4.4.4 Special Shapes

4.4.4 Special Shapes

noticed is that when we use NumPy

we tend to make everything a matrix.

Often, the scalar we just create as a 1 by 1 matrix,

a vector we create as a matrix with one column, a row

vector we create as a matrix with one row, and so forth.

What we're going to do now is we're going

to look at matrix-matrix multiplication and we're

going to see that when we're working with one of those special shapes,

exactly the same operation comes out as we have seen before.

Let's have a look.

Remember that we're always looking at a matrix-matrix multiply that

0:00 / 7:52 1.0x

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Discussion

Hide Discussion

Topic: Week 4 / 4.4.4

Add a Post

Show all posts ▾ by recent activity ▾

There are no posts in this topic yet.

✖

In this section in particular, it really helps to read the "Related Reading" as you do the homework problems.

Homework 4.4.4.1

1/1 point (graded)

Let $A = \begin{pmatrix} 4 \end{pmatrix}$ and $B = \begin{pmatrix} 3 \end{pmatrix}$. Then $AB =$___.

12

12

Submit

Homework 4.4.4.2

1/1 point (graded)

Let $A = \begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix}$ and $B = \begin{pmatrix} 4 \end{pmatrix}$. Then $AB =$___.

- [ ] $\begin{pmatrix} 2 \\ -6 \\ 4 \end{pmatrix}$
- [x] $\begin{pmatrix} 4 \\ -12 \\ 8 \end{pmatrix}$
- [ ] $\begin{pmatrix} 4 \\ 9 \\ -12 \end{pmatrix}$

Submit

Homework 4.4.4.3

1/1 point (graded)

This problem talks about IPython Notebooks and Python. It points out an interesting problem with the `numpy` package, which one can use with Python to do matrix computations. In MATLAB, the described behavior is not observed, so we can't create an equivalent homework for MATLAB. We left the problem here, because it points out interesting behavior when one considers a scalar to be a 1×1 matrix.

Start up a new IPython Notebook and try this:

import numpy as np

x = np.matrix('1;2;3')

print(x)

alpha = np.matrix('-2')

print(alpha)

print(x * alpha)

Notice how x, alpha, and x * alpha are created as matrices. Now try

print(alpha * x)

This causes an error! Why? Because numpy checks the sizes of matrices alpha and x and deduces that they don't match. Hence the operation is illegal. This is an artifact of how numpy is implemented.

Now, for us a 1 X 1 matrix and a scalar are one and the same thing, and that therefore $\alpha x = x \alpha$.

Indeed, our laff.scal routine does just fine:

import laff

laff.scal(alpha, x)

print(x)

yields the desired result. This means that you can use the laff.scal routine for both update $x := \alpha x$ and $x := x\alpha$.

☑ Done/Skip

Submit

Homework 4.4.4.4

1/1 point (graded)

Let $A = \begin{pmatrix} 4 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & -3 & 2 \end{pmatrix}$. Then $AB =$___.

- ○ $\begin{pmatrix} 2 & -6 & 4 \end{pmatrix}$
- ◉ $\begin{pmatrix} 4 & -12 & 8 \end{pmatrix}$
- ○ $\begin{pmatrix} 4 & 9 & -12 \end{pmatrix}$

Submit

4.4.4.5

1/1 point (graded)

Like Homework 4.4.4.3, this problem talks about IPython Notebooks and Python. It points out an interesting problem with the `numpy` package, which one can use with Python to do matrix computations. In MATLAB, the described behavior is not observed, so we can't create an equivalent homework for MATLAB. We left the problem here, because it points out interesting behavior when one considers a scalar to be a 1×1 matrix.

Start up a new IPython Notebook and try this:

import numpy as np

xt = np.matrix('1,2,3')

print(xt)

alpha = np.matrix('-2')

print(alpha)

print(xt * alpha)

This causes an error! Why? Because numpy checks the sizes of matrices alpha and xt and deduces that they don't match. Hence the operation is illegal. This is an artifact of how numpy is implemented.

