反矩陣的應用

給兩點求一直線

給不共線三點求一平面

Matrix

```
A = np.matrix([range(1,5)])
```

```
matrix([[1, 2, 3, 4]])
```

```python
import numpy as np
from numpy.linalg import inv
```

```python
a = np.matrix([1,2,3,4])
A = np.reshape(a,(2,2))
inv(A)
```

可使用
array

```
>>> a
matrix([[1, 2, 3, 4]])
```

```
>>> A
matrix([[1, 2],
        [3, 4]])
```

```
>>> inv(A)
matrix([[-2. ,  1. ],
        [ 1.5, -0.5]])
```

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$$

如何驗證？

$$A * A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

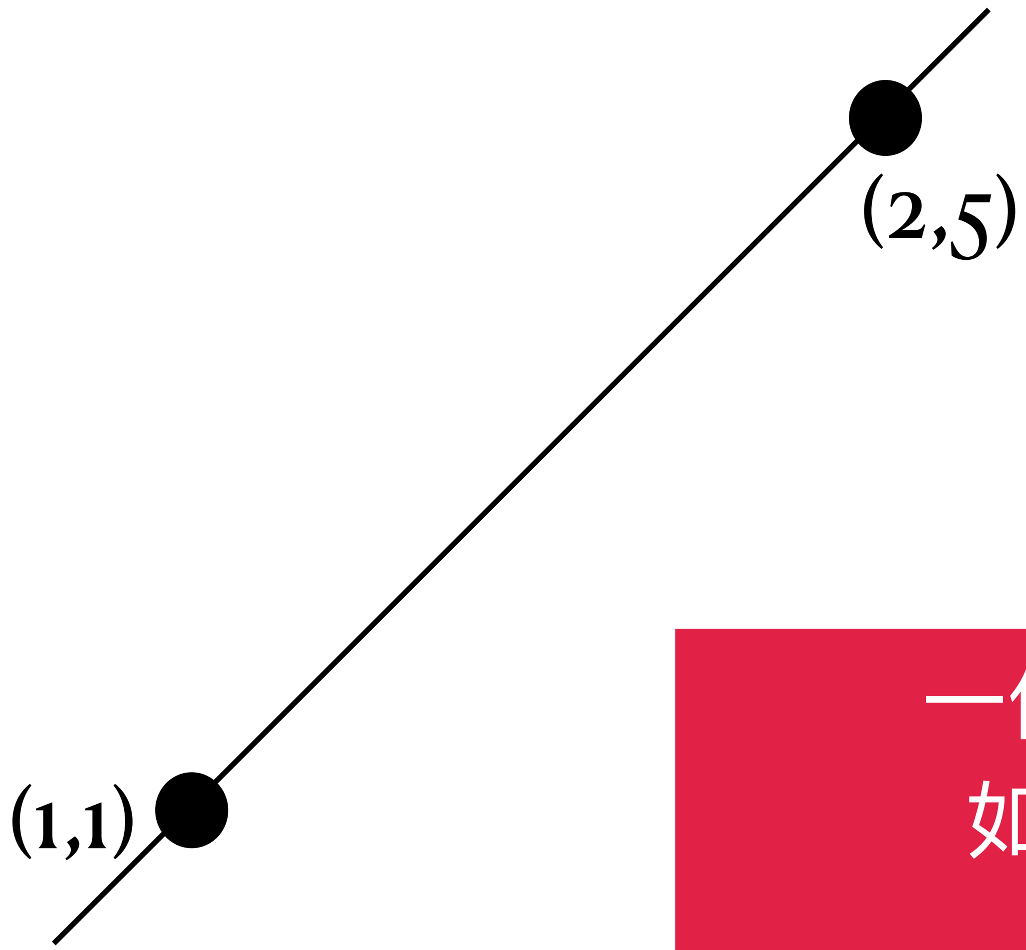
```python
import numpy as np
from numpy.linalg import inv
```

```python
a = np.matrix([1,2,3,4])
A = np.reshape(a,(2,2))
inv(A) @ A
```

