HW1:

Taylor Series of sin(x)

- $sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \quad |x| < \infty$
- $S_1 = x$
- $S_2 = x - \frac{x^3}{6}$
- $S_3 = x - \frac{x^3}{6} + \frac{x^5}{120}$
- Reference to s1.m, s2.m s3.m and sinplot.m
  - https://web.ma.utexas.edu/CNA/NMC7/nmc7-matlab.html

Hung-Jui Chang | Numerical AnalysisIntroduction

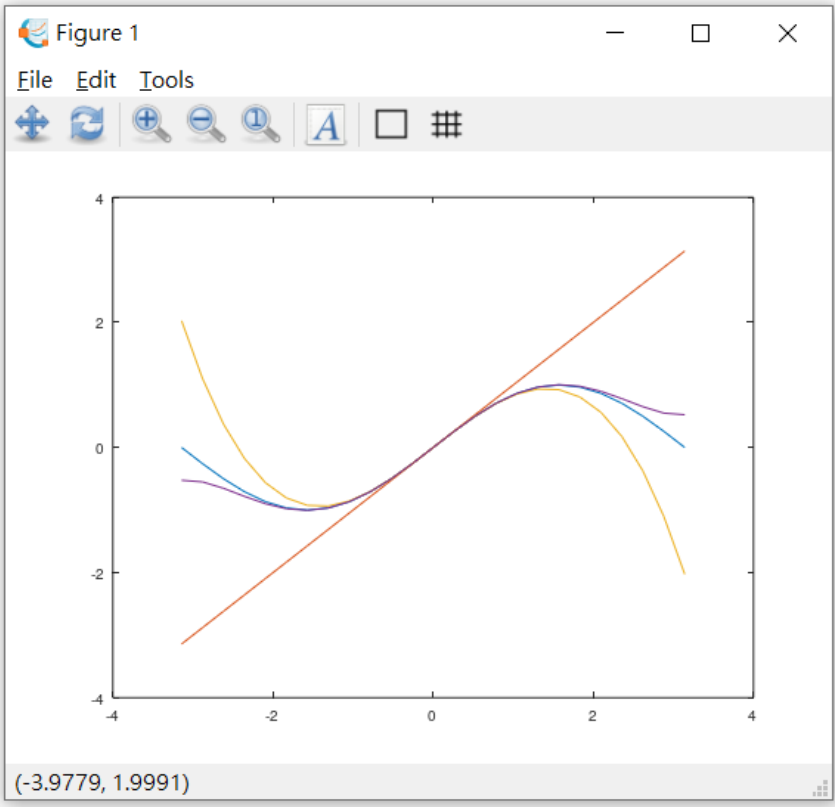
```
25 function Y1 = s1 (X)
26   Y1=X;
27 endfunction
```

```
25 function Y2 = s2 (X)
26   Y2=X-((X.^3)/6);
27 endfunction
```

```
25 function Y3 = s3 (X)
26   Y3=X-((X.^3)/6)+((X.^5)/120);
27 endfunction
```

Command Window

```
GNU Octave, version 7.2.0
Copyright (C) 1993-2022 The Octave Project Developers.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type 'warranty'.

Octave was configured for "x86_64-w64-mingw32".

Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

>> edit s1.m
>> edit s2.m
>> edit s3.m
>> X=-pi:pi/12:pi;
>> Y=sin(X);
>> Y1=s1(X);
>> Y2=s2(X);
>> Y3=s3(X);
>> plot(X,Y)
>> hold on
>> plot(X,Y1)
>> plot(X,Y2)
>> plot(X,Y3)
>>
```



HW2:

Programming Exercise

- Write a function with 3 parameters: x, b, n
- Show the decimal number x representing in the base b with n fractional digits
  - b may consider only 2, 8 and 16.
- Example: $x = 10.125$, $b = 2$, $n = 3 \Rightarrow (1010.001)_2$
- You may consider the integer part and fractional part separately.


```
26 function [C,D]= baseconvert (x, b, n)
27   a=floor(x);
28   f=x-a;
29   i=1;
30   C(1)=0;
31
32   while(a)
33   C(i)=mod(a,b);
34   a=(a-C(i))/b;
35   i++;
36 endwhile
37 C=flip(C);
38
39  for i =1:n
40    D(i)=floor(f*b);
41    f=f*b-floor(f*b);
42   endfor
43
44 endfunction
```

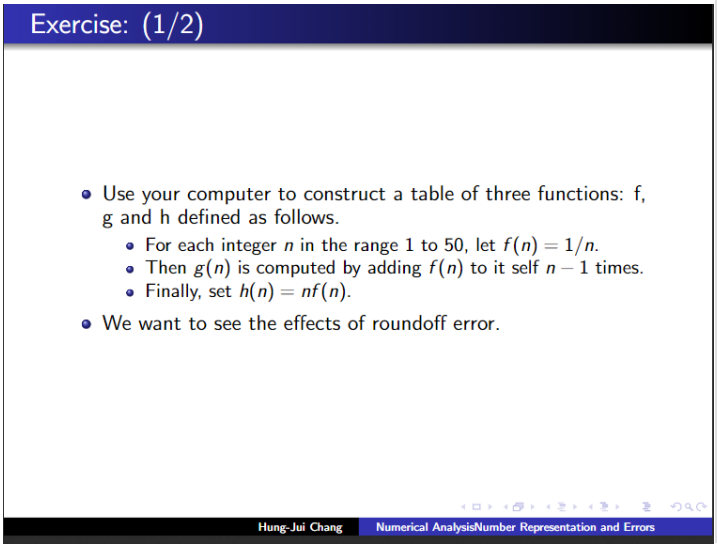
Command Window

```
GNU Octave, version 7.2.0
Copyright (C) 1993-2022 The Octave Project Developers.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type 'warranty'.

Octave was configured for "x86_64-w64-mingw32".

Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

>> [C,D]=baseconvert(10.125,2,3)
C =

   1   0   1   0

D =

   0   0   1

>>
```

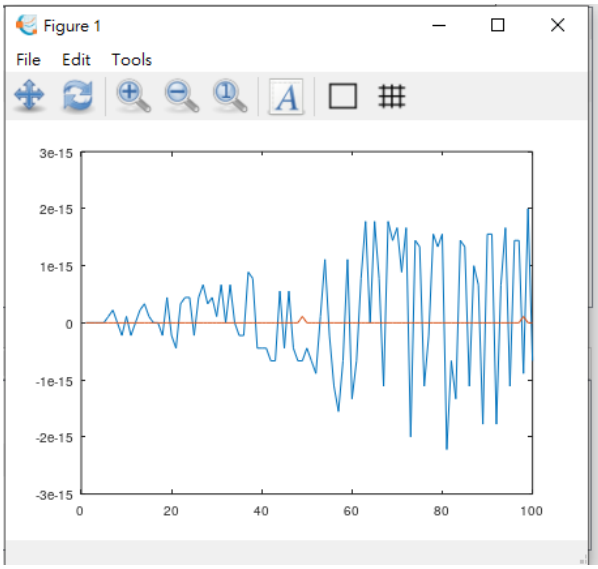
HW3:

(1) 請實作完成投影片Chapter 2, Page 14的練習作業。請完成function [f, g, h] = fgh(n)，其中 回傳值f是每個1/n的值，g是將n個1/n相加後的結果，h是n乘上1/n後的結果。請畫圖比較n從1到100時1-g和1-h的結果，並將結果存為fgh.png後上傳 (2) 請實做投影片chapter 2 Page 15計算Euler constant的程式。請實作並上傳 function gamma = eulerconst(k)計算 n=k時計算出的Euler constant。然後畫出k從1到100的Euler constant的變化圖 euler.png後上傳。

