue in Example ???. The corresponding eigenvector is the second column of the matrix V : $\begin{bmatrix} (0, -0.707107) \\ (0.707107, -0) \end{bmatrix}$. This should be interpreted as $\begin{bmatrix} (-\sqrt{2}/2)i \\ \sqrt{2}/2 \end{bmatrix}$. Compare this to the eigenvector in Example ???. There is a discrepancy! Did we make a mistake? How do we reconcile the two eigenvectors?

Octave Exercises

Problem 34 Let $A = \begin{bmatrix} -148 & 16 & -528 \\ 71 & -28 & 292 \\ 39 & -4 & 140 \end{bmatrix}$.

1. Use Octave to find the eigenvalues of A .
2. Find the eigenvalues of $4A$. How do they compare?
3. Formulate and prove a conjecture about the relationship between the eigenvalues of A and the eigenvalues of kA , where k is a constant.
4. How do the eigenvectors of A compare to the eigenvectors of kA ? Prove your claim.

Problem 35 Let

$$A = \begin{bmatrix} 2 & -20 & 36 \\ -1 & 8 & -14 \\ 0 & 5 & -7 \end{bmatrix}$$

1. Find the eigenvalues of A using Octave. List the eigenvalues below in increasing order.

$$\lambda_1 = \dots, \quad \lambda_2 = \dots, \quad \lambda_3 = \dots$$

2. By hand, find the corresponding eigenvectors of A . List them below.

$$\mathbf{x}_{\lambda_1} = \begin{bmatrix} \dots \\ \dots \\ 1 \end{bmatrix}, \quad \mathbf{x}_{\lambda_2} = \begin{bmatrix} \dots \\ \dots \\ 5 \end{bmatrix}, \quad \mathbf{x}_{\lambda_3} = \begin{bmatrix} \dots \\ \dots \\ 1 \end{bmatrix}$$

3. Use Octave to find the eigenvectors of A . Reconcile your answers from the previous part with the answers you got from Octave.

Problem 36 Use Octave to find the eigenvalues of $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Interpret the numbers you are seeing. Interpret your results geometrically.

Problem 37 Let

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

When we use Octave to find the eigenvalues of A , we get the following printout.

```
ans =
    16.1168
   -1.11684
  -1.30368e-15
```

Should we interpret the last eigenvalue as 0, or as a very small non-zero number? How can you be sure of your answer in this particular case?

Problem 38 Modify the code for the Power Method given in [The Power Method and the Dominant Eigenvalue](#) to approximate the dominant eigenvalue of $A = \begin{bmatrix} -4 & 64 \\ 16 & 44 \end{bmatrix}$. Experiment with the number of iterations. Compare your answer to the answer you get when using the `eig` function.

11 Octave for Chapter 9

Octave for Chapter 9

Examples and Templates in this section provide sample Octave code for least squares, and QR -factorization. You can access our code through the link at the bottom of each template. Feel free to modify the code and experiment to learn more!

You can write your own code using Octave software or online Octave cells. To access Octave cells online, go to the [Sage Math Cell Webpage](#), select OCTAVE as the language, enter your code, and press EVALUATE.

To "save" or share your online code, click on the *Share* button, select *Permalink*, then copy the address directly from the browser window. You can store this link to access your work later or share this link with others. You will need to get a new Permalink every time you modify the code.

Octave Tutorial

Least Squares

The following theorem establishes one way to implement least squares.

Theorem 4 (??). Let A be an $m \times n$ matrix, let $\mathbf{b}$ be a column vector in $\mathbb{R}^m$. Consider the matrix equation

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

1. Any solution $\mathbf{z}$ to the normal equations

$$(A^T A)\mathbf{z} = A^T \mathbf{b}$$

is a best approximation to a solution to $A\mathbf{x} = \mathbf{b}$ in the sense that $\|\mathbf{b} - A\mathbf{z}\|$ is minimized.

2. If the columns of A are linearly independent, then $A^T A$ is invertible and $\mathbf{z}$ is given uniquely by

$$\mathbf{z} = (A^T A)^{-1} A^T \mathbf{b}$$

We can implement this process as follows.

Template 12. We show how to find a least squares solution to $A\mathbf{x} = \mathbf{b}$.

```
% Define matrix A
A=[9 -3 3;
  1 -1 1;
  4 0 1;
  1 1 1;
  9 -1 2];

% Define vector b
```

```
b=[3;1;1;2;4];

% Now we solve for z
z=inv(transpose(A)*A)*transpose(A)*b
```

[Link to code.](#)

Warning 4. Finding matrix inverses is both computationally expensive and unstable.

To do least squares in Octave, we use the backslash operator (`\`). Depending on the type of matrix, backslash uses different techniques to compute the least squares solution more efficiently. For more information about *inv* and the backslash operator (`\`) see [Reference](#).

The following two examples illustrate the two implementations of least squares side-by-side.

Example 17. We will compare the results of the two implementations of least squares for a 7×3 coefficient matrix.

```
% Define matrix A
A=[4 -1 0;
   2 -1 1;
   6 3 1;
  -1 1 2;
   3 1 -2;
  -2 8 7;
   1 1 -5];

% Define vector b
b=[-2;4;1;2;-5;1;-3];

% Implement least squares using our original method
z1=inv(transpose(A)*A)*transpose(A)*b

% Implement least squares using the backslash operator
z2=A\b
```

[Link to code.](#)

Example 18. We will compare the performance of two the implementations of least squares for a 2000×20 coefficient matrix.

```
% Define matrix A
A=rand(2000,20); % rand(m,n) creates an m by n matrix whose entries are between 0 and 1

% Define vector b
b=rand(2000,1);

% Start the timer
```

```
tic

% Implement least squares using our original method
z1=inv(transpose(A)*A)*transpose(A)*b;

% Stop the timer
timeINV=toc

% Start the timer
tic

% Implement least squares using the backslash operator
z2=A\b;

% Stop the timer
timeBSlash=toc
```

Link to code.

