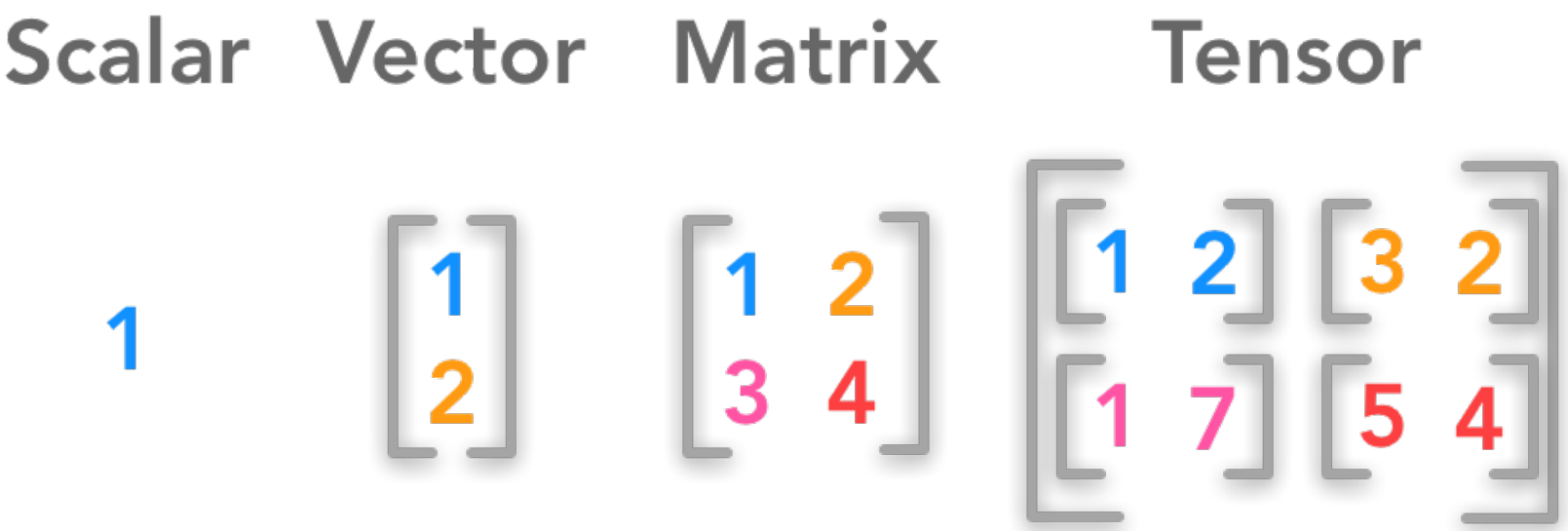
Linear Algebra with Numpy

David Arroyo Menéndez

October 11, 2019

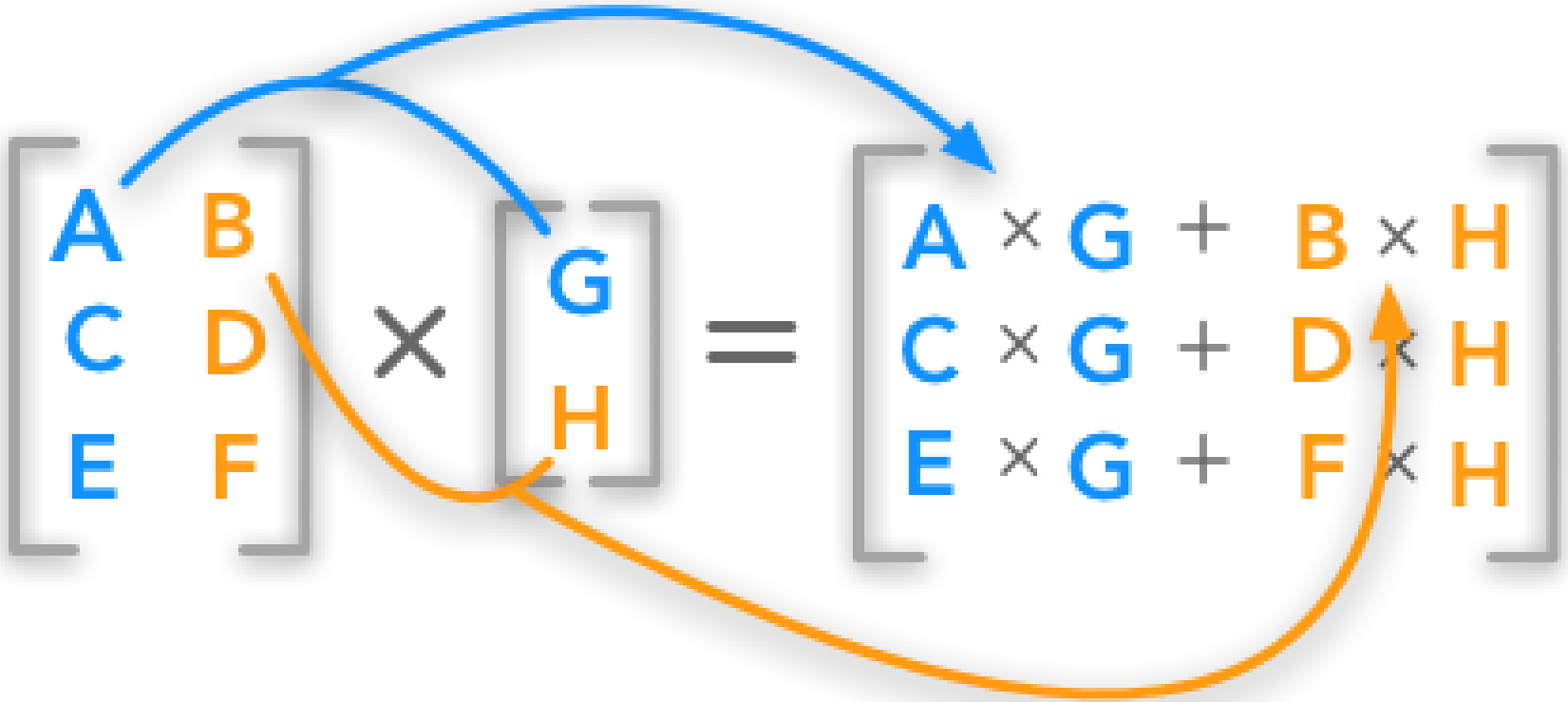
Scalars, Vectors, Matrices and Tensors (I)