HW3-1:




```
25 function [f,g,h] = fgh (n)
26   #回傳值f是每個1/n的值
27   #g是將n個1/n相加後的結果
28   #h是n乘上1/n後的結果
29   #請畫圖比較n從1到100時1-g和1-h的結果
30   #並將結果存為fgh.png後上傳
31   for i=1:n
32     f(i)=1/i;
33     g(i)=0;
34     for j=1:i
35       g(i)=g(i)+f(i);
36     endfor
37     h(i)=i*f(i);
38   endfor
39 endfunction
```




```
This is free software; see the sour
There is ABSOLUTELY NO WARRANTY; no
FITNESS FOR A PARTICULAR PURPOSE.

Octave was configured for "x86_64-w

Additional information about Octave

Please contribute if you find this
For more information, visit https:/

Read https://www.octave.org/bugs.ht
For information about changes from

>> edit fgh.m
>> [f,g,h] = fgh (100);
>> plot([1:100],1-g,[1:100],1-h)
>>
```

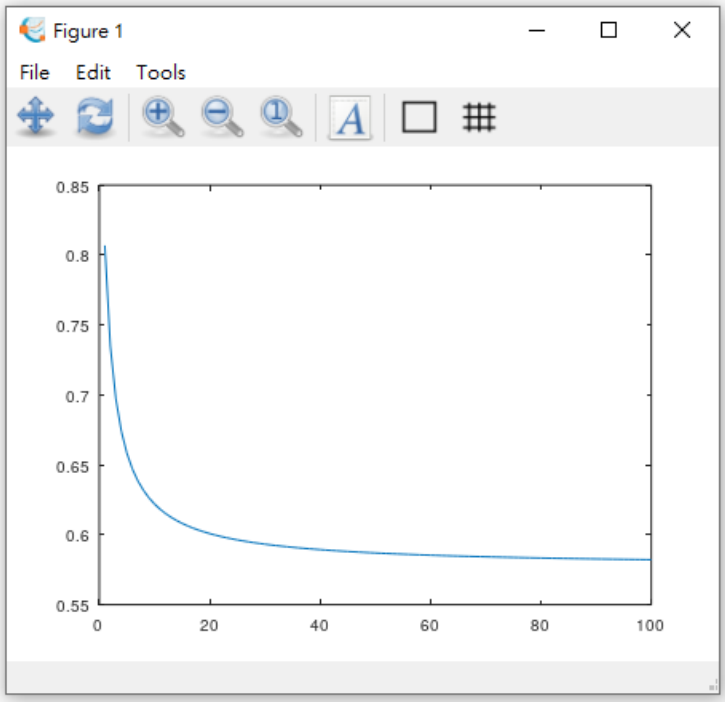
HW3-2:

Excercise: (2/2)

- The harmonic series $1+\frac{1}{2}+\frac{1}{3}+\ldots+$ is known to diverge to ∞. The n-th partial sum approaches ∞ at the same rate as $\ln(n)$.
- Euler's constant is defined to be $\gamma = \lim_{n\to\infty}[\sum_{k=1}^{n}\frac{1}{k} - \ln(n)] \approx 0.5772$
- Consider the pseudocode:
  - real s, x
  - x←1.0
  - s←1.0
  - repeat
    - x← $x+1.0$
    - s← $s+1.0/x$
  - end repeat
- If the loop repeats n times, calculate the value of $s-\ln(n)$
  - Draw a figure with x as n, y as $s-\ln(x)$


```
25 function gamma = eulerconst (k)
26   s=1;
27   x=1;
28   for i=1:k
29     x=x+1;
30     s=s+1/x;
31     gamma(i)=s-log(x);# ln x
32   endfor
33 endfunction
```

```
s = 4.9403
s = 4.9530
s = 4.9655
s = 4.9778
s = 4.9900
s = 5.0021
s = 5.0140
s = 5.0257
s = 5.0374
s = 5.0489
s = 5.0602
s = 5.0715
s = 5.0826
s = 5.0936
s = 5.1044
s = 5.1152
s = 5.1258
s = 5.1363
s = 5.1468
s = 5.1571
s = 5.1673
s = 5.1774
s = 5.1874
s = 5.1973
>>
>>
>>
>> edit eulerconst.m
>> a=eulerconst(100);
>> plot(1:100,a)
>>
```



