Research Methods

Robert E Simpson

Singapore University of Technology & Design

robert_simpson@sutd.edu.sg

October 10, 2018

In Week 5 Content

You should be able to:

- Understand how finite and central differences can be used to numerically differentiate
- Apply the perturbation method using python to analyse the propagation of errors
- Perform convergence test to check the stability of the perturbation method

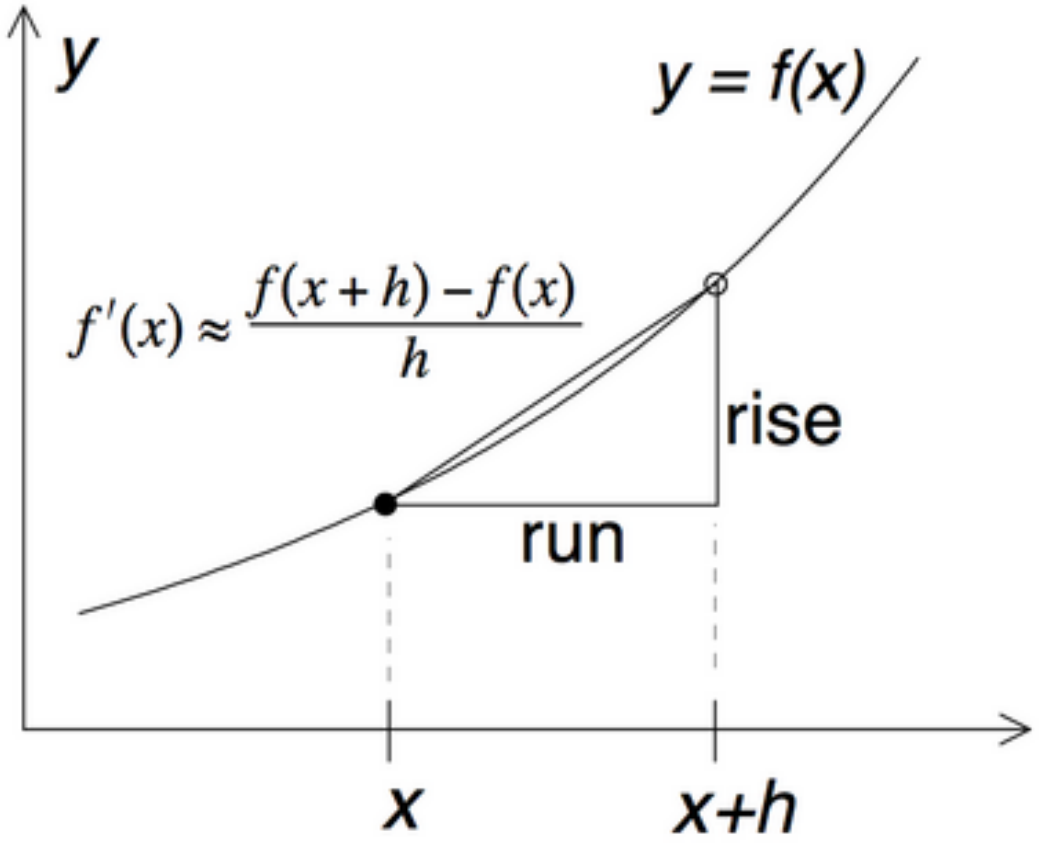
Perturbation Approach to PoE

Numerically solve:

$$\boxed{\sigma_x^2 = \sigma_u^2 \left(\frac{\partial x}{\partial u}\right)^2 + \sigma_v^2 \left(\frac{\partial x}{\partial v}\right)^2} \tag{1}$$