可使用
array

$$A * A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
>>> inv(A) @ A
matrix([[1.00000000e+00, 0.00000000e+00],
        [1.11022302e-16, 1.00000000e+00]])
```

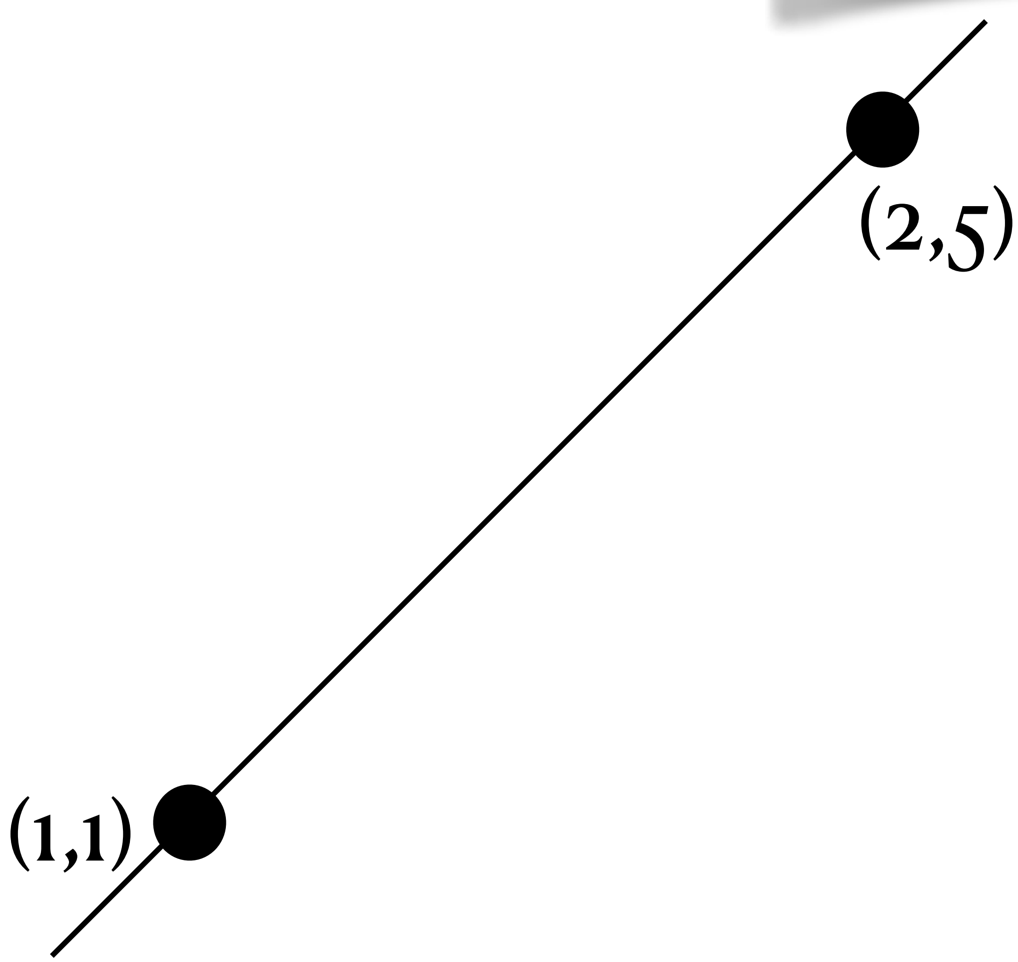


一條直線通過兩點
如何求方程式？

使用反矩陣，解線性系統

$$ax + b = y$$

$$\begin{bmatrix} x & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = y$$



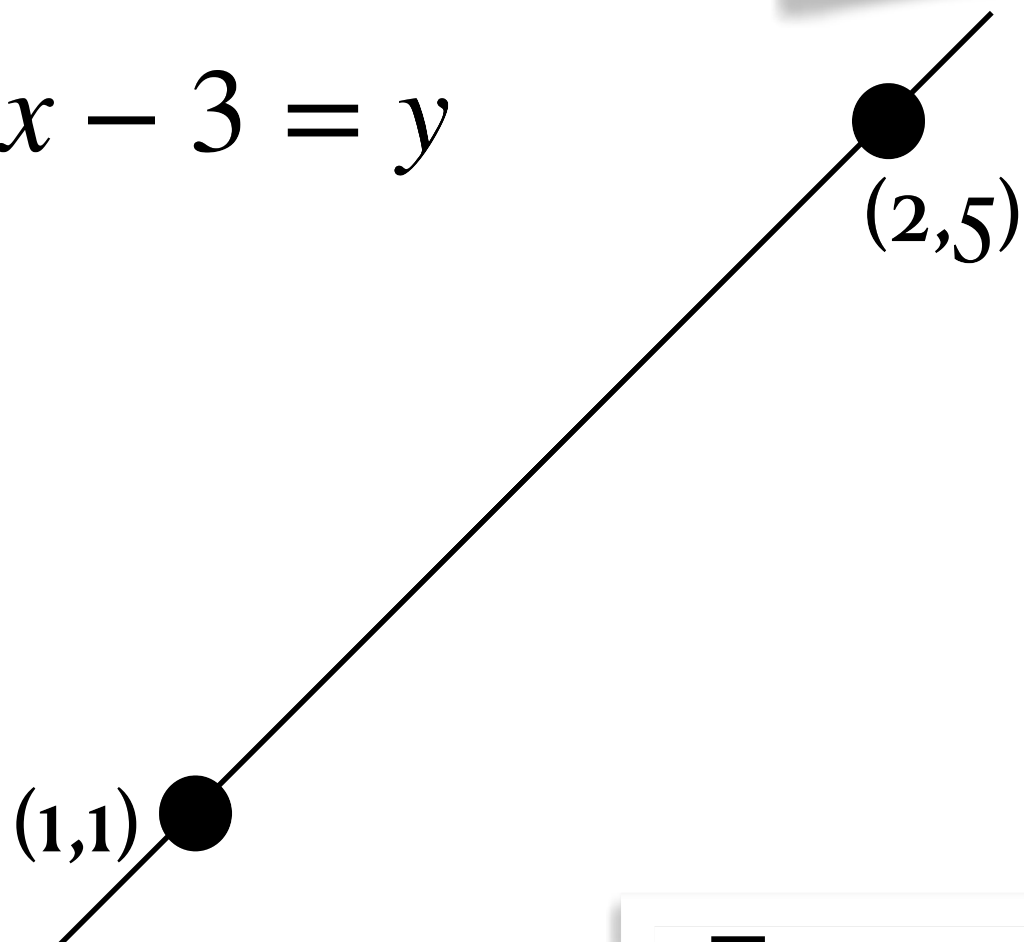
$$\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = 5$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = 1$$

$$ax + b = y$$

$$\begin{bmatrix} x & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = y$$

$$4x - 3 = y$$



$$\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = 5$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = 1$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

A Linear System

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \boxed{inv\left(\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}\right)\begin{bmatrix} 5 \\ 1 \end{bmatrix}}$$

