```
[51]  import numpy as np
```

Seminar 6

Open Channel Flow

1. **Hydrodynamics III**

Problem 1 - Open Channel Flow

For the rectangular channel ($B = 4.6$ m, and $S_o = 1/400$) taking the n value as 0.040 s/m $^{1/3}$, calculate the depth of flow, D, that corresponds to a discharge of 2.83 m^3/s

(a) when $B >> D$,

(b) When $B >> D$ is not true.

Solution of Problem 1

The relevant equations can be found in slides L6 - slides 13-15, 19

The main equation is:

$$Q = \frac{A}{n} \cdot R^{2/3} \cdot S_o^{1/2}$$

with $A = B \cdot B, P = (B + 2D)$

Case 1. For wide rectangular channel $R = D$

$$Q = (B \cdot D/n)(D)^{2/3} S_o^{1/2}$$

$$D^{5/3} = (Q \cdot n)/(B S_o^{1/2})$$

Case 2. For channel $R \neq D$

$$Q = (B \cdot D/n)(B \cdot D/(B + 2 \cdot D)^{2/3} S_o^{1/2}$$

D can not be isolated and so can not be directly solved.

```
[7]  #Solution problem 1(a)
```

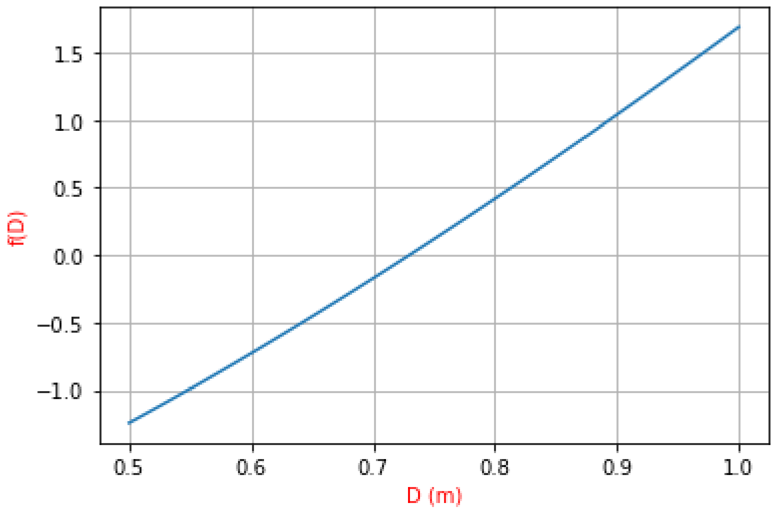
```
# Given are:
B1 = 4.6 # m, width of channel
So1= 1/400 # [ ], slope of riverbed
n1 = 0.040 # sm^1/3, Manning coefficient
Q1 = 2.83# m^3/s, discharge.
g1 = 9.81 # m/s^2, gravity

# Interim calculation
Int1 = (B1*So1**(1/2)) # B^2/3*So^1/2

# calculation (see equation above for case 1)
D1_a = ((Q1*n1)/Int1)**(3/5)

#output
print("The resulting diameter is: {0:1.2f}".format(D1_a), "m")
```

```
The resulting diameter is: 0.65 m
```

[52]

```
#Solution problem 1(b)
# No direct solution available.
# Let us define the equation Q = f(D), and f(D)-Q = 0

def mych(D):
    D_1 = (B1*D/n1)*((B1*D)/(B1+2*D))**(2/3)*So1**(1/2)-Q1
    return D_1

# Q = f(D); f(D)-Q = 0, with known Q, we need to find a D where
f(D)-Q = 0.
# Easiest way to do is to plot f(D) for many D's

import matplotlib.pyplot as plt
D = np.linspace(0.5,1, 100)
D_n = mych(D)
plt.plot(D, D_n ), plt.grid()
plt.xlabel("D (m)", color = 'r'), plt.ylabel("f(D)", color = 'r')

print("In this example f(D) = 0 for D = 0.7-0.8 m")
```

```
In this example f(D) = 0 for D = 0.7-0.8 m
```



Iterative Method

The plot provides an estimate for D, which is between 0.7-0.8 m.

To get the exact iterative methods are required. There are quite a few iterative methods that can solve this problem, e.g., Newton's method, Bisection method, Secant method.

The solution is also called **root**.

Details of this method can be found in standard Numerical Mathematics books.

python ***scipy.optimize*** library provide functions to easily compute the root. We use ***fsolve*** which uses Newton method to find root here.

