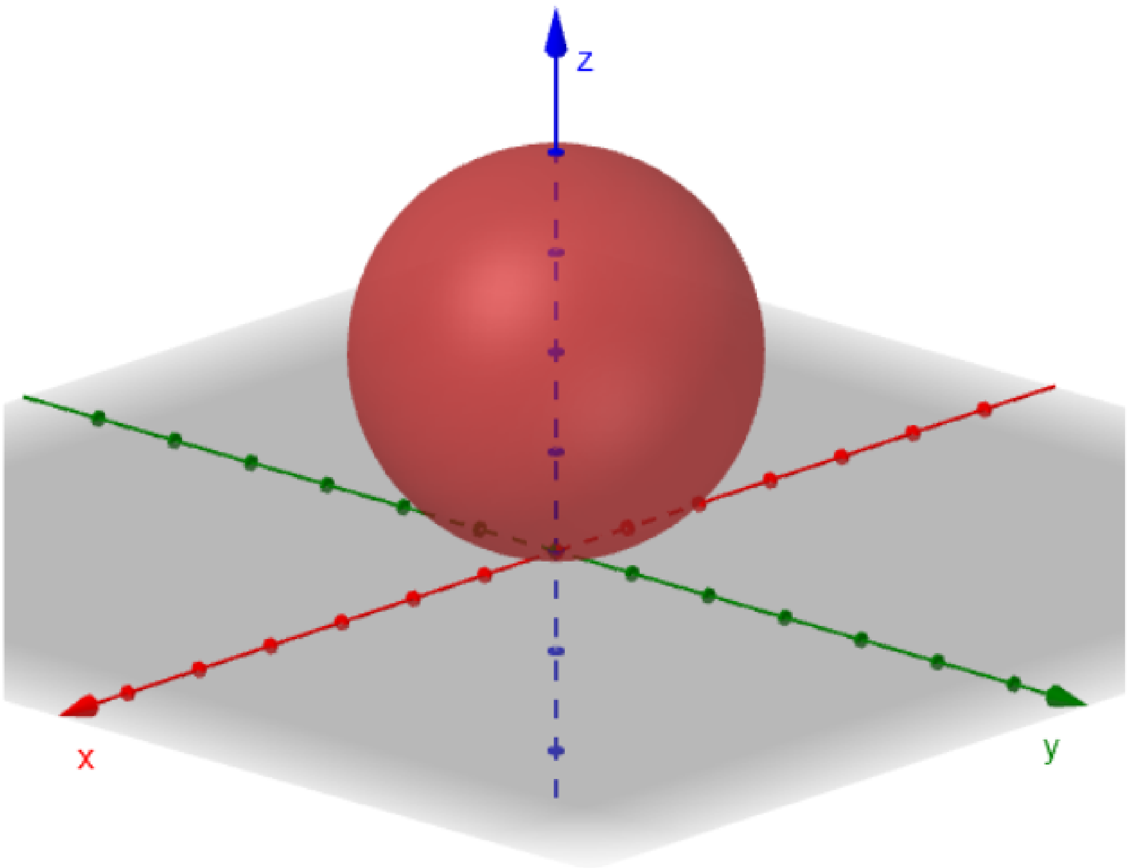
Alunos:
Vinícius de Oliveira Peixoto Rodrigues (245294)
Emerson José Bezerra da Silva (233865)
Fabienne Araujo Santos (215556)

1 Discussão do problema

Comecemos pela equação da densidade. Para chegarmos na massa, queremos resolver a integral $\iiint_V \rho \, \mathrm{dV}$. Porém, ρ não é variável, podendo, portanto, ser tirado da integral, $\rho \iiint_V 1 \, \mathrm{dV}$. Essa integral, porém, se reduz ao volume, temos então ρV. Nos bastará, para conferirmos a fórmula $m = \frac{4\,\pi\,r^3}{3}\,\rho$ que $V = \frac{4\,\pi\,r^3}{3}$. Essa fórmula para o volume poderá ser testada juntamente com a fórmula para o volume da calota esférica, já que a esfera será uma calota esférica com $h = 2\,R$.

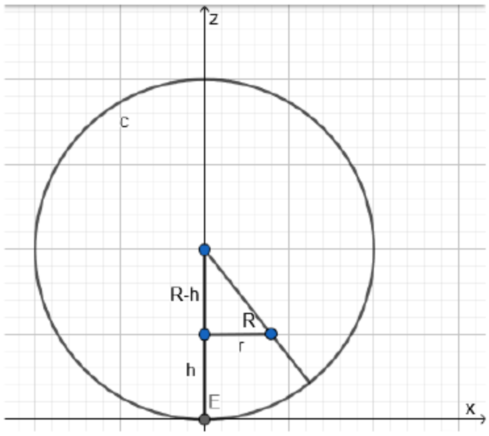
Tomemos a esfera:



Para a fórmula da calota esférica, consideremos um elemento de área da nossa esfera, que será

um cilindro com área da base equivalente ao círculo da esfera quando cortada pelo plano z = h e altura dH.