Scalars, Vectors, Matrices and Tensors (II)

```
x = 1 # scalar
x = [1, 2] # vector
x = [[1, 2], [3, 4]] # matrix
x = [[[1, 2], [3, 2]], [[1, 7], [5, 4]]] # tensor
```

Multiplying Matrices and Vectors (I)



Multiplying Matrices and Vectors (II)

```
x = np.array([[3, 6, 7], [5, -3, 0]])
y = np.array([[1, 1], [2, 1], [3, -3]])
z = x.dot(y)
```

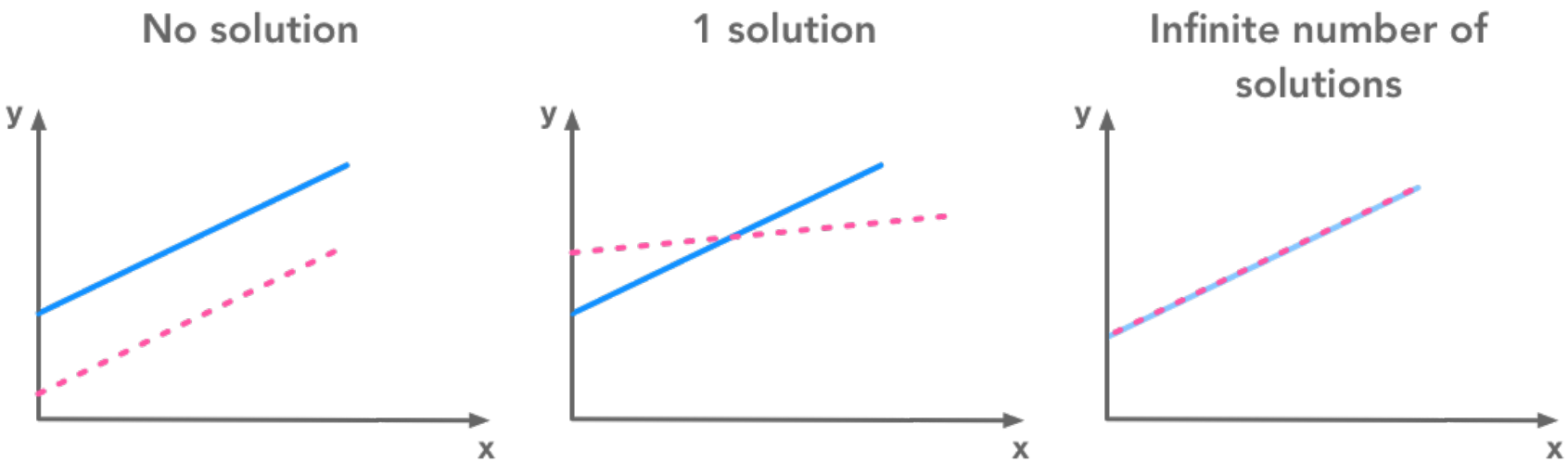
Identity and Inverse Matrices (I)

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Identity and Inverse Matrices (II)

```
np.identity(3)
```

Linear Dependence



Linear Algebra

- Rank, determinant, trace, etc. of an array.
- Eigen values of matrices
- Matrix and vector products (dot, inner, outer,etc. product), matrix exponentiation
- Solve linear or tensor equations