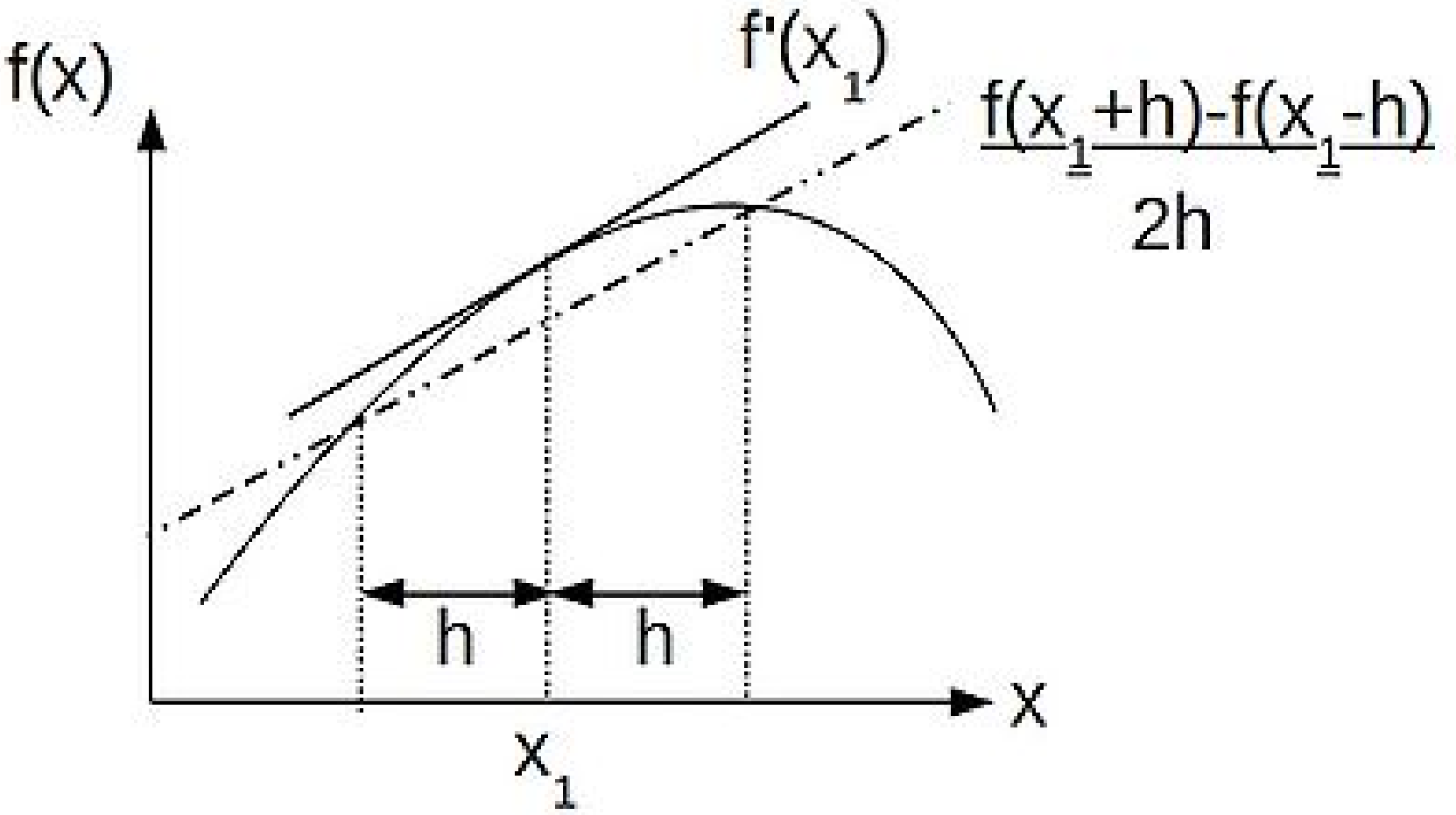
We can numerically calculate $\frac{\partial x}{\partial u}$ and $\frac{\partial x}{\partial v}$

Finite Differences



$$f(x+h) \approx f(x) + \frac{f'(x)}{1!}h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 \quad (2)$$