Now try

print(alpha * xt)

Now, for us a 1 X 1 matrix and a scalar are one and the same thing, and that therefore $\alpha x^T = x^T \alpha$.

Indeed, our laff.scal routine does just fine:

import laff

laff.scal(alpha, xt)

print(xt)

yields the desired result. This means that you can use the laff.scal routine for both update $x^T := \alpha x^T$ and $x^T := x^T \alpha$.

- [x] Done/Skip

Submit

Homework 4.4.4.6

1/1 point (graded)

Let $A = \begin{pmatrix} 1 & -3 & 2 \end{pmatrix}$ and $B = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$. Then $AB =$___.

5

5

Submit

Homework 4.4.4.7

1/1 point (graded)

Try this in MATLAB:

```
>> xt = [ 1 2 3 ]

>> y = [
-1
0
2
]

>> xt * y

>> laff_dot( xt, y )
```

The point is that

- `xt` can be thought of as a 1 x 3 matrix or a row vector.
- `y` can be thought of as a 3 x 1 matrix or a column vector.
- `xt * y` (matrix-matrix multiplication) computes the same as `laff_dot( xt, y )`

We prefer using our `laff_dot` and `laff_dots` routines, which don't care about whether x and y are rows or columns, making the adjustment automatically. This is in part because it explicitly tells us we are performance a dot product of two vectors, because of the names of the routines. In addition, when we use these routines in a code that uses the `FLAME@lab` API, we can use PictureFLAME to visualize the algorithm executing.

☑ Done/Skip

Submit

Homework 4.4.4.8

6/6 points (graded)

Let $A = \begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix}$ and $B = \begin{pmatrix} -1 & -2 \end{pmatrix}$. Then $AB =$

-1 ✔	-2 ✔
3 ✔	6 ✔
-2 ✔	-4 ✔

Submit

Homework 4.4.4.9

7/7 points (graded)

Let $a = \begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix}$ and $b^T = \begin{pmatrix} -1 & -2 \end{pmatrix}$ and $C = ab^T$. Partiton C by columns and by rows:

$$C = \left(\begin{array}{c|c} c_0 & c_1 \end{array} \right) \text{ and } C = \begin{pmatrix} \tilde{c}_0^T \\ \tilde{c}_1^T \\ \tilde{c}_2^T \end{pmatrix}$$

then

- $c_0 = (-1) \begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix} = \begin{pmatrix} (-1) \times (1) \\ (-1) \times (-3) \\ (-1) \times (2) \end{pmatrix}$

TRUE ✔

- $c_1 = (-2)\begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix} = \begin{pmatrix} (-2)\times(1) \\ (-2)\times(-3) \\ (-2)\times(2) \end{pmatrix}$

TRUE ✔

- $C = \left(\begin{array}{r|r} (-1)\times(1) & (-2)\times(1) \\ (-1)\times(-3) & (-2)\times(-3) \\ (-1)\times(2) & (-2)\times(2) \end{array}\right)$

TRUE ✔

- $\tilde{c}_0^T = (1)\begin{pmatrix} -1 & -2 \end{pmatrix} = \begin{pmatrix} (1)\times(-1) & (1)\times(-2) \end{pmatrix}$

TRUE ✔

- $\tilde{c}_1^T = (-3)\begin{pmatrix} -1 & -2 \end{pmatrix} = \begin{pmatrix} (-3)\times(-1) & (-3)\times(-2) \end{pmatrix}$

TRUE ✔

- $\tilde{c}_2^T = (2)\begin{pmatrix} -1 & -2 \end{pmatrix} = \begin{pmatrix} (2)\times(-1) & (2)\times(-2) \end{pmatrix}$

TRUE ✔

- $C = \left(\begin{array}{rr} (-1)\times(1) & (-2)\times(1) \\ \hline (-1)\times(-3) & (-2)\times(-3) \\ \hline (-1)\times(2) & (-2)\times(2) \end{array}\right)$

TRUE ✔

Submit

Homework 4.4.4.10

16/16 points (graded)