```python
import numpy as np
from numpy.linalg import inv

A = np.matrix([[2,1],[1,1]])
invA = inv(A)
ans = invA @ np.matrix([[5],[1]])
```

$$inv\left(\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}\right)\begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

```
>>> A
matrix([[2, 1],
        [1, 1]])
```

```
>>> invA
matrix([[ 1., -1.],
        [-1.,  2.]])
```

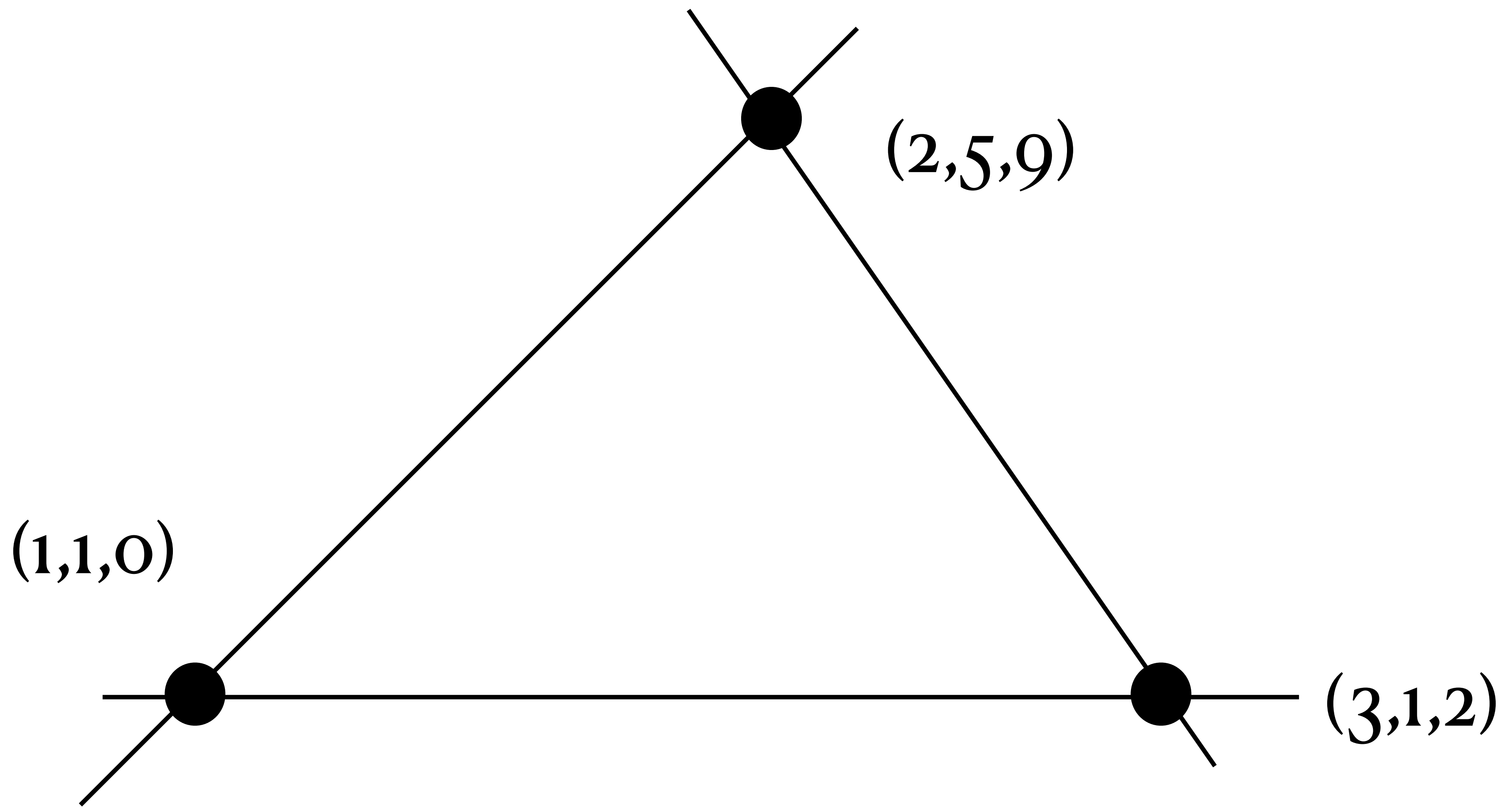
```
>>> ans
matrix([[ 4.],
        [-3.]])
```

驗證

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