Approach: Central Differences



Central Differences

$$f(x_1 + h) \approx f(x_1) + \frac{f'(x_1)}{1!}h + \frac{f''(x_1)}{2!}h^2 + \frac{f'''(x_1)}{3!}h^3 \tag{3}$$

$$f(x_1 - h) \approx (x_1) - \frac{f'(x_1)}{1!}h + \frac{f''(x_1)}{2!}h^2 - \frac{f'''(x_1)}{3!}h^3 \tag{4}$$

$$f(x_1 + h) - f(x_1 - h) \approx 2f'(x_1)h + 2\frac{f'''(x_1)}{3!}h^3 \tag{5}$$

$$\frac{f(x_1 + h) - f(x_1 - h)}{2h} - \frac{f'''(x_1)}{3!h}h^3 \approx f'(x_1) \tag{6}$$

$$\frac{f(x_1 + h) - f(x_1 - h)}{2h} - \frac{f'''(x_1)}{6}h^2 \approx f'(x_1) \tag{7}$$

$$\frac{f(x_1 + h) - f(x_1 - h)}{2h} + error \approx f'(x_1) \tag{8}$$

Therefore the expansion to the first derivative gives an error of order h^2 or $O(h^2)$. This is better than finite difference method, where the error is O(h)

PoE & Central Differences

There are no rules for the size of the perturbation or step size, h, since the error depends on the underlying the function.
Typically $h = 1$ to 4% is reasonable. The error stability should be checked using a few different step sizes.

Useful python functions

```
numpy.logspace(-6,-1,num=60)
```

Similar to the `numpy.linspace` function, but produces a list on a logrithmic scale. The above example starts at 10^{-6} and ends at 10^{-1} with 60 points separated on a $\log_{10}$ scale.

Case Problem 5.4

The boat question again

Use the perturbation approach to alculate the error in the time for a boat to move 30° around a curve of radius 10 m, moving at a speed of 0.6 ms^{-1}. Assume that the captain is able to estimate angle with an error of 5°, radius with an error of 0.1 m, and velocity with an error of 0.1 ms^{-1}.

$$\sigma_t^2 = \sigma_u^2 \left(\frac{\partial t}{\partial r}\right)^2 + \sigma_v^2 \left(\frac{\partial t}{\partial v}\right)^2 + \sigma_\theta^2 \left(\frac{\partial t}{\partial \theta}\right)^2$$

In equation 7, h is the perturbation. Usually h is a few percent of the actual value.

Case Problem 5.5

Exponential growth and uncertainty

The consumption of resources is modelled as:

$$Q(t) = \int_0^t P_0 e^{rt} dt = \frac{P_0}{r}(e^{rt} - 1) \tag{9}$$

Where P_0 is the initial consumption rate, and r is the exponential rate of growth. The world coal consumption in 1986 was equal to 5.0 billion tonnes per a year and the estimated recoverable reserves of coal were estimated at 1000 billion tonnes. The consumption rate, r, is 2.7%
Use the perturbation method to compute the standard error in the estimated time before the coal reserves are depleted, σ_t.
You may assume that the growth rate, r, and the recoverable reserves, Q, are subject to random uncertainty with $\sigma_r = 0.2$ % (absolute) and $\sigma_Q = 10\%$ (relative) respectively.

PoE Summary

We have now looked at 3 different methods to investigate the propagation of error:

- Analytically solving the PoE
- Numerically -using the Monte Carlo Approach
- Numerically -using the Perturbation Approach

CP5.4 Solution

Boat Perturbation PoE Example

$$t = \frac{r\theta}{v} \tag{10}$$

$$\frac{\partial t}{\partial r} \approx \frac{t(r + h_r, v, \theta) - t(r - h_r, v, \theta)}{2h_r} \tag{11}$$

$$\frac{\partial t}{\partial v} \approx \frac{t(r, v + h_v, \theta) - t(r, v - h_v, \theta)}{2h_v} \tag{12}$$

$$\frac{\partial t}{\partial \theta} \approx \frac{t(r, v, \theta + h_\theta) - t(r, v, \theta - h_\theta)}{2h_\theta} \tag{13}$$