Consider

$$\begin{pmatrix} \square \\ \square \\ \square \\ \square \end{pmatrix} \begin{pmatrix} 2 & -1 & 3 \end{pmatrix} = \begin{pmatrix} 4 & \square & \square \\ -2 & \square & \square \\ 2 & \square & \square \\ 6 & \square & \square \end{pmatrix}$$

Fill in the boxes:

2 ✔

-1 ✔

1 ✔

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

4 ✔	-2 ✔	6 ✔
-2 ✔	1 ✔	-3 ✔
2 ✔	-1 ✔	3 ✔
6 ✔	-3 ✔	9 ✔

Submit

Homework 4.4.4.11

15/15 points (graded)

Consider

$$\begin{pmatrix} 2 \\ -1 \\ 1 \\ 3 \end{pmatrix} \begin{pmatrix} \square & \square & \square \end{pmatrix} = \begin{pmatrix} 4 & -2 & 6 \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{pmatrix}$$

Fill in the boxes:

$$\begin{pmatrix} 2 \\ -1 \\ 1 \\ 3 \end{pmatrix}$$

2

-1

3

$=$

4		
-2	-2	6
2	1	-3
6	-1	3
	-3	9

Submit

Homework 4.4.4.12

3/3 points (graded)

Let $A = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & -2 & 2 \\ 4 & 2 & 0 \\ 1 & 2 & 3 \end{pmatrix}$. Then $AB =$

4	2	0

Submit

Homework 4.4.4.13

1/1 point (graded)

Let $e_i \in \mathbb{R}^m$ equal the ith unit basis vector and $A \in \mathbb{R}^{m \times n}$. Then $e_i^T A = \tilde{a}_i^T$, the ith row of A.

Always

Submit

Homework 4.4.4.14

1/1 point (graded)

If you don't find the file `PracticeGemm.m` in `LAFF-2.0xM/Programming/Week04`, then

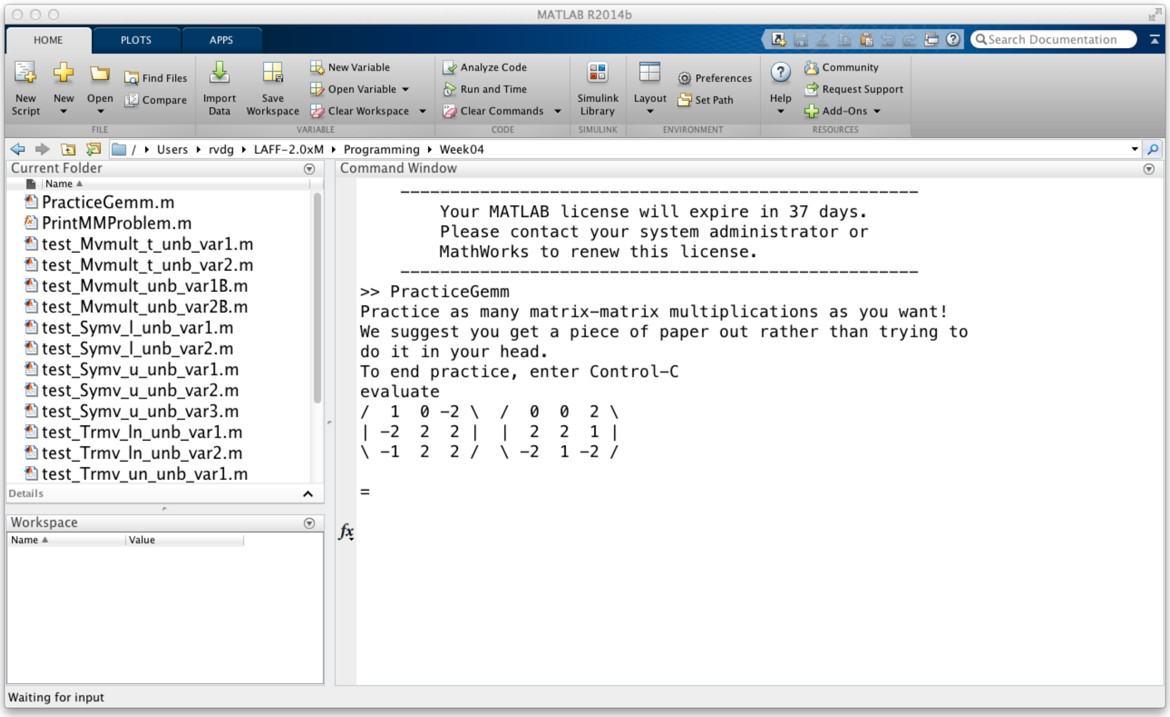
- Download the file `PracticeGemm.zip`.
- Unzip the file