```
>>> A @ ans
matrix([[5.],
        [1.]])
```

在三度空間的三個點，不共線，構成一個面



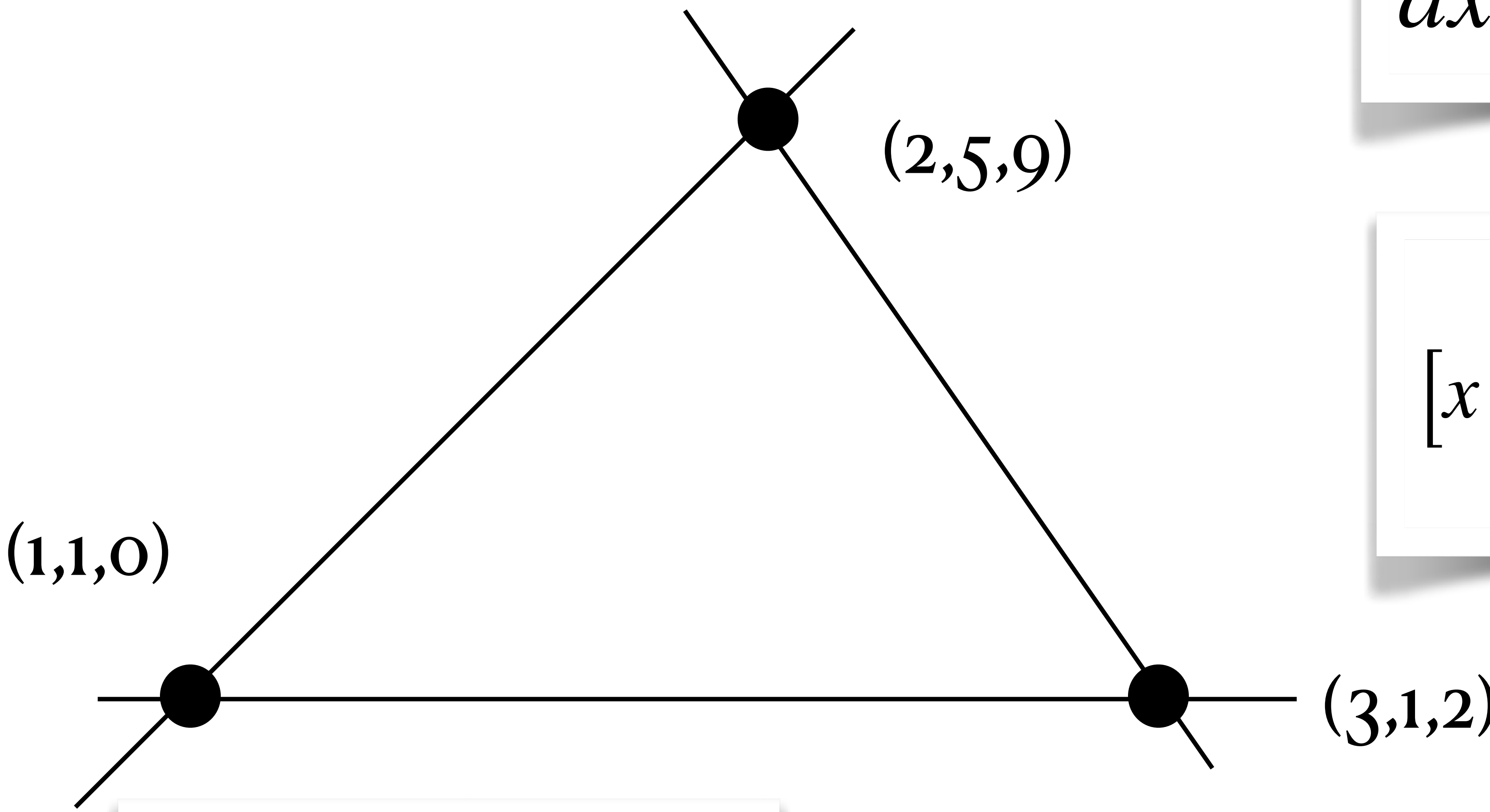
一個平面通過三個點
如何求方程式？

使用反矩陣，解線性系統

$$\begin{bmatrix} 2 & 5 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 9$$

$$ax + by + c = z$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = z$$



$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

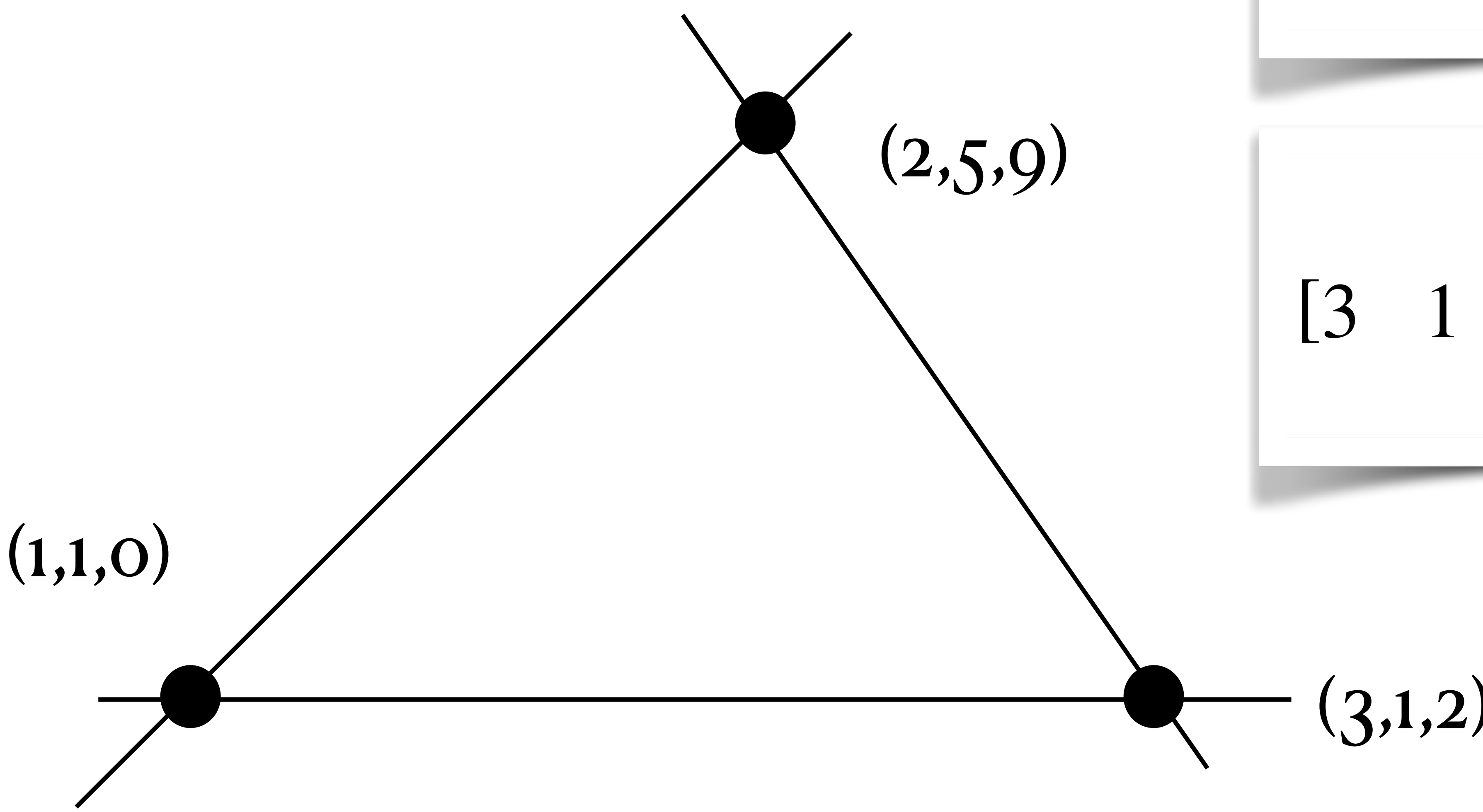
$$\begin{bmatrix} 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 2$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = z$$

$$\begin{bmatrix} 2 & 5 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 9$$

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

$$\begin{bmatrix} 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 2$$



$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = z$$

$$\begin{bmatrix} 2 & 5 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 9$$

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

$$\begin{bmatrix} 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 2$$

$$\begin{bmatrix} 2 & 5 & 1 \\ 1 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \\ 2 \end{bmatrix}$$

Linear System

$$\begin{bmatrix} 2 & 5 & 1 \\ 1 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = inv\left(\begin{bmatrix} 2 & 5 & 1 \\ 1 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix}\right)\begin{bmatrix} 9 \\ 0 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

