Arrays

MATLAB was originally written to ease dealing with tools of linear algebra – vectors and matrices.

Array - is a multi dimensional grid of data.

Single number: is a 1 x 1 array.

Column vector: a m x 1 array.

$$\boldsymbol{x} = \underset{\xleftrightarrow{\;l\;}}{\begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \end{pmatrix}} \Bigg\updownarrow m$$

Row vector: a 1 x n array.

$$\mathbf{y} = \underset{\xleftrightarrow{\qquad n \qquad}}{(y_1\ \ y_2\ \ y_3\ \ y_4\ \ y_5)} \updownarrow l$$

Matrix: a m x n array.

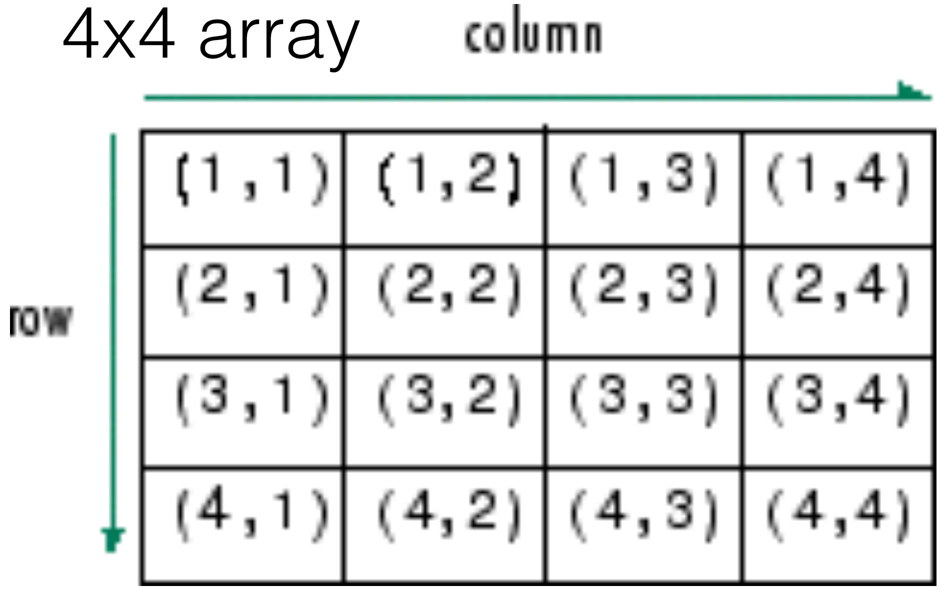
$$\boldsymbol{A} = \underset{\xleftrightarrow{\qquad\qquad n \qquad\qquad}}{\begin{pmatrix} -3 & 0 & 2 & 3 & -3 \\ -5 & -1 & 3 & 0 & -3 \\ 2 & 3 & -5 & 2 & 2 \\ -1 & 0 & 2 & -1 & -2 \\ 4 & -3 & -1 & -2 & 0 \end{pmatrix}} \Bigg\updownarrow m$$