CP5.4 Solution

```
import numpy as np
import matplotlib.pyplot as plt

def boat(vel,ang,rad):
    return (ang*np.pi/180)*rad/vel;

def centdiff(vel,ang,rad,Evel,Eang,Erad, h): #
    calculate POE using a Talylor Approximation
    RE_v=h*vel
    RE_a=h*ang
    RE_r=h*rad #relative error
    dtdv=(boat(vel+RE_v,ang,rad)-boat(vel-RE_v,ang,rad
        ))/(2*RE_v)
    dtda=(boat(vel,ang+RE_a,rad)-boat(vel,ang-RE_a,rad
        ))/(2*RE_a)
    dtdr=(boat(vel,ang,rad+RE_r)-boat(vel,ang,rad-RE_r
        ))/(2*RE_r)
    SqErr=pow(Evel*dtdv,2) + pow(Eang*dtda,2) + pow(
        Erad*dtdr,2)
    return np.sqrt(SqErr)
```

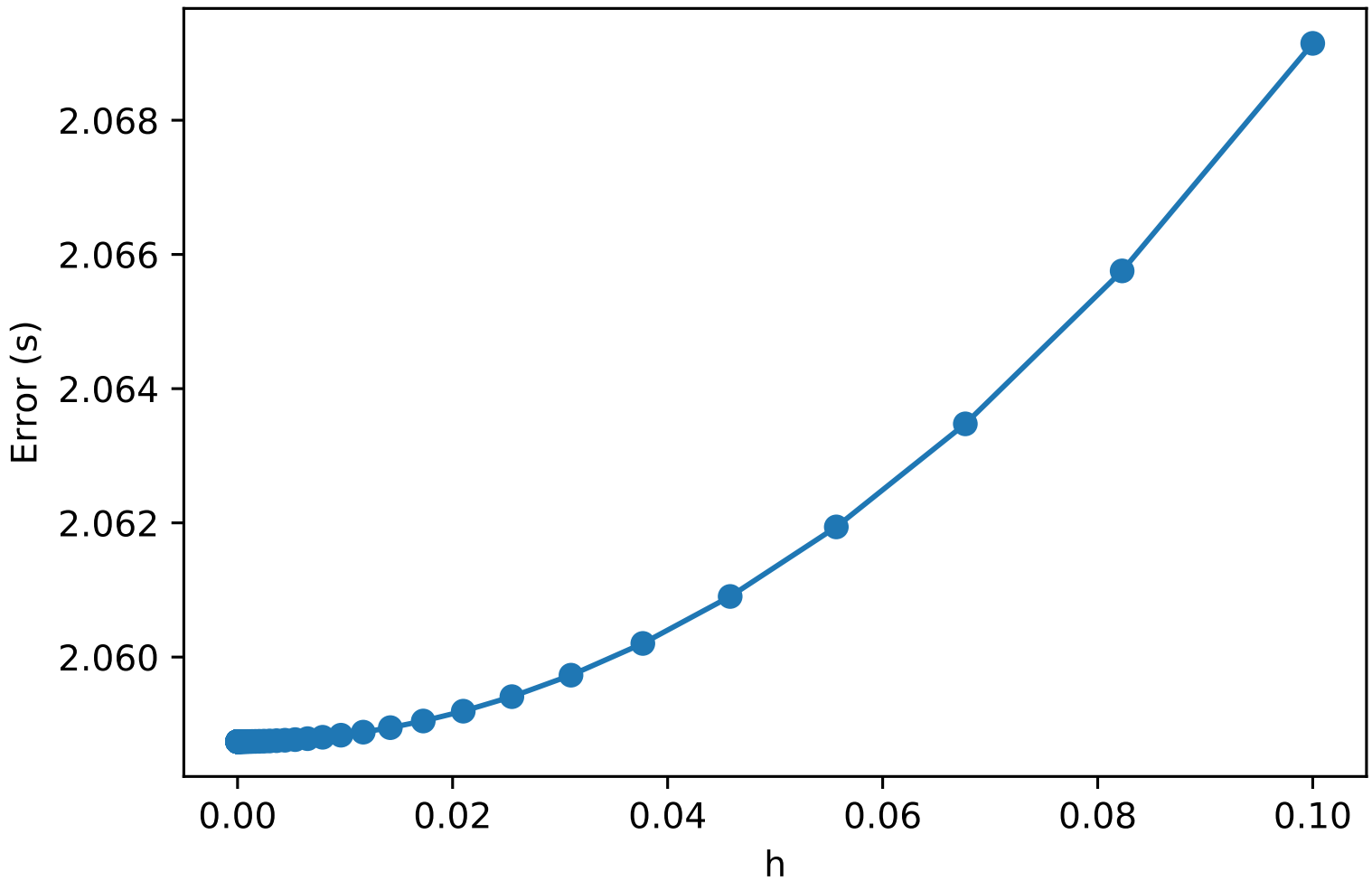
CP5.4 Solution

```
def main():
    vel=0.6
    ang=30
    rad=10
    Evel=0.1
    Eang=5
    Erad=0.1

    h=np.logspace(-6,-1,num=60)
    print h
    poe=[]
    for dh in h:
        poe.append(centdiff(vel,ang,rad,Evel,Eang,Erad
            , dh))

    plt.figure()
    plt.plot(h,poe, marker='o')
    plt.xlabel('h')
    plt.ylabel('Error (s)')
    plt.savefig('W5CP1_boat.pdf')
```

