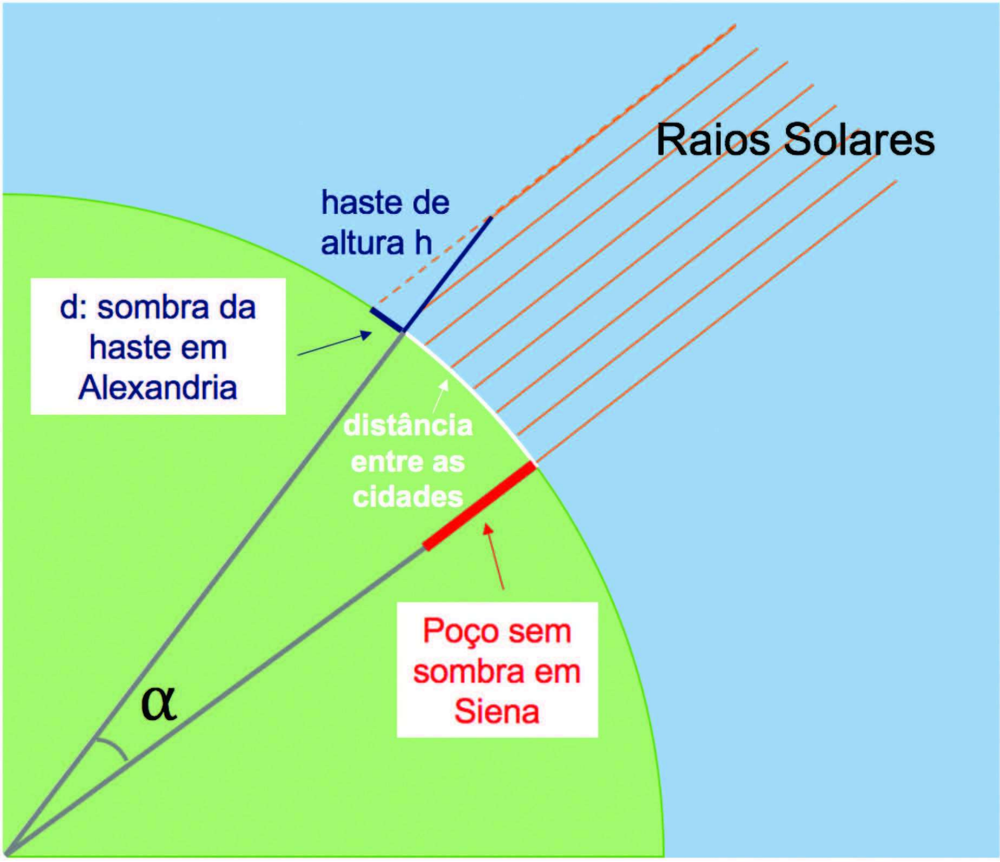
Eratóstenes de Alexandria (c. 276 a.C. – c. 194 a.C.) foi um dos grandes pensadores da Antiguidade, conhecido por sua atuação como matemático, astrônomo, geodesista, filósofo e poeta grego. Nascido em Cirene (atualmente Líbia), trabalhou durante boa parte da vida em Alexandria, no Egito, onde foi diretor da famosa Biblioteca de Alexandria.

Eratóstenes é mais lembrado por ter realizado uma das primeiras medições precisas do tamanho da Terra. Para isso, utilizou um método geométrico baseado na observação da posição do Sol em duas cidades diferentes:



Em Siena (atual Assuã), o Sol estava exatamente a pino ao meio-dia do solstício de verão. Já em Alexandria, no mesmo momento, o Sol formava um certo ângulo em relação ao zênite.

Conhecendo a distância entre as duas cidades, Eratóstenes aplicou princípios da geometria para estimar a circunferência da Terra — e obteve um valor surpreendentemente próximo do real.

Para medir a distância entre as duas cidades, Eratóstenes baseou-se em cinco itinerários de viagem, que estimavam a distância entre elas como sendo: 785 km, 785,5 km, 787,5 km, 789,5 km e 789 km.