MATLAB stores data all in arrays

So working with arrays is fundamental to working with MATLAB

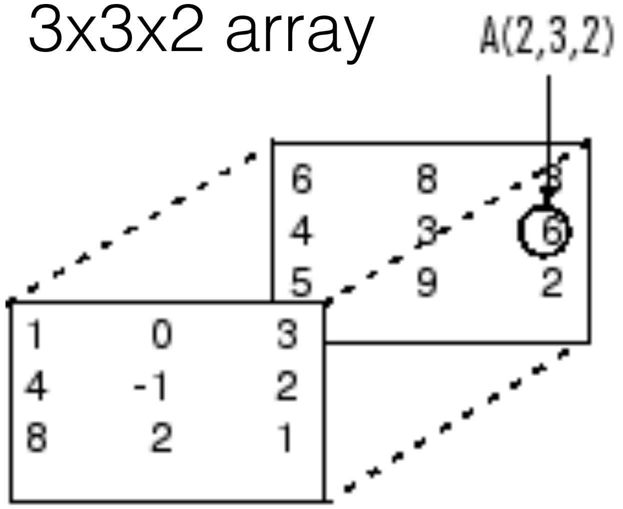
Arrays

4x4 array



(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

3x3x2 array



6	8	3
4	3	6
5	9	2

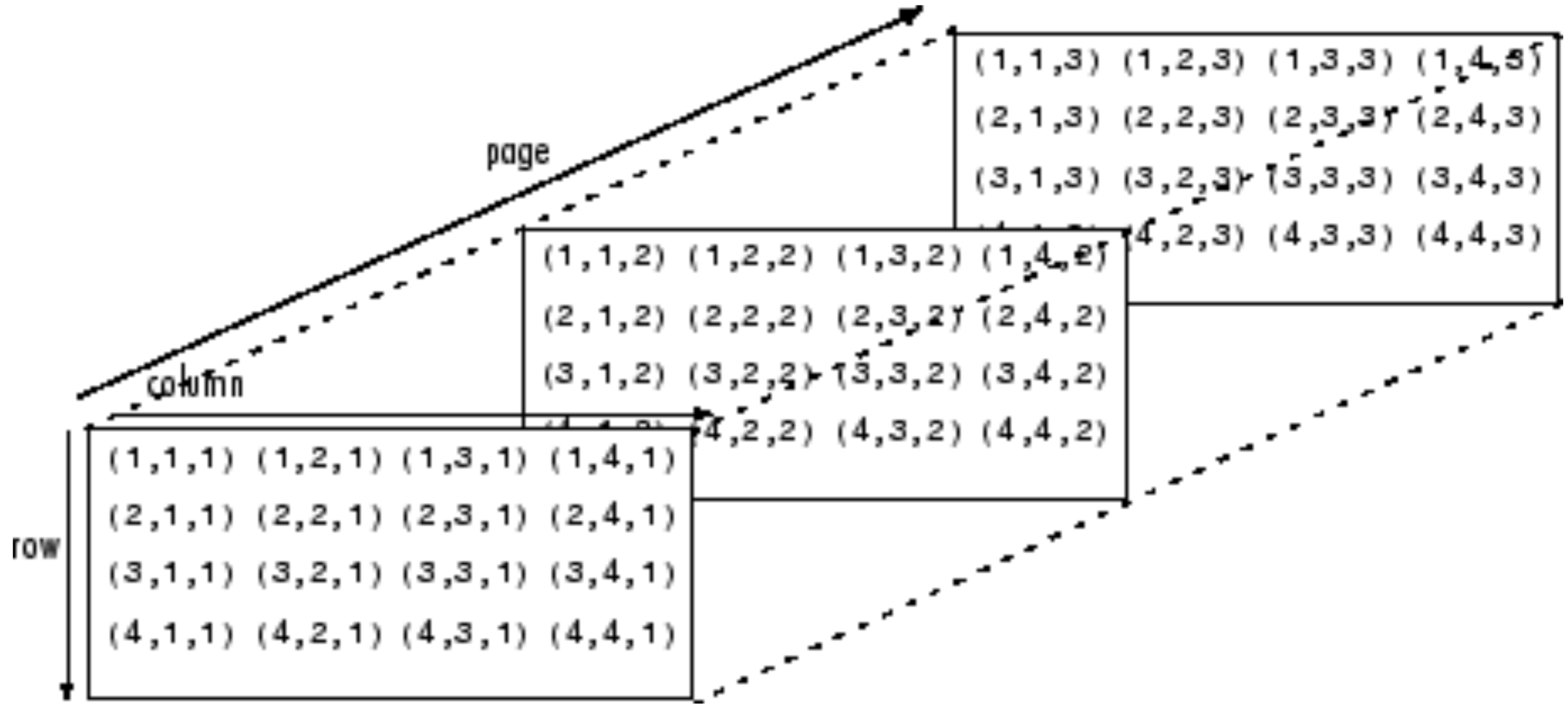
1	0	3
4	-1	2
8	2	1

```
A(:,:,1) =

     1     0     3
     4    -1     2
     8     2     1


A(:,:,2) =

     6     8     3
     4     3     6
     5     9     2
```