Newton Method:

$$D_{new} = D - \frac{f'(D)}{f(D)}$$

When $D_{new} = D$, i.e. $\frac{f'(D)}{f(D)} \approx 0$, the required D is obtained.

fsolve code in scipy check *help(scipy.optimize)* to see detail

fsolve(f, Dg) with function *f* and *Dg* is the first guess of *D*.

[9]

```
from scipy.optimize import fsolve

D1_g = 0.65

D1_ex = fsolve(mych,D1_g)

print("The required diameter is: {0:1.3f}".format(D1_ex[0]),"m")
```

```
The required diameter is: 0.730 m
```

Problem 2 - Optimum section

A rectangular, concrete lined channel is to be constructed to carry floodwater. The slope of the ground surface is 1 in 500. The design discharge is 10 m^3/s.

(a) Calculate the proportions of the rectangular channel that will minimize excavation and result in the optimum hydraulic section.

(b) If the cross-sectional area of flow is kept the same as in part (a) but for safety reasons the depth of flow in the channel is limited to 1.00 m, what will be the discharge now?

Solution of Problem 2

The relevant lecture slides are: L6 -13 and 21

Important relations are:

$$Q = \frac{A}{n}R^{2/3}S_o^{1/2}$$

For optimum section $A = 2D^2$ and $R = 0.5D$

$$Q = \frac{2D^2}{n}(0.5D)^{2/3}S_o^{1/2}$$

[2]

```
# solution 2(a)
# Given are

So2 = 1/500 # [], slope of bed
n2 = 0.015 # s/m^1/3, Manning coeff.
Q2 = 10 # m^3/s, discharge

#interim calculation
Int2 = (2/n2)*0.5**(2/3)*So2**(1/2) # all in RHS except D

#Calculation
D2 = (Q2/Int2)**(3/8)
A2 = 2*D2**2

# Output
print("The required diameter is: {0:1.3f}".format(D2),"m")
print("The required width is: {0:1.3f}".format(2*D2),"m")
print("The required Area is: {0:1.3f}".format(A2),"m\u00b2")
```

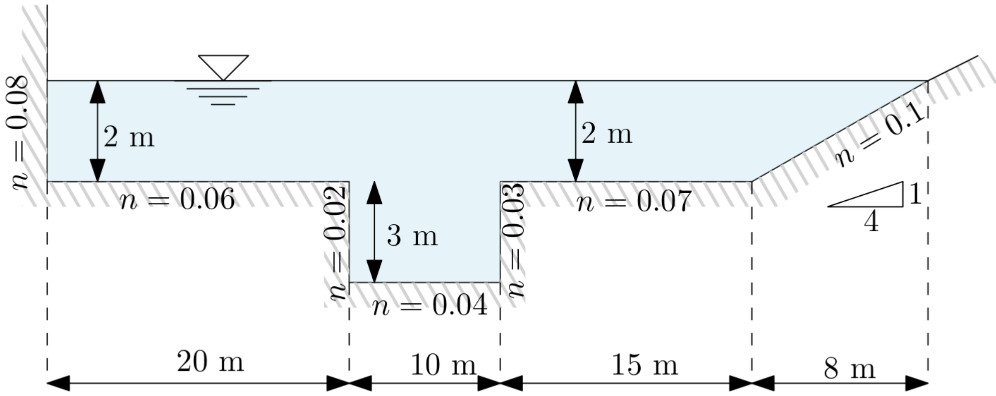
```
The required diameter is: 1.444 m
The required width is: 2.887 m
The required Area is: 4.168 m²
```

```python
# solution 2(b)
# Given are

D2_a = 1.0 # m, new depth

# interim equation
W2 = A2/D2_a # m, Width = Area/Depth
Pr2 = W2 + 2*D2_a # m, wetted perimeter Pr = W+2D

# Calculation (using Manning eq.)
Q2_a = A2/n2*(A2/Pr2)**(2/3)*(1/500)**(1/2)

# Output
print("The required discharge is:
{0:1.3f}".format(Q2_a),"m\u00b3/s")
if Q2_a>Q2:
    Q2_c = (Q2_a-Q2)/Q2*100
    print("The increase in discharge is: {0:1.3f}".format(Q2_c),
"%")
elif Q2_a<Q2:
    Q2_c = (Q2-Q2_a)/Q2*100
    print("The decreased in discharge is: {0:1.2f}".format(Q2_c),
"%")
```

```
The required discharge is: 9.570 m³/s
The decreased in discharge is: 4.30 %
```

Problem 3 - Compound Channel

A compound channel has the cross-section shown in Fig. below. The values next to the perimeter boundaries are the Manning roughness values (n s/m$^{1/3}$1/3). Calculate n_{av}.