Command Window | Documentation | Variable Editor | Editor

Profiler

HW4-1:

Bisection1(f, a, b, n_max)

- integer n, n_max
- real a, b, c, u, v, w
- u ← f(a)
- v ← f(b)
- output a, b, u, v
- if $uv \geq 0$ then stop
- for $n = 0$ to n_max do
  - c ← $\frac{a+b}{2}$
  - w ← $f(c)$
  - output n, c, w
  - if $wu = 0$ then exit loop
  - else if $wu < 0$ then
    - b ← c
    - v ← w
  - else
    - a ← c
    - u ← w
  - end if
- end for

```
26 function x= bisection ( x, y, n_max)
27   #考慮f(x) = (x-a)(x^2+x+(b+c))，其中abc是你學號的末三碼。
28   #請分別撰寫用二分搜尋法及牛頓法找到這個函數的解的程式。
29   a=1;
30   b=0;
31   c=7;
32   u=(x-a)*(x^2+x+(b+c));
33   v=(y-a)*(y^2+y+(b+c));
34   if u*v>=0
35     exit;
36   endif
37   for n=0:n_max
38     z=(x+y)/2;
39     w=(z-a)*(z^2+z+(b+c));
40     if w*u==0
41       break
42     endif
43     if w*u<0
44       y=z;
45       v=w;
46     endif
47     if w*u>0
48       x=z;
49       u=w;
50     endif
51   endfor
52 endfunction
```

```
>> x= bisection ( -200, 200, 1000)
x = 1.0000
>>
```

HW4-2:

Procedure Newton(f, f', x, n_max, ϵ, δ)

- integer n, n_max
- real x, fx, fp, ϵ
- interface external function f, f'
- fx $\leftarrow f(x)$
- output 0, x, fx
- for $n = 1$ to n_max do
  - fp $\leftarrow f'(x)$
  - if $|fp| < \delta$ then
    - output "small derivate"
    - return
  - end if
  - d $\leftarrow$ fx/fp
  - x $\leftarrow$ x-d
  - fx $\leftarrow$ f(x)
  - output n, x, fx
  - if $|d| < \epsilon$ then
    - output "converge"
    - return
  - end if
- end for

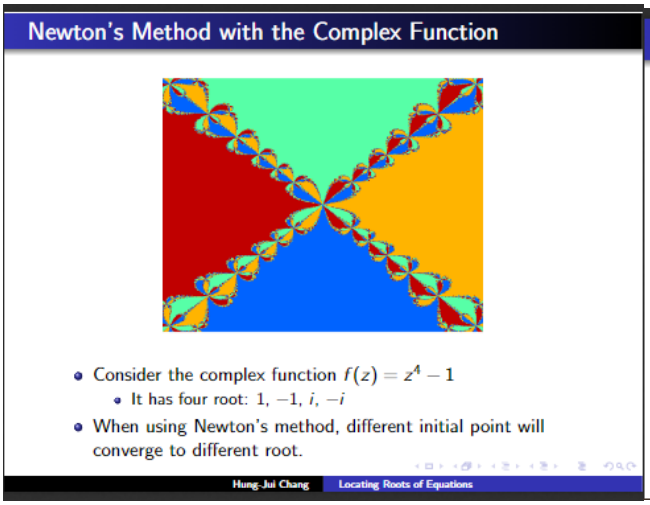
Hung-Jui Chang | Locating Roots of Equations