5.4 Convergence



Analytical Result

$$\sigma_t = \sqrt{(7.52 \times 10^{-3} + 2.104 + 2.115)}$$

$$\sigma = 2.055 \text{ sec}$$

CP5.5 Solution

Exponential growth and uncertainty

This PoE problem can be solved analytically but for fun we are going to solve it using a perturbation method (by brute force).

$$t = \ln\left(\frac{Qr}{P_0} + 1\right) r^{-1}$$

P_0= 5 billion tonnes/year
$r = 0.027$ and $\sigma_r = 0.002$
$Q = 1000$ billion tonnes and $\sigma_Q = 100$ billion tonnes

Calculate the error in the estimate of time remaining until the world's coal resources are depleted using the perturbation method. Check that your error is properly converged. Express your estimate a tolerance set to 95% confidence limits.

CP5.5 Solution

Suggested Implementation Method

1. Define a function called "Coal" to compute:
   $t = \ln\left(\frac{Qr}{P_0} + 1\right) r^{-1}$
2. Define a new function to do the Central Differences.
   - Define the step size h. (i.e. ΔQ and Δr as one one or two percent of Q and r)
   - Call the "Coal" function and compute $\frac{\partial t}{\partial r} \approx \frac{t(r+\Delta r, Q) - t(r-\Delta r, Q)}{2\Delta r}$ and $\frac{\partial t}{\partial Q} \approx \frac{t(r, Q+\Delta Q) - t(r, Q-\Delta Q)}{2\Delta Q}$
   - Return the PoE: $\sigma_t = \left[\sigma_r^2 \left(\frac{\partial t}{\partial r}\right)^2 + \sigma_Q^2 \left(\frac{\partial t}{\partial Q}\right)^2\right]^{1/2}$
3. Make a for-loop to check the convergence of the central differences with perturbation step size.

CP5.5 Solution

```
def coalt(Q,r,P0):
    return (np.log((Q*r/P0)+1))/r

def centdiffCP2(Q,r,P0,EQ,Er,h): #calculate POE using
    a Talylor Approximation
    RE_Q=h*Q
    RE_r=h*r

    dtdQ=(coalt(Q+RE_Q,r,P0)-coalt(Q-RE_Q,r,P0))/(2*
        RE_Q)
    dtdr=(coalt(Q,r+RE_r,P0)-coalt(Q,r-RE_r,P0))/(2*
        RE_r)

    SqErr=pow(EQ*dtdQ,2) + pow(Er*dtdr,2)
    return np.sqrt(SqErr)
```

CP5.5 Solution

```
def main2():
    Q=1000
    r=0.027
    P0=5
    EQ=100
    Er=0.002
    h=np.logspace(-6,-1,num=60)
    print h
    poe=[]
    for dh in h:
        poe.append(centdiffCP2(Q,r,P0,EQ,Er,dh))

    plt.figure()
    plt.plot(h,poe, marker='o')
    plt.xlabel('h')
    plt.ylabel('Error (Years)')
    plt.savefig('W5CP2_coal.pdf')
```

CP5.5 Solution