Teremos o seguinte, num corte pelo plano xz ou yz:



Isso significa que o círculo que será base do nosse elemento de área terá, por Pitágoras, raio:

$$R^2 = (R-h)^2 + r^2$$
$$r = \sqrt{R^2 - (R-h)^2}$$
$$r = \sqrt{R^2 - R^2 + 2\,\mathrm{Rh} - h^2}$$
$$r = \sqrt{2\,\mathrm{Rh} - h^2}$$

Para o caso de h>R, teremos um triângulo similar, mas com cateto h-R ao invés de R-h. Para

a fórmula, que usa apenas o quadrado dessa diferença, isso não irá importar. Nosso elemento de

volume, portanto, será igual a

$$\mathrm{dV} = \pi\, r^2\, \mathrm{dh}$$
$$\mathrm{dV} = \pi\left(\sqrt{2\,\mathrm{Rh} - h^2}\right)^2 \mathrm{dh}$$
$$\mathrm{dV} = \pi\,(2\,\mathrm{Rh} - h^2)\,\mathrm{dh}$$

Para termos o volume, teremos que integrar esse valor do zero até a altura da calota, que será

$$\int_0^h \pi\,(2\,\mathrm{Rh}' - h'^2)\,\mathrm{dh}'$$

Essa integral definida terá como solução

$$V = \pi\left(\mathrm{Rh}^2 - \frac{h^3}{3}\right)$$

Que significa que a fórmula para o volume da calota esférica do enunciado está correta.

Para h = 2R, teremos

$$\pi\left(R\,(2\,R)^2-\frac{(2\,R)^3}{3}\right)$$
$$\pi\left(4\,R^3-\frac{8\,R^3}{3}\right)$$
$$\frac{4\,\pi\,R^3}{3}$$

O que confirma a fórmula para o volume da esfera que buscávamos testar.

Com as fórmulas em mãos, passemos para o problema. Queremos o ponto de equilíbrio da esfera.

Temos da segunda lei de Newton que esse será o ponto em que as forças resultantes se somam.

Existem, atuando na esfera duas forças, o peso, cujo módulo é dado pela multiplicação da massa

por uma constante g e cuja direção será para baixo, e o empuxo, que conforme o princípio de

Arquimedes terá módulo igual ao peso do líquido deslocado e direção para cima. Como o volume

de líquido deslocado será igual ao volume submerso da calota, teremos então o seguinte

$$\mathrm{mg}-V_{\mathrm{desloc}}\,d_{\text{água}}\,g=0$$
$$\frac{4\,\pi\,R^3\,\mathrm{dg}}{3}-\pi\left(\mathrm{Rh}^2-\frac{h^3}{3}\right)d_{\text{água}}\,g=0$$
$$\frac{4\,R^3\,d}{3}-\left(\mathrm{Rh}^2-\frac{h^3}{3}\right)d_{\text{água}}=0$$

Utilizando para a água a densidade de 1 (assumindo aqui unidades convencionais do SI) e para a

esfera o valor dado no enunciado, 0,6, teremos

$$\frac{2,4\,R^3}{3}-\left(\mathrm{Rh}^2-\frac{h^3}{3}\right)=0$$
$$2,4\,R^3-(3\,R\,h^2-h^3)=0$$
$$h^3-3\,\mathrm{Rh}^2+2,4\,R^3=0$$

Agora, queremos encontrar h em fração de R. Convertendo a equação para um sistema de

unidades em que, para comprimentos, R = 1 (ou, equivalentemente, fazendo $h:=h/R$):

$$h^3-3\,h^2+2,4=0$$

Temos então a função para a qual queremos buscar uma raiz. Lembremos que só terão significado físico as soluções em que $0 \leq h \leq 2$, portanto apenas essas nos interessarão.

A derivada dessa função será:

$$3\,h^2 - 6\,h$$

Que tem raiz em 0 e 2. Por conta da derivada ser uma função contínua e negativa em $[0, 2]$, isso significa que, nesse

intervalo, a função ou é estritamente decrescente, não podendo, portanto,

passar duas vezes pelo zero. Já que $0^3 - 3 \times 0^2 + 2,4 = 2,4 > 0$ e $8 - 3 \times 4 + 2,4 = -1,6 < 0$ teremos uma

raiz nesse intervalo, que será por isso única.

2 Métodos numéricos

Dado que a função é um polinômio sua resposta se encontra num intervalo bem definido e sabemos

ser a única raiz nesse intervalo, além de sua derivada ser de cálculo simples, o problema maior

parece ser não a convergência do algoritmo, mas sim a velocidade da convergência.