Each time you run the algorithm, you will get slightly different run times for each implementation. Most of the time, you will see that the backslash operator outperforms our initial method.

The backslash operator (`\`) can be used to solve systems of equations when a solution exists. Here is an example.

Example 19. Consider the system

$$\begin{array}{rclcrcl} x & - & y & & & = & 0 \\ 2x & - & 2y & + & z & + & 2w & = & 4 \\ & & y & & & + & w & = & 0 \\ & & & & 2z & + & w & = & 5 \end{array} \quad (5)$$

(You may remember this system from [Augmented Matrix Notation and Elementary Row Operations](#).) We will solve this system using the backslash operator (`\`).

```
% Define the coefficient matrix A
```

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 2 & -2 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix};$$

```
% Define vector b
```

```
b = [0;4;0;5];
```

```
% Solve the system
```

$$x = A \setminus b$$

```
% Compare the solution above to the solution obtained
```

```
% using rref
A_b=[A b];
rref(A_b)
```

[Link to code.](#)

Warning 5. While the backslash (\) operator can be used to solve feasible systems, we urge you to be very cautious because

- If a system has infinitely many solutions, the backslash (\) operator will find one particular solution.
- If a system has no solutions, the backslash (\) operator will find an approximate solution using least-squares.

The way the answer is presented, you cannot tell the difference!

QR-factorization

Recall the definition of *QR*-factorization.

Definition 3 (??). Let A be an $m \times n$ matrix with independent columns. A *QR-factorization* of A expresses it as $A = QR$ where Q is $m \times n$ with orthonormal columns and R is an invertible and upper triangular matrix with positive diagonal entries.

Example 20. Find the *QR*-factorization of

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

and demonstrate that matrix Q is orthogonal (columns are orthonormal).

Explanation. % Define A

```
A=[2 -1 4 0;
    3 1 1 -2;
    1 5 -1 6;
    -3 4 0 7];
```

```
% Find the QR-factorization of A
[Q,R]=qr(A)
```

```
% Verify that columns of Q are orthonormal
% You may get numbers that are close to zero but not zero
% due to round-off error
transpose(Q)*Q
```

```
% why does this verify that the columns are orthonormal?
```

[Link to code.](#)

Question 39 In the code above, we used the product $Q^T Q$ to verify that the columns of Q are orthonormal. What do you expect this product to look like if the columns are orthonormal? What will the product look like if they are not? You will be asked to explain your reasoning in Problem ??.

Recall the following algorithm for using QR -factorization to approximate eigenvalues from [QR-Factorization](#).

Algorithm 1 (??). Let A be an invertible matrix.

Step 1: Define $A_1 = A$ and factor it as $A_1 = Q_1 R_1$.

Step 2: Define $A_2 = R_1 Q_1$ and factor it as $A_2 = Q_2 R_2$.

Step 3: Define $A_3 = R_2 Q_2$ and factor it as $A_3 = Q_3 R_3$.

$\vdots$

Step $k + 1$: Define $A_{k+1} = R_k Q_k$ (where $A_k = Q_k R_k$).

Note that A_{k+1} is similar to A_k (in fact, $A_{k+1} = R_k Q_k = (Q_k^{-1} A_k) Q_k$), and hence each A_k has the same eigenvalues as A . If the eigenvalues of A are real and have distinct absolute values, the sequence of matrices $A_1, A_2, A_3, \dots$ converges to an upper triangular matrix with these eigenvalues on the main diagonal.

Compare the following example to Example ??.

Example 21. Use the QR algorithm to approximate the eigenvalues of $A = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$.

Explanation. Here is a very simple implementation of the algorithm.

```
% Define A
A=[1 1;
   2 0];

% Set the desired number of iterations
iter=20;

% Implement the algorithm. Note that we are over-writing the previous A with every iteration.
for i=1:iter
    [Q,R]=qr(A);
    A=R*Q
end
```

[Link to code.](#)

Observe that the main diagonal entries are approaching the eigenvalues of A .

Octave Exercises

Problem 40 Attempt to solve each of the following systems of equations using the backslash ($\backslash$) operator, and the `rref` function. Interpret your results.

GeoGebra link: <https://www.geogebra.org/m/kmt5ukhc>

1. Find the residual, $\|\mathbf{b} - \mathbf{Az}\|$, for your line of best fit.
2. The original line, $y = x + 1$ initially shown in the interactive, visually appeared to be a pretty good fit. Find the residual for the line $y = x + 1$ and compare it to the residual you got for your line. Is your line a better fit?

Problem 42 Explain how to interpret the product $\mathbf{Q}^T \mathbf{Q}$ to determine whether matrix $\mathbf{Q}$ has orthonormal columns. Prove your claims.

Problem 43 Modify the code in Example ?? to approximate the eigenvalues of $\mathbf{A} = \begin{bmatrix} 9 & 2 & 8 \\ 2 & -6 & -2 \\ -8 & 2 & -5 \end{bmatrix}$. Compare your answers to the answers you got for Problem ??.

Problem 44 Modify the code in Example ?? for the matrix $\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. What are you observing? Are we getting close to finding the eigenvalues of $\mathbf{A}$? Explain what is happening and why.