Para medir a sombra do Sol e a altura da torre, Eratóstenes utilizou uma corrente de agrimensor, na qual cada elo correspondia a 10 cm. Com esse instrumento, ele encontrou 30 elos de sombra e 238 elos de altura da torre.

Sabendo que a circunferencia da Terra no equador é de 40.075 Km, e dos seus conhecimentos em propagação de erros, determine se a atual circunferencia da terra esta dentro do intervalo de 1 sigma do valor determinado por Erastostenes em 200 a.c.

Sendo assim o modelo matemático será

$$\text{circunferencia} = d \cdot \left(\frac{2\pi}{\alpha}\right)$$

```
import numpy as np
```

Primeiro passo calcular a média das distancias e o desvio padrão

```
# Lista de distâncias (exemplo)
distancias = np.array([787, 787.2, 787.3, 788, 788.1])*1000

# Cálculo da média
dist_m = np.mean(distancias)

# Cálculo do desvio padrão
dp_m = np.std(distancias)

print(f"Média: {dist_m:.2f}")
print(f"desvio padrão: {dp_m:.4f}")
```

```
Média: 787520.00
desvio padrão: 444.5222
```

Calcular o angulo da projeção da sombra e sua devida precisão

$$\alpha = \arctan\left(\frac{\text{co}}{\text{ca}}\right)$$

Derivada Parcial em relação a co

$$\frac{\partial \alpha}{\partial \text{co}} = \arctan\left(\frac{\text{co}}{\text{ca}}\right)$$

$$\frac{\partial \alpha}{\partial \text{co}} = \frac{\text{ca}}{\text{co}^2 + \text{ca}^2}$$

Derivada parcial em relação a ca

$$\frac{\partial \alpha}{\partial \text{co}} = \arctan\left(\frac{\text{co}}{\text{ca}}\right)$$

$$\frac{\partial \alpha}{\partial \text{ca}} = -\frac{\text{co}}{\text{co}^2 + \text{ca}^2}$$

Dado o modelo matemático de propagação de variâncias e covariâncias que relacionam as medidas com as grandezas calculadas

$$\Sigma_y = \mathbf{D}\Sigma_x\mathbf{D}^T$$

Onde

$$\mathbf{D}$$

é a matriz Jacobiana das derivadas parciais em relação às observações

E

$$\Sigma_x$$

é a matriz da covariancia das observações.

```
# os valores do cateto oposto e cateto adjacente
co =3
ca =23.8

dp =5/100

alpha=np.arctan(co/ca)

G= np.matrix([ca/(co**2+ca**2),-co/(co**2+ca**2)])

Sy= np.matrix([[dp**2,0],[0,dp**2]])

Sigma_alpha = G*Sy*G.T

print(f"Média: {np.degrees(alpha):.2f}")
print(f"Desvio padrão: {np.sqrt(np.degrees(Sigma_alpha[0,0])):.4f}")
```

```
Média: 7.18
Desvio padrão: 0.0158
```

Portanto seguimos para a a primeira equação

$$\text{circunferencia} = d \cdot \left(\frac{2\pi}{\alpha}\right)$$

e relizamos as derivadas pasciais em relação as observações

Derivadas parciais:

$$\frac{\partial C}{\partial d} = \frac{2\pi}{\alpha} \quad \text{e} \quad \frac{\partial C}{\partial \alpha} = -\frac{2\pi d}{\alpha^2}$$

```
D= (dist_m*2*np.pi/alpha)

G2=np.matrix([(2*np.pi/alpha), (-2*np.pi/alpha**2)])
Sy2=np.matrix([[dp_m**2, 0],[0,(Sigma_alpha[0,0])]])

Sigma_circunferencia = G2*Sy2*G2.T

print(f"circunferencia: {D/1000:.2f}")
print(f"Desvio padrão: {np.sqrt(Sigma_circunferencia[0,0])/1000:.4f}")

print(f"Máximo: {np.sqrt(Sigma_circunferencia[0,0]/1000) + D/1000:.4f}")
print(f"Mínimo: { D/1000 -np.sqrt(Sigma_circunferencia[0,0]/1000):.4f}")
```

```
circunferencia: 39462.23
Desvio padrão: 22.2748
Máximo: 40166.6188
Mínimo: 38757.8378
```