Solution Problem 3 - Compound Channel

The relevant information can be found in slides: L6 - 25.

The main equation is:

$$n_{av} = \left[\frac{\sum_{i=1}^{N} P_i \cdot n_i^2}{P} \right]^{1/2}$$

Provides information are:

[8]

```
# solution Problem 3,
#Given

n3_l = np.array([0.080, 0.060, 0.020, 0.040, 0.030, 0.070, 0.1])
P3_l = np.array([2, 20, 3, 10, 3, 15, 8.246])

# interim calculation
P3 = np.sum(P3_l)
Pn3_l = P3_l*n3_l**2

# calculation
n3_av = (np.sum(Pn3_l)/P3)**(1/2)

#Output
print("The required average Manning coefficient is:
{0:1.3f}".format(n3_av),"s/m^1/3")
#import pandas as pd
#Data3 = pd.DataFrame({"n": n3_l, "Pr": P3_l, "n.Pr":Pn3_l })
#Data3
```

```
The required average Manning coefficient is: 0.065 s/m^1/3
```

Problem 4- Specific Energy

Rectangular channel:

Breadth B = 1.0 m

Flow Q = 1 m^3/s

Water depth D = 5 m – 0.1 m

Calculate Area A, Velocity v, specific energy E and Froude number F

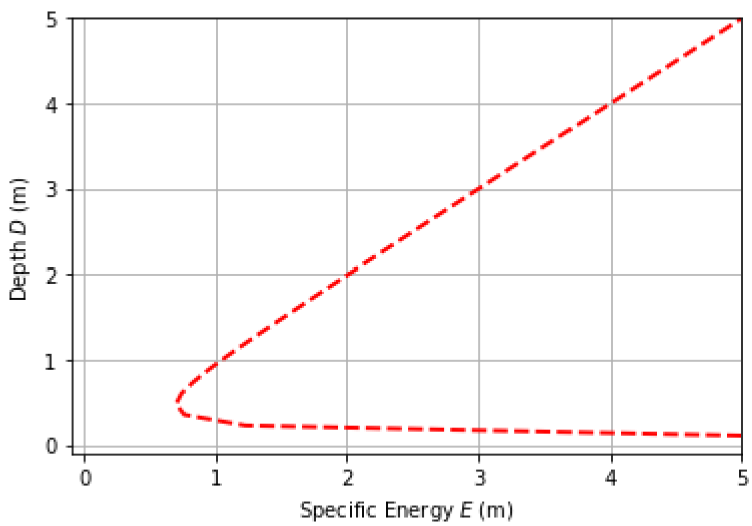
[98]

```
#Given
Q4 = 1 # m^3/s, constant discharge
B4 = 1.0 # m, width of the channel
g4 = 9.81 # m/s^2, gravity
D4 = np.linspace(0.1, 5, 40) # m, 30 depths from 0.1 to 5 m
```

```
# computation
A4 = B4*D4 # m^2, area of flow
v4 = Q4/A4 # m/s, velocity = Q/A
v4_2g = (v4**2)/(2*g4) # m, kinetic head v^2/2g
E4 = D4+v4_2g # m, Specific energy, E = D+ v2/2g
F4 = v4/(g4*D4)**(1/2) # (), Froude Number

import pandas as pd
Data = pd.DataFrame({"Depth": D4, "Area": A4, "velocity":v4,
"V^2/2g": v4_2g, "E": E4, "F": F4 })
Data.head(5)
```

	Depth	Area	velocity	V^2/2g	E	F
0	0.100000	0.100000	10.000000	5.096840	5.196840	10.096376
1	0.225641	0.225641	4.431818	1.001071	1.226712	2.978780
2	0.351282	0.351282	2.846715	0.413037	0.764319	1.533492
3	0.476923	0.476923	2.096774	0.224081	0.701004	0.969378
4	0.602564	0.602564	1.659574	0.140377	0.742941	0.682591

[147]
```
#solution Problem 4
#Plot

import matplotlib.pyplot as plt
plt.plot(E4, D4, '--r', lw=2)
plt.xlabel(r"Specific Energy $E$ (m)")
plt.ylabel(r"Depth $D$ (m)")
plt.axis([-0.1,5, -0.1,5])
plt.grid()
plt.savefig("fig11.pdf", dpi=300)
```



Assignment problems - Next week