[1]:
```
import numpy as np
import cvxpy as cp

n = 2
A1 = np.array([[-1/4, -1/4],[-4/4, 0]])
A2 = np.array([[3/4, 3/4],[-2/4, 1/4]])
id = np.array([[1, 0],[0, 1]])
C = np.array([[0,0],[0,0]])

X = cp.Variable((n,n), symmetric=True)
constraints = [X - A1.T @ X @ A1 - 0.0002*id >> 0, X - A2.T @ X @ A2 - 0.0002*id␣
↪>> 0 , X - 0.0002*id >> 0]
prob = cp.Problem(cp.Minimize(cp.trace(C @ X)),constraints)
prob.solve()
X.value
```

[1]:
```
array([[93.85890325, 22.86467265],
       [22.86467265, 73.98959241]])
```

Let's try it out with $P = \begin{bmatrix} 93 & 22 \\ 22 & 73 \end{bmatrix}$.

[2]:
```
from numpy import linalg as LA
P = np.array([[93, 22],[22, 73]])
B1 = P - (A1.T) @ P @ A1
B2 = P - (A2.T) @ P @ A2
print(LA.det(B1),LA.det(B2))
print("\nP - (A1.T) @ P @ A1 = \n", B1)
print("\nP - (A2.T) @ P @ A2 = \n", B2)
```

```
99.93749999999997 15.503906249999993

P - (A1.T) @ P @ A1 =
 [[ 3.1875 10.6875]
 [10.6875 67.1875]]

P - (A2.T) @ P @ A2 =
 [[ 38.9375 -17.0625]
 [-17.0625   7.875 ]]
```