```
from aide_design.play import*
x = 0.2
X = 0.15
K = 2
delta_t = 1
def Co(X,delta_t,K):
  Co = (-2 * X + delta_t / K)/(2 * (1 - X) + delta_t / K)
  return Co

def C1(X,delta_t,K):
  C1 = (2 * X + delta_t / K)/(2 * (1 - X) + delta_t / K)
  return C1

def C2(X,delta_t,K):
  C2 = (2 * (1 - X) - delta_t / K)/(2 * (1 - X) + delta_t / K)
  return C2

C_0 = Co(X,delta_t,K)
C_1 = C1(X,delta_t,K)
C_2 = C2(X,delta_t,K)

if C_0 + C_1 + C_2 == 1:
  print('Valid') Valid
```

$$Q_{e2} = c_0 Q_{i2} + c_1 Q_{i1} + c_2 Q_{e1}$$

$$Q_{e2} = \frac{1}{11} Q_{i2} + \frac{4}{11} Q_{i1} + \frac{6}{11} Q_{e1}$$

```
time = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20])
Q_i = np.array([4000,7000,11000,17000,22000,27000,30000,28000,25000,23000,20000,17000,
14000,11000,8000,5000,4000,4000,4000,4000])
Q_e = np.zeros(len(time))
Q_e[0] = Q_i[0]
for i in range(len(time)-1):
  Q_e[i+1] = C_0 * Q_i[i+1] + C_1 * Q_i[i] + C_2 * Q_e[i]

plt.plot(time, Q_i)
plt.plot(time, Q_e)
plt.xlabel('Time (days)')
plt.ylabel('Flow Rate (cfs)')
plt.legend(['Qi','Qe'])
plt.savefig('/Users/jillianwhiting/github/Jillian-Whiting/Images/Q_3')
plt.show()
```

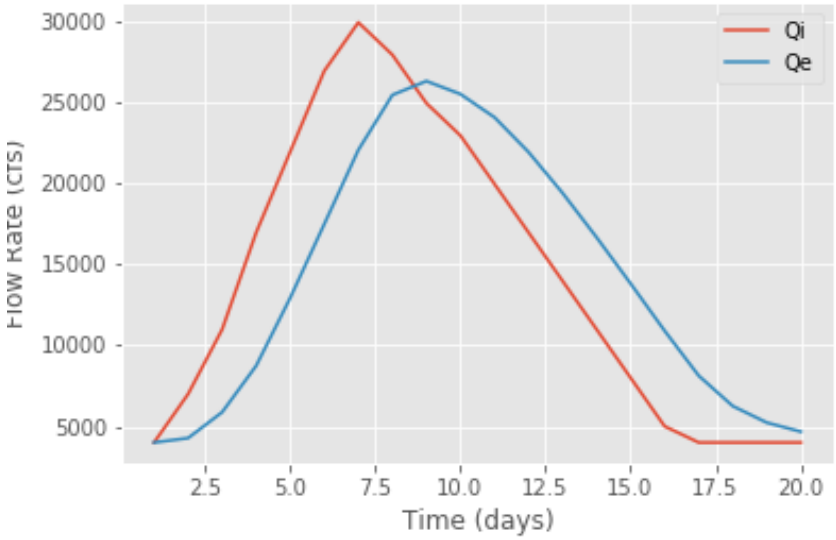
Qi
Qe
30000
25000
20000
15000
10000
5000
Flow Rate (cfs)
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (days)

Figure 1: q_3