
1 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 1. 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 2. 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 1. 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 2. 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 3. 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 1. • 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 1 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 $\mathbf{Ax} = \mathbf{b}$ 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 2 Let $A = \begin{bmatrix} 0 & 1 \\ 5 & 3 \end{bmatrix}$. Find A^{-1} in two different ways. First, use the `inv` function. Second, find `rref[A|I]`.

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

Problem 4 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](#):

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

Problem 5 Write a routine that multiplies an $m \times n$ matrix by an $n \times k$ matrix.

Problem 6 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 7 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