There is a problem with the script in PracticeGemm.m when used with Matlab Online. Please download PracticeGemm.m and place in `LAFF-2.0xM/Programming/Week04`.

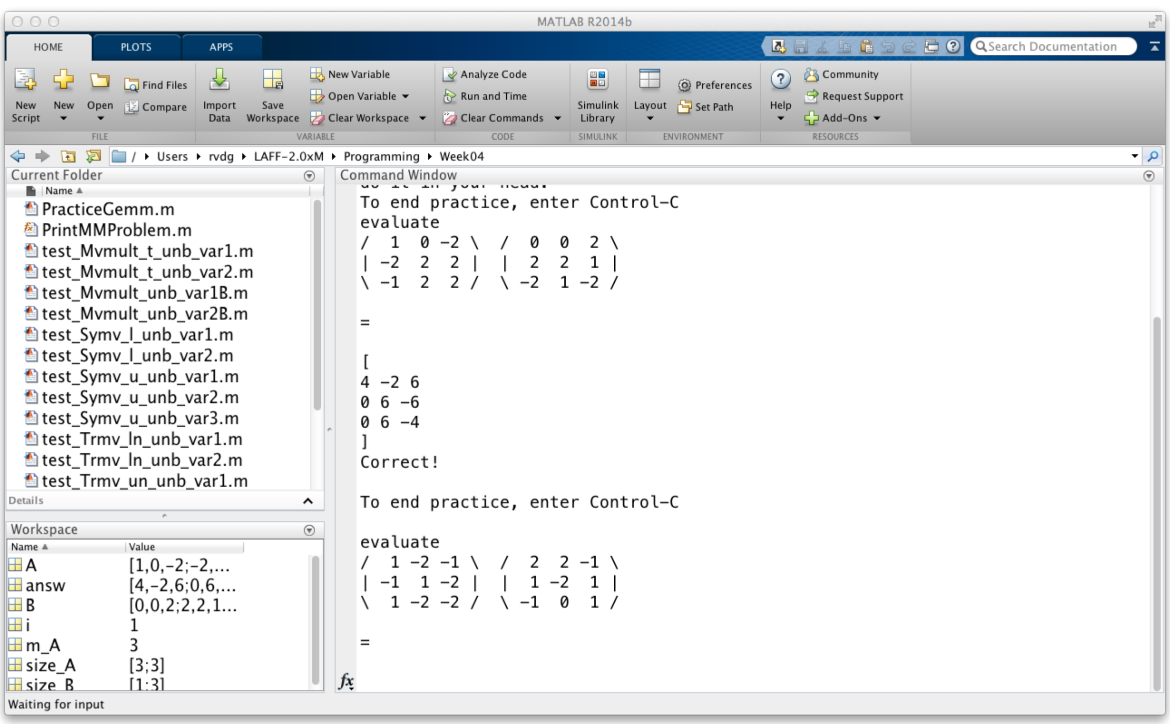
Next,

- Start MATLAB and make the directory in which the files `PracticeGemm.m` and `PrintMMProblem.m` exist the current directory for the Command Window.
- Execute `PracticeGemm`

  You will see something like



- Type in the answer:



(Notice that you enter it as you would enter a matrix in MATLAB.)

- Practice all you want!

☑ done/skip

✔

Submit