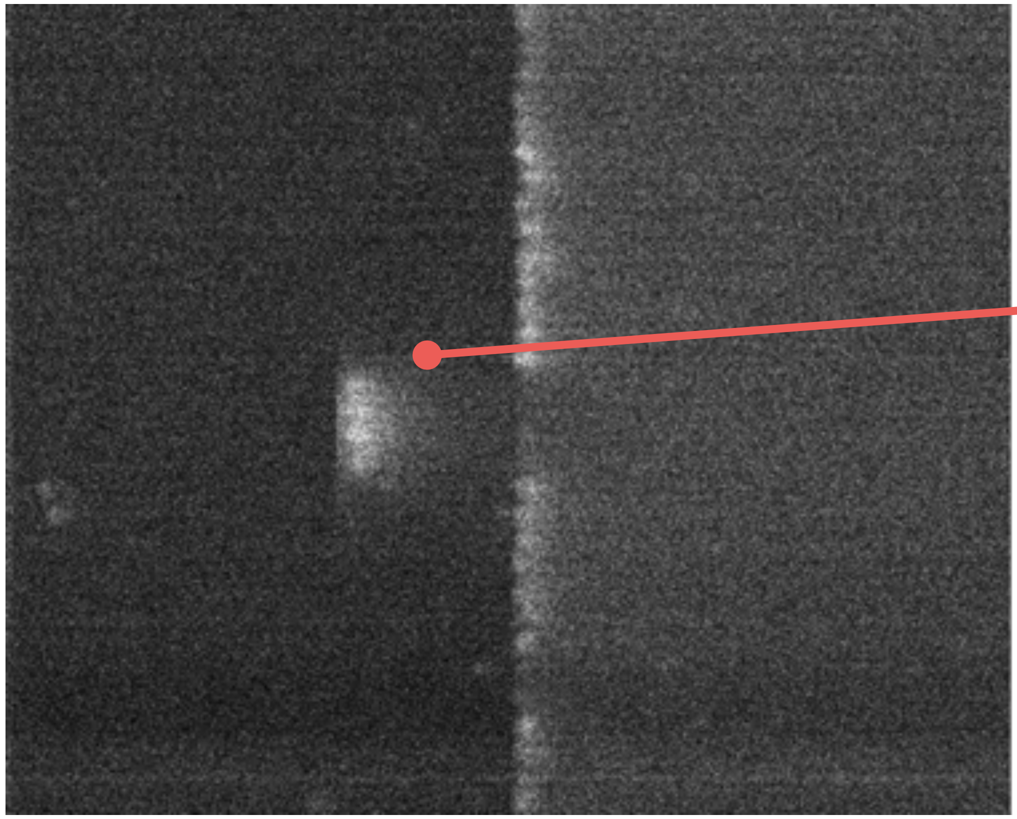
(1,1,3)	(1,2,3)	(1,3,3)	(1,4,3)
(2,1,3)	(2,2,3)	(2,3,3)	(2,4,3)
(3,1,3)	(3,2,3)	(3,3,3)	(3,4,3)
[illegible]	(4,2,3)	(4,3,3)	(4,4,3)

(1,1,2)	(1,2,2)	(1,3,2)	(1,4,2)
(2,1,2)	(2,2,2)	(2,3,2)	(2,4,2)
(3,1,2)	(3,2,2)	(3,3,2)	(3,4,2)
[illegible]	(4,2,2)	(4,3,2)	(4,4,2)

(1,1,1)	(1,2,1)	(1,3,1)	(1,4,1)
(2,1,1)	(2,2,1)	(2,3,1)	(2,4,1)
(3,1,1)	(3,2,1)	(3,3,1)	(3,4,1)
(4,1,1)	(4,2,1)	(4,3,1)	(4,4,1)

Arrays - Example 1
Image storage

Image: set of data that is real-valued, ordered, represents color and intensity



73	61	55	27	137	112	121	197	239
9	25	131	55	124	147	173	133	135
32	42	86	76	144	68	143	94	178
26	76	57	78	91	87	51	176	148
42	60	95	90	95	36	150	158	122
38	26	84	65	51	49	106	66	119
32	48	78	28	24	19	94	127	62
61	128	88	92	55	99	110	126	127
55	70	57	63	59	101	118	90	88
54	26	38	31	67	78	31	127	107
67	57	81	70	83	142	143	99	88
103	48	78	119	61	55	120	139	201

