Applied Computational Methods in Mechanical Sciences

(ME466)

Assignment 4

Himanshu Kumar

16ME234

August 26, 2019

**Problem Statement 1:**

A spherical tank is to be designed to hold water for a small village in a developing country. The volume of liquid it can hold can be computed as:

with R=3m, what depth must the tank be filled so that it holds 30m3. Solve the problem by developing a code using Newton-Raphson method and compute your answer for 5 decimal points accuracy. Calculate the number of iterations required to converge to the given criteria. Compute approximate relative error after every iteration.

**Python Code:**

import time

from math import pi as pi

err\_lim = 0.00001

init\_guess= 3

r\_val=3

def V(h,r):

return(pi\*h\*h\*(3\*r-h)/3)

def dV(h,r):

return(pi\*h\*(2\*r-h))

def NR(r,vol,guess\_h,err\_lim):

x= guess\_h

err = 1

itr = 0

while(err>err\_lim):

itr = itr+1

y= x-( (V(x,r)-vol)/(dV(x,r)) )

err = (y-x)/y

if(err<0):

err = err\*(-1)

print("\n X[",itr-1,"] =",x,"\n X[",itr,"] =",y)

print("Error is :",err)

x=y

print("\nSolution :",y)

print("Iterations :",itr)

print ("CPU time: ", time.process\_time(),'s')

return(y,itr)

z = NR(r\_val,30,init\_guess,err\_lim)

**Results:**

Solution : 2.0269057283100134

Iterations : 4

CPU time: 0.234375 s

**Problem Statement 2:**

The manning equation can be written for a rectangular open channel flow as:

where Q is flow rate, S is slope and H is depth, n is manning roughness coefficient and b is breadth. Develop fixed point iteration snippet to solve for H, given: Q=5, s = 0.0002, B=20m and n=0.03 with an error limit of 0.05%. Prove that your scheme converges for all initial guess greater than or equal zero.

**Python Code:**

import time

from math import pi as pi

Q=5

s=0.0002

B=20

n=0.03

init =25

lim = 0.0005

def manning(Q,B,n,s,H):

a=pow((B+2\*H),(2/3))

z = pow(s,0.5)

f = (Q\*n\*a)/z

ret = pow(f,(3/5))\*(1/B)

return(ret)

def fpi(f,args,init\_guess,err\_lim):

#using function manning

itr = 0

x = init\_guess

print("\ninitial value :",x)

while(1):

itr = itr+1

argm = args + (x,)

y = f(\*argm)

err = (y-x)/y

if(err<0):

err = (-1)\*err

print("\nx :",y)

print("error :",err)

if(err<err\_lim):

break

x=y

print("\nIterations:",itr)

print ("CPU time: ", time.process\_time(),'s')

return(y)

ans = fpi(manning,(Q,B,n,s),init,lim)

**Result:**

x : 0.7023008646426819

error : 0.0004019013200968386

Iterations: 4

CPU time: 0.203125 s