Foi realizado para esta atividade um conjunto de testes envolvendo quatro métodos numéricos de obtenção de raízes: o método da bissecção, o método do ponto falso (regula falsi), o método do ponto fixo com uma função simples, e o método de Newton-Raphson. Considerou-se que esses algoritmos convergiram a partir do ponto em que a diferença entre duas estimativas sucessivas se tornou menor que 10^{-12}; o erro de cada método foi definido como o módulo da ordenada de cada estimativa e, i.e. $E = |f(e)|$, $f(h) = h^3 - 3h^2 + 2.4$.

Os métodos foram implementados em Python 3, por eliminar a necessidade de se levar em consideração problemas relacionados à precisão de ponto flutuante (fator gerenciado automaticamente pela linguagem, permitido cálculos com precisão arbitrária). O código-fonte pode ser encontrado em https://github.com/nukelets/metodos_numericos/blob/main/root-finding/EstimadorRaizes.py.

2.1 Bissecção

A implementação do algoritmo é relativamente simples:

```
def bisseccao(f, a, b, erro):
    if a > b or f(a)*f(b) > 0:
        # valores inválidos de a e b
        return float("NaN"), 0
```

```
        mid = a
        iteracoes = 0
        while (b-a) > erro:
            iteracoes += 1
            mid = (a+b)*0.5
            # mesmo sinal
            if f(a)*f(mid) > 0:
                a = mid
            else:
                b = mid

        return mid, iteracoes
```

Para esse algoritmo, encontrou-se uma estimativa de **1.134137845704572**, após **41 iterações**, com erro da ordem de 10^{-13}.

2.2 Ponto falso

Abaixo se encontra a implementação do algoritmo:

```
def ponto_falso(f, a, b, erro):
        if a > b or f(a)*f(b) > 0:
            # valores inválidos de a e b
            return float("NaN"), 0

        c = (a*f(b) - b*f(a))/float(f(b) - f(a))
        iteracoes = 0

        while abs(f(c)) > erro:
            iteracoes += 1
            if abs(f(a)) < erro:
                return a, iteracoes
            elif abs(f(b)) < erro:
                return b, iteracoes

            if f(c)*f(a) > 0:
                a = c
            else:
                b = c

            c = (a*f(b) - b*f(a))/float(f(b) - f(a))
            if b-a < erro:
                return 0.5*(a+b), iteracoes

        return c, iteracoes
```

Com esse método se chegou a uma estimativa de **1.1341378457045543**, após apenas **6** iterações, com erro da ordem de 10^{-13}.

2.3 Ponto fixo

A implementação do algoritmo se encontra abaixo:

```
def ponto_fixo(f, a, b, erro):
        iteracoes = 0
        x = 0.5*(a+b)
        proximo_x = f(x)

        while (abs(proximo_x - x) > erro):
            x = proximo_x
            proximo_x = f(x)
            iteracoes += 1

        return proximo_x, iteracoes
```

Para esse método, obteve-se a função a ser iterada fazendo-se

$$h^3 - 3\,h^2 + 2,4 = 0 \Rightarrow h = \frac{1}{3}\sqrt{h^3 + 2.4} = f(h)$$

Os resultados não foram muito bons: estimativa de **1.1341378457036264**, após **44** iterações, com erro da ordem de 10^{-12}.

2.4 Newton-Raphson

Para este algoritmo, foi definida a derivada como

```
def derivada(f, x, h=0.001):
    return (f(x+h) - f(x))/h
```

A partir disso, definiu-se a função $g(x) = x - \frac{f(x)}{f'(x)}$, que foi fornecida como argumento para o método do ponto fixo da seção anterior:

```
def newton_raphson(f, a, b, erro):
    g = lambda x: x - (f(x)/derivada(f, x))
    return ponto_fixo(g, a, b, erro)
```

Com este método, foi obtida a estimativa de **1.1341378457045364**, após somente **4** iterações, com erro da ordem de 10^{-15}.

3 Resultados

Na tabela abaixo se encontram distribuídos os resultados do conjunto de testes:

Método	**Estimativa**	**Erro**	**Iterações**
Bissecção	1.134137845704572	10^{-13}	41
Ponto falso	1.134137845704572	10^{-13}	6
Ponto fixo	1.1341378457036264	10^{-12}	44
Newton-Raphson	1.1341378457045364	10^{-15}	4

Esses resultados indicam que o método de Newton-Raphson foi não somente o mais preciso, mas também o mais rápido, convergindo após apenas 4 iterações. Por essa razão, foi considerado como solução da equação o resultado desse método, de modo que $h \approx 1.1341378457045364$, indicando que $\boldsymbol{h \approx 1.13R}$.