徐昶亘　r07525117 工海碩二

一、

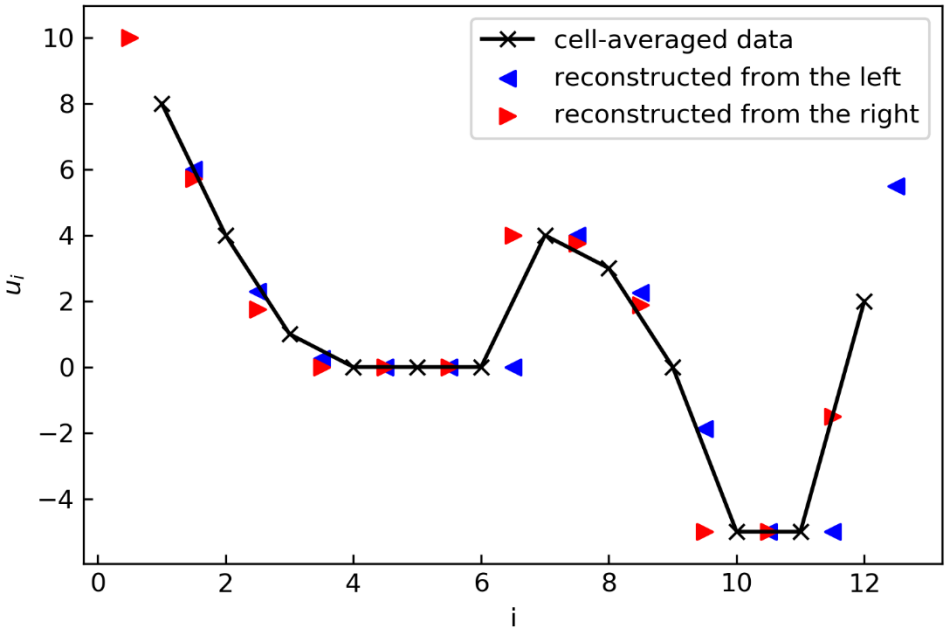
Input :

$\vec{u} = [8,4,1,0,0,0,4,3,0,-5,-5,2]$

Output:

$\overrightarrow{u^{+}} = [10,5.71429,1.75,0,0,0,4,3.75,1.875,-5,-5,-1.5]$

$\overrightarrow{u^{-}} = [6,2.28571,0.25,0,0,0,4,2.25,-1.875,-5,-5,5.5]$



Code:

```
import numpy as np
import matplotlib.pyplot as plt
import os
os.chdir(r'D:\ShallowWaterComputation\HW5')

def MUSCL(u):
    u = np.concatenate( ( [ 2*u[0] -u[1] ] , u , [ 2*u[-1] -
u[-2] ] ) )
    GN = 1
    DelP = u[GN+1:] - u[GN:-GN]
```

```
    DelN = u[GN:-GN]- u[GN-1:-GN-1]

    DelBar = ( DelP*abs(DelN) + abs(DelP)*DelN )/( abs(DelP) +
abs(DelN) )
    DelBar[np.isnan(DelBar)] = 0

    uP = u[GN:-GN] - 0.5*DelBar
    uN = u[GN:-GN] + 0.5*DelBar
    return uP,uN

u = np.array( [0,0,1,4,-4,0,0,5,4,2,-1,2] )
uP,uN = MUSCL(u)

u2 = np.array( [8,4,1,0,0,0,4,3,0,-5,-5,2] )
u2P,u2N = MUSCL(u2)

i = np.arange(1,13)
#plt.figure(figsize=(5,5))
plt.plot(i , u2 , '-x' ,c = 'black' , label = "cell-averaged
data" )
#plt.scatter(i , u2 , marker = 'x' , c = 'black')
plt.scatter(i+0.5 , u2N , marker = '<' , c = 'blue', label =
"reconstructed from the left" )
plt.scatter(i-0.5 , u2P , marker = '>' , c = 'red', label =
"reconstructed from the right")
plt.legend()

plt.xlabel(xlabel='i')
plt.ylabel(ylabel='$u_i$')

plt.tick_params(which='both',direction='in')
plt.savefig("HW5Q1.png" , dpi = 300)
```

二、

Input :

$$\begin{cases} \overrightarrow{\eta^-} = [-0.01, -0.01, -0.33, -0.40, 0.28, 0.29, 0.13] \\ \overrightarrow{\eta^+} = [-0.01, -0.13, -0.4, 0.17, 0.29, 0.13, 0.13] \\ \overrightarrow{U^-} = [-0.02, -0.09, -0.38, -0.45, 0.36, 0.36, -0.02] \\ \overrightarrow{U^+} = [0.02, -0.18, -0.45, 0.28, 0.36, 0.20, 0.017] \\ h = 10 \end{cases}$$

F	0	-0.7518	-3.671	-3.523	3.652	3.689	-0.01517
G	-2.958	-2.287	-30.43	-47.79	29.64	30.34	10.96

Code :

```
import numpy as np
g = 9.806

def HLLC(EtaN, EtaP, UN, UP, h):
    HP = EtaP + h
    HN = EtaN + h
    HU_N = HN*UN
    HU_P = HP*UP

    NegPara = HN*(UN**2) + 0.5*g*(EtaN**2 + 2*EtaN*h)
    PosPara = HP*(UP**2) + 0.5*g*(EtaP**2 + 2*EtaP*h)



    us = 0.5*(UN+UP) + (g*HN)**0.5 - (g*HP)**0.5
    cs = 0.5*( (g*HN)**0.5 + (g*HP)**0.5 ) + 0.25*( UN - UP )
    SN = np.array( [ min(i,j) for i , j in zip( UN -
(g*HN)**0.5 , us - cs ) ] )
    SP = np.array( [ max(i,j) for i , j in zip( UP +
(g*HP)**0.5 , us + cs ) ] )
    PhiSN = HN*(SN - UN)/(SN - us)*np.array( [[1]*len(us),
list(us)] )
    PhiSP = HP*(SP - UP)/(SP - us)*np.array( [[1]*len(us),
list(us)] )
    PhiN = np.array( [ HN , HU_N] )
    PhiP = np.array( [ HP , HU_P] )
```

```
    fN = np.array( [ HU_N , NegPara ] )
    fP = np.array( [ HU_P , PosPara ] )

    f = np.zeros((2,len(SN)))
    for i in range( len(SN) ):
        if ( 0 <= SN[i] ):
            f[:,[i]] = fN[:,[i]]
        elif ( SN[i] < 0 <= us[i] ):
            f[:,[i]] = fN[:,[i]] + SN[i]*( PhiSN[:,[i]] -
PhiN[:,[i]] )
        elif ( us[i] < 0 <= SP[i]  ):
            f[:,[i]] = fP[:,[i]] + SP[i]*( PhiSP[:,[i]] -
PhiP[:,[i]] )
        elif ( SP[i] < 0 ):
            f[:,[i]] = fP[:,[i]]
    return f[0] , f[1]

EtaN2 = np.array([-0.01,-0.01,-0.33,-0.40,0.28,0.29,0.13])
EtaP2 = np.array([-0.01,-0.13,-0.4,0.17,0.29,0.13,0.13])
UN2 = np.array([-0.02,-0.09,-0.38,-0.45,0.36,0.36,-0.02])
UP2 = np.array([0.02,-0.18,-0.45,0.28,0.36,0.20,0.017])
h2 = 10

F2,G2 = HLLC(EtaN2, EtaP2, UN2, UP2, h2)
```