In [37]:
```python
from sympy import *
import numpy as np
```

In [38]:
```python
x,y,z,a,b,c,d=symbols('x, y, z, a, b, c,d')
```

In [39]:
```python
P1=Matrix([2,7])
P2=Matrix([1,-2])
P3=Matrix([0,-7])
P4=Matrix([-2,-27])
P5=Matrix([5,110])
P6=Matrix([-1,-20])
```

In [40]:
```python
expr_1=a*x**3 + b*x**2 + c*x +d
```

In [41]:
```python
expr_1
```

Out[41]: $ax^3 + bx^2 + cx + d$

In [42]:
```python
Exp_P1=expr_1.subs(x,P1[0])
Exp_P2=expr_1.subs(x,P2[0])
Exp_P3=expr_1.subs(x,P3[0])
Exp_P4=expr_1.subs(x,P4[0])
Exp_P5=expr_1.subs(x,P5[0])
Exp_P6=expr_1.subs(x,P6[0])
pprint(Exp_P1)
pprint(Exp_P2)
pprint(Exp_P3)
pprint(Exp_P4)
pprint(Exp_P5)
pprint(Exp_P6)
```

```
8⋅a + 4⋅b + 2⋅c + d
a + b + c + d
d
-8⋅a + 4⋅b - 2⋅c + d
125⋅a + 25⋅b + 5⋅c + d
-a + b - c + d
```

In [43]:
```python
eq_P1=Eq(Exp_P1,P1[1])
eq_P2=Eq(Exp_P2,P2[1])
eq_P3=Eq(Exp_P3,P3[1])
eq_P4=Eq(Exp_P4,P4[1])
eq_P5=Eq(Exp_P5,P5[1])
eq_P6=Eq(Exp_P6,P6[1])
pprint(eq_P1)
pprint(eq_P2)
pprint(eq_P3)
pprint(eq_P4)
pprint(eq_P5)
pprint(eq_P6)
```

```
8⋅a + 4⋅b + 2⋅c + d = 7
a + b + c + d = -2
d = -7
-8⋅a + 4⋅b - 2⋅c + d = -27
125⋅a + 25⋅b + 5⋅c + d = 110
-a + b - c + d = -20
```

In [44]:
```python
A=Matrix([[8,4,2,1],[1,1,1,1],[0,0,0,1],[-8,4,-2,1],[125,25,5,1],[-1,1,-1,1]])
display(A.transpose())
```

```python
display(A)
```

$$\begin{bmatrix} 8 & 1 & 0 & -8 & 125 & -1 \\ 4 & 1 & 0 & 4 & 25 & 1 \\ 2 & 1 & 0 & -2 & 5 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 8 & 4 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ -8 & 4 & -2 & 1 \\ 125 & 25 & 5 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

In [45]:
```python
U=Matrix([7,-2,-7,-27,110,-20])
U
```

Out[45]:
$$\begin{bmatrix} 7 \\ -2 \\ -7 \\ -27 \\ 110 \\ -20 \end{bmatrix}$$

In [46]:
```python
X=Matrix([a,b,c,d])
X
```

Out[46]:
$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

In [47]:
```python
ATA=(A.transpose())*A
ATA
```

Out[47]:
$$\begin{bmatrix} 15755 & 3125 & 659 & 125 \\ 3125 & 659 & 125 & 35 \\ 659 & 125 & 35 & 5 \\ 125 & 35 & 5 & 6 \end{bmatrix}$$

In [48]:
```python
ATU=A.transpose()*U
ATU
```

Out[48]:
$$\begin{bmatrix} 14040 \\ 2648 \\ 636 \\ 61 \end{bmatrix}$$

In [49]:
```python
system=Matrix([ATA.transpose(),ATU.transpose()]).transpose()
display(system)
```

$$\begin{bmatrix} 15755 & 3125 & 659 & 125 & 14040 \\ 3125 & 659 & 125 & 35 & 2648 \\ 659 & 125 & 35 & 5 & 636 \\ 125 & 35 & 5 & 6 & 61 \end{bmatrix}$$

In [50]:
```
result=solve_linear_system(system,a,b,c,d)
result
```

Out[50]:
```
{a: 21643/29802, b: -1343/14901, c: 61495/9934, d: -47663/4967}
```

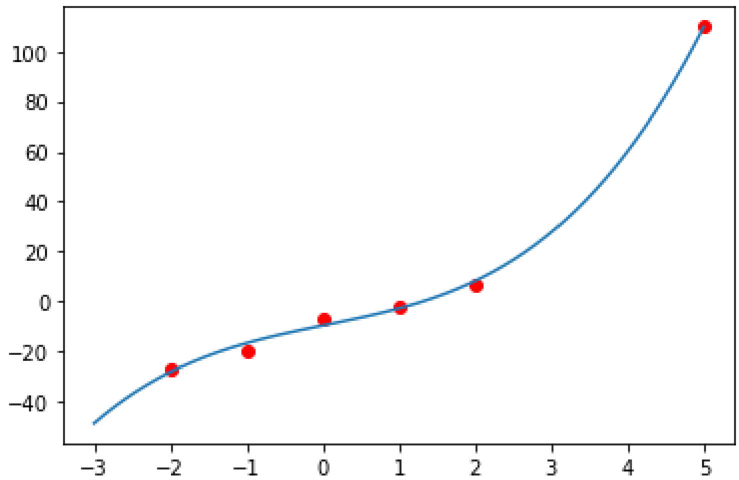
In [51]:
```
expr_r=expr_1.subs(result)
expr_r
```

Out[51]:

$$\frac{21643x^3}{29802} - \frac{1343x^2}{14901} + \frac{61495x}{9934} - \frac{47663}{4967}$$

In [52]:
```
equat=Eq(expr_r,y)
```

In [54]:
```
import matplotlib.pyplot as plt
#plt1=plot_implicit(equat,(x, -25, 25), (y, -150, 150),show=False)
plt2=plt
plt2.figure()
plt2.scatter([P1[0],P2[0],P3[0],P4[0],P5[0],P6[0]],[P1[1],P2[1],P3[1],P4[1],P5[1],I
u=np.linspace(-3,5,1000)
v=np.array([expr_r.subs(x,i) for i in u])
plt2.plot(u,v)
plt2.show()
```



In [ ]:

In [ ]: