Breaking Ice with Air cushion vehicle-Python

Aim:

1. Solve the minimum value of pressure using NR method.

$$x = p^3(1-\beta^2) + \left(0.4h\beta^2 - \frac{\sigma h^2}{r^2}\right)p^2 + \frac{\sigma^2 h^4}{3r^4} * p - \left(\frac{\sigma h^2}{3r^2}\right)^3$$

   Here, p-cushion pressure, h-thickness of the ice field , r-size of air cushion, σ-Tensile strength, β is the width of the ice field
   Consider β=0.5, r=40 feet, σ=150 psi
2. Find the minimum value of pressure for h=0.6ft
3. Table the values of P for h=[0.6,1.2,1.8,2.4,3,3.6,4.2]

History:

Ice sheet can be broken into small slabs with a heavy air cushion vehicle (ACV) traveling on float ice sheet at the critical speed.According to the simulation results, ACV traveling at the critical speed always chases after and pushes the wave crest ahead of ACV. Thus, the amplitude of the wave around ACV gradually increases due to the continual push of ACV. Ice sheet cracks when the stress and strain in the ice sheet accumulated to the critical value of failure. The critical speed of ACV should consist with the velocity of the surface wave around ACV. Under the thick ice sheet and shallow water environment in this paper, the critical speed of ACV is estimated as the velocity of the surface wave with a wavelength about triple the length of ACV hull.

Source:
https://www.researchgate.net/publication/288723093_Analysis_on_the_ice-breaking_mechanism_of_air_cushion_vehicle_and_the_critical_speed_estimation

- **Newton Raphson Method:**

In numerical analysis, the **Newton–Raphson method**, is a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function. The most basic version starts with a single-variable function f defined for a real variable x, the function's derivative $f\ '$, and an initial guess x_0 for a root of f. If the function satisfies sufficient assumptions and the initial guess is close, then

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}\,.$$

is a better approximation of the root than x_0. Geometrically, $(x_1, 0)$ is the intersection of the x-axis and the tangent of the graph of f at $(x_0, f\ (x_0))$: that is, the improved guess is the unique root of the linear approximation at the initial point. The process is repeated as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Derivative of the given expression is;

$$\frac{dx}{dp} = 3 * p^2 * (1 - \beta^2) + 2 * p * (0.4 * h * \beta^2 - (\sigma * h^2/r^2)) + (\sigma^2 * h^4/3 * r^4)$$

Procedure:

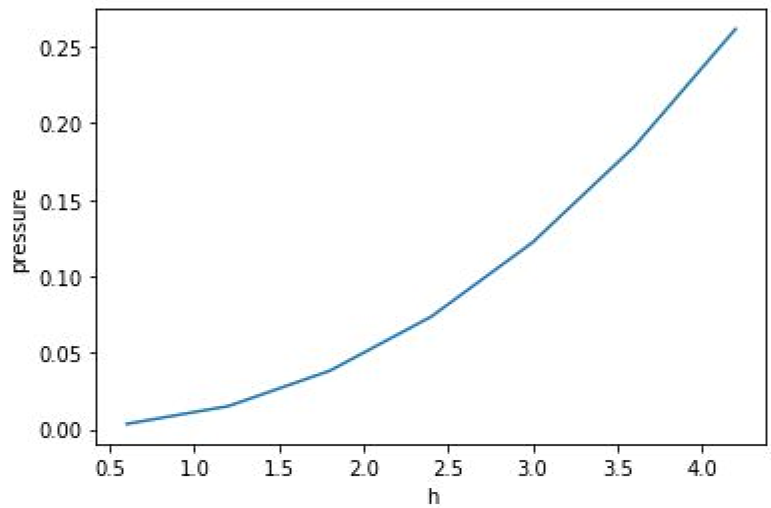
1. Mathematical modules are imported
2. Both the function and function derivative are declared as python functions.
3. All the given parameters are taken.
4. While the absolute value of guess pressure is greater than tol, the value gets appended
5. Plot the values.

Main Program :

```
'''Newton Raphson Technique to find the minimum value of RF'''

import math
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint

#main function
def f(p,h):
    return (pow(p,3)*(1-pow(beta,2)))+(((0.4*h*pow(beta,2))-(sigma*pow(h,2)/pow(r,2)))*pow(p,2))+(((pow(sigma,2)*pow(h,4))/(3*pow(r,4)))*p)-(pow((sigma*pow(h,2)/(3*pow(r,2))),3))
#derivative function
def fdash(p,h):
    return (3*pow(p,2)*(1-pow(beta,2)))+(2*p*((0.4*h*pow(beta,2))-(sigma*pow(h,2)/pow(r,2)))) + ((pow(sigma,2)*pow(h,4))/(3*pow(r,4)))



#thickness of ice field
h=[0.6,1.2,1.8,2.4,3.0,3.6,4.2]
#width of ice field
beta=0.5
#air cushion field
r=40
#tensile strength of ice
sigma=150
```

```
#cushion guess pressure
pguess=80
#tolerance
tol=1e-6
#relaxation factor
alpha=1

#empty array to store the updated pressure values
pressure=[]

iter=1

for i in h:
    while(abs(f(pguess,i))>tol):
        pguess=pguess-alpha*((f(pguess,i)/fdash(pguess,i)))
        iter=iter+1
    pressure.append(pguess)

for j in range(0,6):
    print(h[j],'equal to',pressure[j])




plt.plot(h,pressure)
plt.xlabel('h')
plt.ylabel('pressure')
plt.show()
```

Result:



This plot shows that the pressure increases as the thickness of the frozen part increases.

```
0.6 equal to 0.003562196315848092
1.2 equal to 0.015175461966401868
1.8 equal to 0.038274759255997624
2.4 equal to 0.07366365813035339
3.0 equal to 0.12219868183097073
3.6 equal to 0.18464373030154083
```

The value of pressure at thickness h=0.6, is equal to 0.00356 bar.

No.of iterations taken to find the pressure value for a specified thickness is 38.