```
import numpy as np
from numpy.linalg import inv

A = np.matrix([[2,1],[1,1]])
invA = inv(A)
ans = invA @ np.matrix([[5],[1]])
```

如何修改成
根據三個點，求平面
係數？

$$inv(\begin{bmatrix} 2 & 5 & 1 \\ 1 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix}) \begin{bmatrix} 9 \\ 0 \\ 2 \end{bmatrix}$$

驗證

$$\begin{bmatrix} 2 & 5 & 1 \\ 1 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \\ 2 \end{bmatrix}$$

如何修改成
驗證ans是否正確？

```
>>> A @ ans
matrix([[5.],
        [1.]])
```

Solve a symmetric linear system by the conjugate gradient method

Python conjugate gradient method

```
def conjgrad(A, b, x0, tol):
    r = b - A @ x0
    p = r
    rsold = np.transpose(r) @ r
    x = x0
    count = 0
    while np.sqrt(rsold) > tol:
        Ap = A @ p
        alpha = rsold /(np.transpose(p) @ Ap)
        x = x + alpha * p
        r = r - alpha * Ap
        rsnew = np.transpose(r) @ r
        p = r + (rsnew/rsold) * p
        rsold = rsnew
        count += 1
        if count % 2000 == 0:
            print('loop ', count)
            break
    return x
```

```
A = np.array([[2,5,1],[1,1,1],[3,1,1]])
A = np.transpose(A) @ A + np.eye(3)
x = np.array([[2],[0],[9]])
b = A @ x
x = np.zeros((3,1))
ans = conjgrad(A, b, x, 10**-8)
print('x :', ans)
print('mean abs error : ', np.mean(abs(A @ ans - b)))
```

```python
import time
n = 20000
m = 30
AA = np.random.rand(n,m)
A = AA @ np.transpose(AA) + np.eye(n)
x = np.random.rand(n,1)
b = A@x
t = time.time()
ans = conjgrad(A,b,np.zeros((n,1)),10**-8)
tt = t = time.time() - t
print('mean abs error : ', np.mean(abs(A @ ans - b)))
print('execution time :', tt)
```