Linear Algebra (II)

```
# Importing numpy as np
import numpy as np

A = np.array([[6, 1, 1],
   [4, -2, 5],
   [2, 8, 7]])

# Rank of a matrix
print("Rank of A:", np.linalg.matrix_rank(A))

# Trace of matrix A
print("\nTrace of A:", np.trace(A))

# Determinant of a matrix
print("\nDeterminant of A:", np.linalg.det(A))

# Inverse of matrix A
```

Linear Algebra (III). Matrix eigenvalues functions

```
from numpy import linalg as geek

# Creating an array using array
# function
a = np.array([[1, -2j], [2j, 5]])

print("Array is :",a)

# calculating an eigen value
# using eigh() function
c, d = geek.eigh(a)

print("Eigen value is :", c)
print("Eigen value is :", d)
```

Linear Algebra (IV). Eigen value

```
from numpy import linalg as geek

# Creating an array using diag
# function
a = np.diag((1, 2, 3))

print("Array is :",a)

# calculating an eigen value
# using eig() function
c, d = geek.eig(a)

print("Eigen value is :",c)
print("Eigen value is :",d)
```

Linear Algebra (V)

```
import numpy as geek

# Scalars
product = geek.dot(5, 4)
print("Dot Product of scalar values  : ", product)

# 1D array
vector_a = 2 + 3j
vector_b = 4 + 5j

product = geek.dot(vector_a, vector_b)
print("Dot Product  : ", product)
```

Linear Algebra (VI). Solve

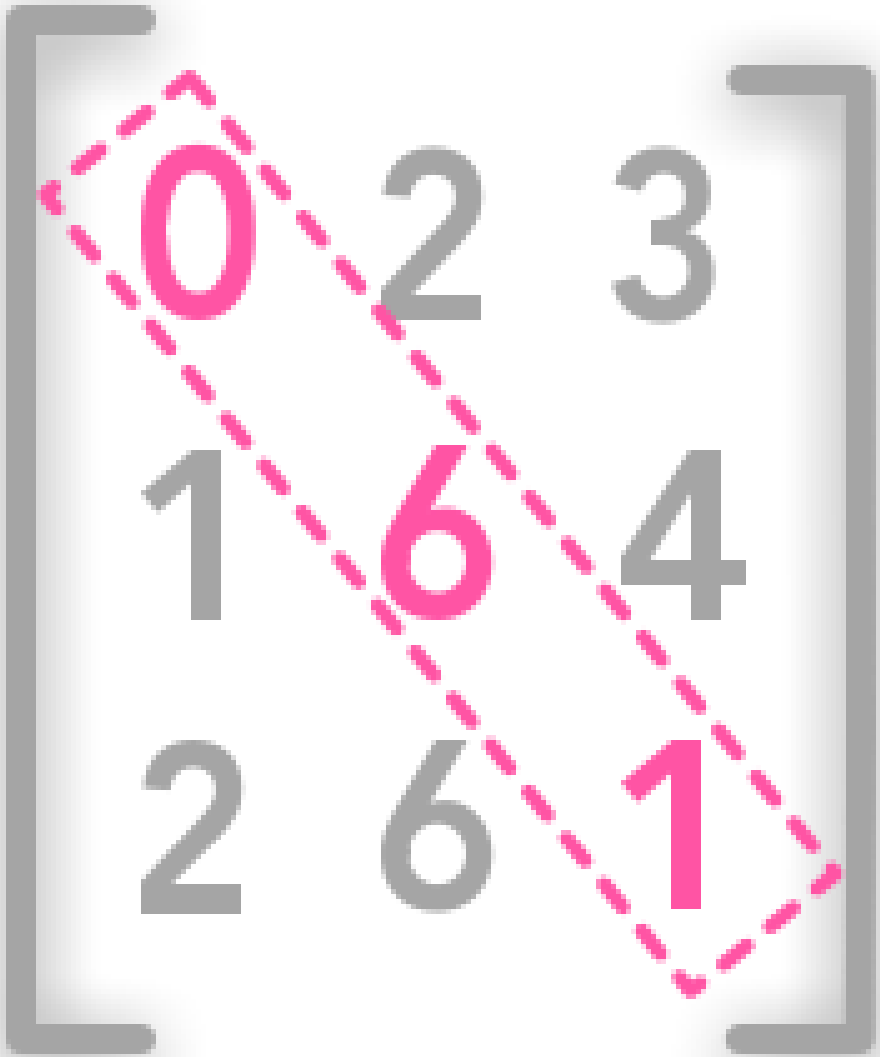
```
import numpy as np

# Creating an array using array
# function
a = np.array([[1, 2], [3, 4]])

# Creating an array using array
# function
b = np.array([8, 18])

print(("Solution of linear equations:",
  np.linalg.solve(a, b)))
```

Trace (I)



Trace (II)

```
import numpy as np

np.trace(np.eye(3))
a = np.arange(8).reshape((2,2,2))
np.trace(a)
```