Arrays - Example 2
Stoichiometry matrix

E + S → ES

k_1*E*S

E--; S--; ES++

ES → E + S

k_2*ES

ES--; E++;S++

ES → E + P

k_3*ES

ES--; E++; P++

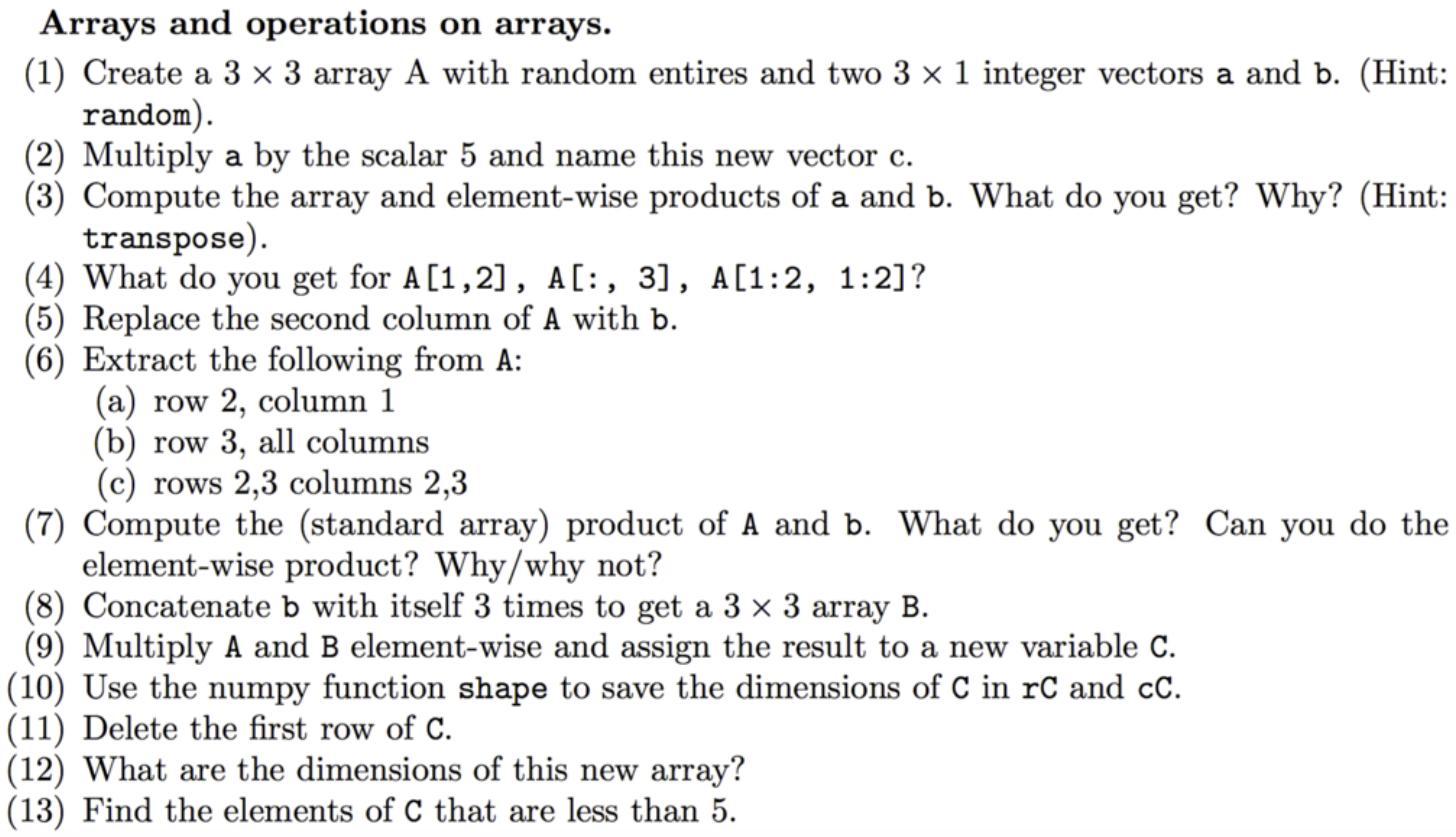
	E	S	ES	P
E + S → ES	-1	-1	1	0
ES → E + S	1	1	-1	0
ES → E + P	1	0	-1	1

Arrays - Other examples

- Can you think of other examples?

Operations on Arrays

Exercises



Arrays and operations on arrays.

(1) Create a 3 × 3 array A with random entires and two 3 × 1 integer vectors `a` and `b`. (Hint: `random`).

(2) Multiply `a` by the scalar 5 and name this new vector c.

(3) Compute the array and element-wise products of `a` and `b`. What do you get? Why? (Hint: `transpose`).

(4) What do you get for `A[1,2]`, `A[:, 3]`, `A[1:2, 1:2]`?

(5) Replace the second column of `A` with `b`.

(6) Extract the following from `A`:

  (a) row 2, column 1

  (b) row 3, all columns

  (c) rows 2,3 columns 2,3

(7) Compute the (standard array) product of `A` and `b`. What do you get? Can you do the element-wise product? Why/why not?

(8) Concatenate `b` with itself 3 times to get a 3 × 3 array `B`.

(9) Multiply `A` and `B` element-wise and assign the result to a new variable `C`.

(10) Use the numpy function `shape` to save the dimensions of `C` in `rC` and `cC`.

(11) Delete the first row of `C`.

(12) What are the dimensions of this new array?

(13) Find the elements of `C` that are less than 5.