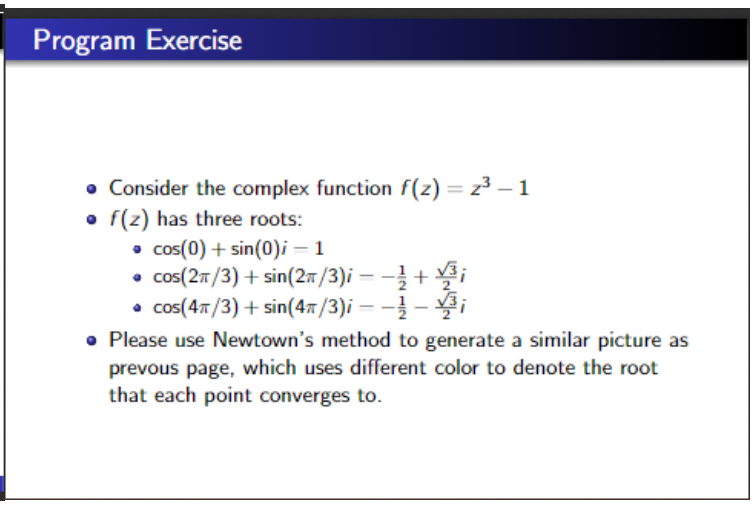
```
26 function x = newton (x,n_max,e,del)
27   #考慮f(x) = (x-a)(x^2+x+(b+c))，其中abc是你學號的末三碼。
28   #請分別撰寫用二分搜尋法及牛頓法找到這個函數的解的程式
29   a=1;
30   b=0;
31   c=7;
32   fx=(x-a)*(x^2+x+(b+c));
33   for n=1:n_max
34     fp=3*x^2+6;
35     if abs(fp)<del
36       #print("small derivate");
37       break
38     endif
39     d=fx/fp;
40     x=x-d;
41     fx=(x-a)*(x^2+x+(b+c));
42     if abs(d)<e
43       #print("converge")
44       break;
45       1+1
46     endif
47
48   endfor
49
50 endfunction
```

```
>> x= bisection ( -200, 200, 1000)
x = 1.0000
>> x = newton (200,100,0.1,0.1)
x = 1.0001
>>
```

HW5:

請實作Chapter 3, page 17的課堂練習，並上傳完成的圖檔”z3.png”及 m-file: newton_z.m。如果牛頓法的程式不在上述的.m中，則請額外一併上傳。






```
2  function maps = newton_z ()
3    p=0.001
4    maps=zeros((2-(-2))/p+1);
5    for m=-2:p:2
6      for n=-2:p:2
7        x=m+n*j;
8        y=newton(x,100,0.0001,0.0001);
9        d=distance(y);
10       maps(round((m+2)/p)+1,round((n+2)/p)+1) = d ;
11     endfor
12   endfor
13
14 endfunction
15
16 function [n_i] = distance (x)
17   root= roots([1, 0, 0, -1]);%[1,(-0.5+((3^0.5)/2)*i),(-0.5-((3^0.5)/2)*i)];
18   [n_v,n_i]=min(abs(x-root));
19 endfunction
20
21 function x = newton (x,n_max,e,del)
22   #考慮f(x) = (x-a)(x^2+x+(b+c))，其中abc是你學號的末三碼。
23   #請分別撰寫用二分搜尋法及牛頓法找到這個函數的解的程式
24
25   fx=x^3-1;
26   for n=1:n_max
27     fp=3*x^2;
28     if abs(fp)<del
```

maps double type 401*401 size

```
29       #print("small derivate");
30       break
31     endif
32     d=fx/fp;
33     x=x-d;
34     fx=x^3-1;
35     if abs(d)<e
36       #print("converge")
37       break;
38     endif
39   endfor
40
41 endfunction
```

```
>>
>> image(maps)
>> image(maps*15)
>>
```