NATIONAL INSTITUE OF TECHNOLOGY KARANATAKA, SURATHKAL

ASSIGNMENT 4

APPLIED COMPUTATIONAL METHODS IN MECHANICAL SCIENCES

RAJAT A CHANDAVAR – 16ME156

26-Aug-19

ASSIGNMENT ON NEWTON RAPHSON, FIXED POINT ITERATION SCHEME

Question 1 Answer:

Given $V = \frac{\pi h^2(3R-h)}{3}$ where R = 3m, V = 90 m^3

Rearranging,

$f(h) = \pi h^2(3R - h) - 90 = 9\pi h^2 - \pi h^3 - 90 = 0$

$f'(h) = 18\pi h - 3\pi h^2$

Since f(h) is a cubic equation, 3 solutions will exist. From Newton Raphson method, the solution will depend on the initial guess.

Program(C++):

```
#include<iostream>
#include<cmath>
#include<time.h>
using namespace std;
int main()
{
    clock_t start=clock();
    double h=-1,h0,rel_error;
    int iter=0;
    cout<<"\nInitial guess:"<<h;
    do
    {
        ++iter;
        h0=h;
        h=h-(9*3.14*pow(h,2)-3.14*pow(h,3)-90)/(18*3.14*h-3*3.14*pow(h,2));
        rel_error=(abs(h-h0)/abs(h))*100;
        cout<<"\nIteration:"<<iter<<" Relative error:"<<rel_error;
    }while(rel_error!=0);
    cout<<"\nFinal Depth:"<<h<<" m"<<"\nFinal Iteration:"<<iter;
    clock_t stop=clock();
    double timespent = (double)(stop-start)/(double)CLOCKS_PER_SEC;
    cout<<"\nCPU Time:"<<timespent<<" seconds";
}
```

Output:

```
Initial guess:1
Iteration:1 Relative error:57.9389
Iteration:2 Relative error:16.6571
Iteration:3 Relative error:0.517855
Iteration:4 Relative error:0.00065178
Iteration:5 Relative error:1.03996e-009
Iteration:6 Relative error:0
Final Depth:2.02751 m
Final Iteration:6
CPU Time:0.008 seconds
```

```
Initial guess:7
Iteration:1 Relative error:32.0506
Iteration:2 Relative error:13.8692
Iteration:3 Relative error:4.54288
Iteration:4 Relative error:0.462177
Iteration:5 Relative error:0.0046094
Iteration:6 Relative error:4.56355e-007
Iteration:7 Relative error:0
Final Depth:8.61369 m
Final Iteration:7
CPU Time:0.008 seconds
```

```
Initial guess:-1
Iteration:1 Relative error:47.0532
Iteration:2 Relative error:13.7108
Iteration:3 Relative error:1.19508
Iteration:4 Relative error:0.00870012
Iteration:5 Relative error:4.59756e-007
Iteration:6 Relative error:0
Final Depth:-1.6412 m
Final Iteration:6
CPU Time:0.012 seconds
```

Since Depth in a spherical tank has to be less than diameter and greater than 0 i.e. 0<=h<=2R, only the first solution or output is valid.

So, Depth = 2.02571m

Question 2 Answer:

$Q = \frac{\sqrt{S}(BH)^{\frac{5}{3}}}{n(B+2H)^{\frac{2}{3}}}$,where Q = 5, S = 0.0002, B = 20 and n = 0.03

Substituting these values,

$f(H) = 0.15(20 + 2H)^{\frac{2}{3}} - \sqrt{0.0002}(20H)^{\frac{5}{3}} = 0$

Since this is a bracketing method, we need select a range. Let us select (0,1) since f(0) = 1.0521 and f(1) = -0.90629. So, solution lies between 0 to 1.

Let us write f(H) in the form H=∅(H).

There are 2 ways of writing this.

Alternative 1:

$$H = \left[\frac{0.15(20 + 2H)^{\frac{2}{3}}}{\sqrt{0.0002}(20)^{\frac{5}{3}}}\right]^{\frac{3}{5}} = \left(\frac{0.15^{\frac{3}{5}}}{20(0.0002)^{0.3}}\right)(20 + 2H)^{\frac{2}{5}} = 0.2062(20 + 2H)^{\frac{2}{5}} = \emptyset(H)$$

$$\emptyset'(H) = 0.2062 * \frac{2}{5}(20 + 2H)^{-\frac{3}{5}} * 2 = \frac{0.165}{(20+2H)^{\frac{3}{5}}}$$

$$\emptyset'(0) = 0.02734 < 1 \; and \; \emptyset'(1) = 0.02582 < 1$$

Hence, Alternative 1 valid.

Also, since $(20 + 2H)^{\frac{3}{5}} > 1$, for all H>0, $\emptyset'(H) < 1$for all initial guesses>0 and hence scheme will converge for all values discussed above.

Alternative 2:

$$H = \frac{1}{2}\left[\left\{\frac{\sqrt{0.0002}(20H)^{\frac{5}{3}}}{0.15}\right\}^{\frac{3}{2}} - 20\right] = 5.606 * 10^{-3}(20H)^{\frac{5}{2}} - 10 = \emptyset(H)$$

$$\emptyset'(H) = 0.2803(20H)^{\frac{3}{2}}$$

$$\emptyset'(0) = 0 < 1 \; and \; \emptyset'(1) = 25.071 > 1$$

Hence, Alternative 2 invalid.

So, Alternative 1 chosen.

Program(C++)

```
1   #include<iostream>
2  #include<cmath>
3  #include<time.h>
4  using namespace std;
5  int main()
6  {
7      clock_t start=clock();
8      float h=0.5,h0,rel_error;
```

```
 9       cout<<"Initial guess = "<<h;
10       int iter=0;
11       do
12       {
13           ++iter;
14           h0=h;
15           h=(pow(0.15,0.6)/(20*pow(0.0002,0.3)))*pow(20+2*h,0.4);
16           rel_error=(abs(h-h0)/abs(h))*100;
17       }while(rel_error>0.05);
18       cout<<"\nIterations:"<<iter<<"\nDepth of channel:"<<h<<" m"<<"\nRelative
error = "<<rel_error;
19       clock_t stop=clock();
20       double timespent = (double)(stop-start)/(double)CLOCKS_PER_SEC;
21       cout<<"\nCPU Time:"<<timespent<<" seconds";
22   }
```

Output

```
Initial guess = 0.5
Iterations:3
Depth of channel:0.70229 m
Relative error = 0.0194357
CPU Time:0.027 seconds
```

```
Initial guess = 10
Iterations:4
Depth of channel:0.702297 m
Relative error = 0.0189517
CPU Time:0.002 seconds
```

```
Initial guess = 100
Iterations:5
Depth of channel:0.702294 m
Relative error = 0.00263101
CPU Time:0.002 seconds
```

It is evident from the outputs that irrespective of initial guess > 0, Depth of channel = 0.7023 m