

# 數值方法

## 作業一

資工三乙 406262319 黃育皓

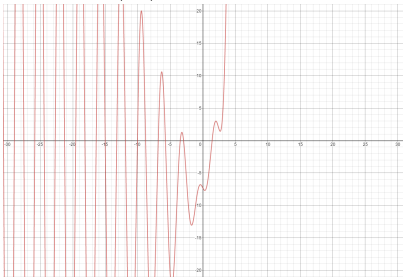
### 程式架構

- 每個方法各自一個 py 檔
- AllMethod.py 中將方程式套用各個方法去進行
- main.py 中定義方程式並操作
- 各個方程式與各個方法的結果儲存在 data 資料夾中
- 執行 main.py 即可

### 方程式與圖形

A

$$e^x - 3x\cos(2x) - 8.3$$



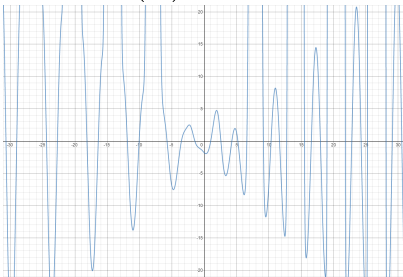
- 紀錄之平均值

Method	誤差	執行次數
Bisection	0.0000000150	30
FalsePosition	0.0000000007	10
ModifyFalsePosition	0.0000000813	30
Secant	0.0000000000	10
Newton	0.1571701275	上限
FixPoint	inf	5

- 以該方程式來看 FalsePosition 與 Secant 這二種方式算出的值最精確且快速
- Newton 與 FixPoint 則幾乎無法得出正確解

B

$$e^{x\sin x} - x\cos(2x) - 2.8$$

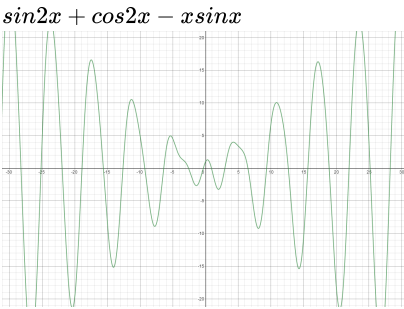


- 紀錄之平均值

Method	誤差	執行次數
Bisection:	0.0000000097	30
FalsePosition:	0.0000000005 or 0.0000030000	10 or 2000
ModifyFalsePosition:	0.0000000093	30
Secant:	0.0000000000	10
Newton:	0.0000002684	50
FixPoint:	inf	5

- 以該方程式來看 Secant 為最佳解
- FixPoint 無法得出解
- FalsePosition 有一半是得到快速且正確的解另一半慢了一些且正確率下降
- 在這個方程式中 ModifyFalsePosition 的結果就比 FalsePosition 來的好很多了

C

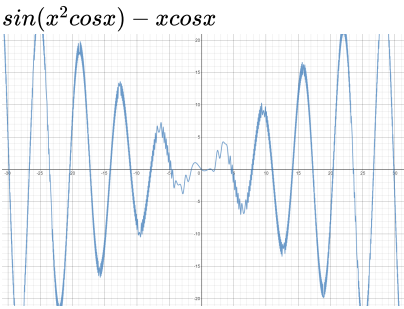


- 紀錄之平均值

Method	誤差	執行次數
Bisection:	0.0000000032	30
FalsePosition:	0.0000000001	8
ModifyFalsePosition:	0.0000000100	30
Secant:	0.0000000000	10
Newton:	0.0000000000	5
FixPoint:	2.5884 or 2.1944	上限

- 以該方程式來說 FalsePosition , Secant 與 Newton 為最佳解
- FixPoint 從結果上看其在2點之間不斷跳來跳去

D



- 紀錄之平均值

Method	誤差	執行次數
Bisection:	0.0000000100	30
FalsePosition:	0.0000000030	8
ModifyFalsePosition:	0.0000000010	30
Secant:	0.0000000000	10
Newton:	0.0000 or 2.xxxx	5 or 上限
FixPoint:	0.0000000000	10

- 以該方程式來說 `FalsePosition` , `Secant` 與 `FixPoint` 為最佳解
- `Newton` 有一半為最佳解另一半則會跑不出來

結論

Bisection

- 在每個方程式中都可以計算出誤差不大的解
- 執行的次數約莫都在30次左右
- 穩定的方法

FalsePosition

- 在每個方程式中都可以計算出誤差不大的解
- 執行的次數約莫都在10次左右
- 可能因初始值的變動導致至誤差與次數增加許多

ModifyFalsePosition

- 在每個方程式中都可以計算出誤差不大的解
- 執行的次數約莫都在30次左右
- 雖次數較多但比 `FalsePosition` 穩定

Secant

- 在每個方程式中都幾乎沒有誤差
- 執行的次數約莫都在10次左右
- 穩定又快速的一個方法



Newton

- 會因方程式的不同而導致可能沒結果或是結果誤差過大
- 但若是其可正常執行時將會是最快速且正確的方法



FixPoint

- 會因方程式的不同而導致可能沒結果
- 但若其可正常執行時誤差是最小的