

1.Upload the notebook of week two.

funcs.py

```
1. def func1():
2.     print('func1')
3.
4. def func2():
5.     print('func2')
```

import.py

```
1. import funcs
2.
3. funcs.func1()
4. funcs.func2()
5.
6. del funcs
7.
8. from funcs import func1
9.
10. func1()
11. func2()
```

2017550029@ubuntu:~/comphy/03.19\$ python import.py

func1

func2

func1

Traceback (most recent call last):

File "import.py", line 11, in <module>

func2()

NameError: name 'func2' is not defined

ipython log

In [2]: hname = open('funcs.py','r')

In [3]: hname

Out[3]: <open file 'funcs.py', mode 'r' at 0x7f9ff6747660>

In [4]: aa = hname.readline()

In [5]: aa

Out[5]: 'def func1():\n'

In [6]: aa

Out[6]: 'def func1():\n'

In [7]: aa

Out[7]: 'def func1():\n'

In [8]: bb = hname.readline()

In [9]: bb

Out[9]: "\t\tprint('func1')\n"

In [10]: cc = hname.readline()

In [11]: cc

Out[11]: '\n'

In [12]: hname.tell()

Out[12]: 30

In [13]: hname.seek(0)

In [14]: hname.tell()

Out[14]: 0

In [15]: cc

Out[15]: '\n'

In [16]: bb

Out[16]: "\tprint('func1')\n"

In [17]: aa

Out[17]: 'def func1():\n'

In [18]: hname.tell()

Out[18]: 0

In [19]: dd = hname.readline()

In [20]: dd

Out[20]: 'def func1():\n'

In [21]: hname.tell()

Out[21]: 13

In [22]: funcs_title=[]

In [23]: for i in range(1,6):

....: dd = hname.readline()

....: funcs_title.append(dd)

....:

In [24]: funcs_title

Out[24]: ["\tprint('func1')\n", '\n', 'def func2():\n', "\tprint('func2')\n", '']

In [25]: for i in hname:

....: abcdef = hname.readline()

....: funcs_title.append(abcdef)

....:

In [26]: func

•

• • • • •

```
('funcs', 'def func1():\n')
('funcs', "\t\tprint('func1')\n")
('funcs', '\n')
('funcs', 'def func2():\n')
('funcs', "\t\tprint('func2')\n")
```

```
In [37]: rj
Out[37]:
['def func1():\n',
 '\t\tprint('func1')\n',
 '\n',
 'def func2():\n',
 '\t\tprint('func2')\n']
```

```
In [38]: ncount
Out[38]: 5
```

```
In [39]: hname.close()
```

```
In [45]: aa
Out[45]: 0.123456789
```

```
In [46]: type(aa)
Out[46]: float
```

```
In [47]: print('{:7.5f}'.format(aa))
0.12346
```

```
In [48]: print('{:8.5f}'.format(aa))
0.12346
```

```
In [49]: print('{:1.2f}'.format(aa))
0.12
```

```
In [50]: bb = 4
```

```
In [51]: print('{:1.2f} {:3d} '.format(aa,bb))  
0.12 4
```

```
In [52]: bb = 50001
```

```
In [53]: print('{:1.2f} {:3d} '.format(aa,bb))  
0.12 50001
```

```
# Problem.  
# Iterate from 0 to 10.  
# Open a file 'nexp.txt' to write the results of i, exp(i).  
# Set an appropriate format.  
# Close the file
```

nexp.py

```
1. def fac(n):  
2.     fac = 1  
3.     for i in range(1,n+1):  
4.         fac = fac * i  
5.     return fac  
6.  
7. def exponential(x,n):  
8.     exp = 0  
9.     for i in range(0,n+1):  
10.         exp = exp + x**i/fac(i)  
11.     return exp  
12.  
13. def exp(x):  
14.     return exponential(x,100)  
15.  
16. aa = []  
17.  
18. f = open('nexp.txt','w')  
19.  
20. for i in range(1,11):  
21.     f.write('{:2d} {:.7.10f}\n'.format(i,exp(i)))  
22.     aa.append('{:2d} {:.7.10f}\n'.format(i,exp(i)))  
23.  
24. f.close()  
25. print(aa)
```

```
$ python nexp.py
```

```
[' 1 2.7182818285\n', ' 2 7.3890560989\n', ' 3 20.0855369232\n', ' 4 54.5981500331\n',  
' 5 148.4131591026\n', ' 6 403.4287934927\n', ' 7 1096.6331584285\n', ' 8  
2980.9579870417\n', ' 9 8103.0839275754\n', '10 22026.4657948067\n']
```

nexp.txt

```
1 2.7182818285
2 7.3890560989
3 20.0855369232
4 54.5981500331
5 148.4131591026
6 403.4287934927
7 1096.6331584285
8 2980.9579870417
9 8103.0839275754
10 22026.4657948067
```

03.20

In [7]:

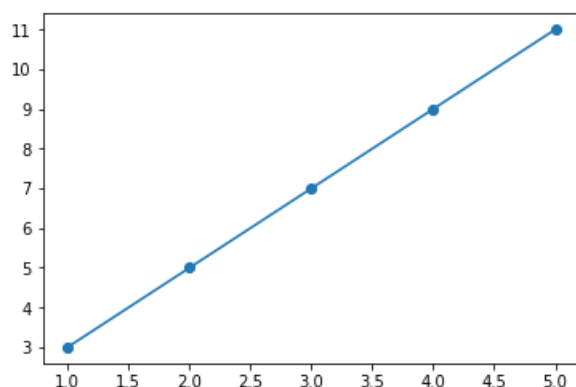
```
1. f=open('Romeo header.txt','r')
2.
3. aa, bb, cc = [], [], []
4.
5. for ii in f.readlines():
6.     aa.append(ii)
7.
8. print(aa)
9. for ii in aa:
10.     bb.extend(ii.split())
11.
12. print(bb)
13.
14. for ii in bb:
15.     if not(ii in cc):
16.         cc.append(ii)
17.
18. print(cc)
```


out [7]:

```
['Romeo and Juliet Act 2,Wn', "Scene 2 SCENE II. Capulet's orchard.Wn", 'Enter ROMEO', 'ROMEO He jests at scars that never felt a wound.Wn', 'JULIET appears above at a window But, soft! what light through yonder window breaks?Wn', 'It is the east, and Juliet is the sun. Arise, fair sun, and kill the envious moon,Wn', 'Who is already sick and pale with grief, That thou her maid art far more fair than she:Wn', 'Be not her maid, since she is envious;']  
['Romeo', 'and', 'Juliet', 'Act', '2,', 'Scene', '2', 'SCENE', 'II.', "Capulet's", 'orchard.', 'Enter', 'ROMEO', 'ROMEO', 'He', 'jest', 'at', 'scars', 'that', 'never', 'felt', 'a', 'wound.', 'JULIET', 'appears', 'above', 'at', 'a', 'window', 'But', 'soft!', 'what', 'light', 'through', 'yonder', 'window', 'breaks?', 'It', 'is', 'the', 'east,', 'and', 'Juliet', 'is', 'the', 'sun.', 'Arise,', 'fair', 'sun,', 'and', 'kill', 'the', 'envious', 'moon,', 'Who', 'is', 'already', 'sick', 'and', 'pale', 'with', 'grief,', 'That', 'thou', 'her', 'maid', 'art', 'far', 'more', 'fair', 'than', 'she:', 'Be', 'not', 'her', 'maid,', 'since', 'she', 'is', 'envious;']  
['Romeo', 'and', 'Juliet', 'Act', '2,', 'Scene', '2', 'SCENE', 'II.', "Capulet's", 'orchard.', 'Enter', 'ROMEO', 'He', 'jest', 'at', 'scars', 'that', 'never', 'felt', 'a', 'wound.', 'JULIET', 'appears', 'above', 'window', 'But,', 'soft!', 'what', 'light', 'through', 'yonder', 'breaks?', 'It', 'is', 'the', 'east,', 'sun.', 'Arise,', 'fair', 'sun,', 'kill', 'envious', 'moon,', 'Who', 'already', 'sick', 'pale', 'with', 'grief,', 'That', 'thou', 'her', 'maid', 'art', 'far', 'more', 'than', 'she:', 'Be', 'not', 'maid,', 'since', 'she', 'envious;']
```

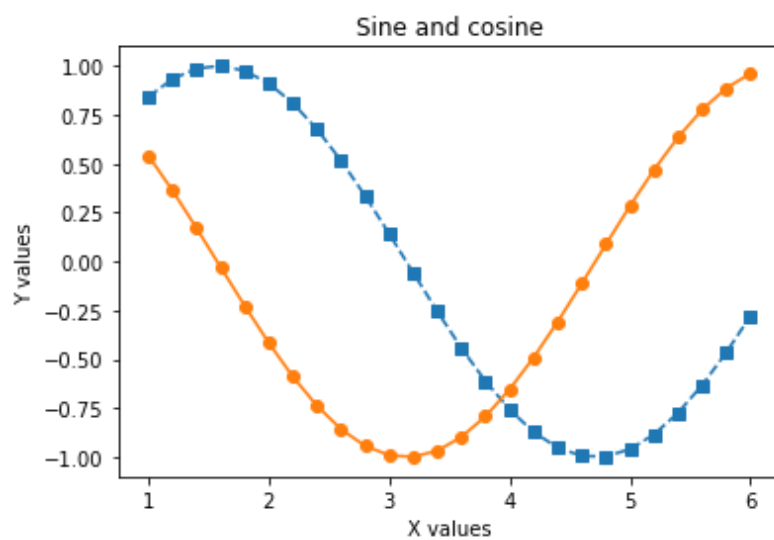
In [21]:

```
1. import matplotlib.pyplot as plt  
2. import numpy as np  
3.  
4. x = np.array([1,2,3,4,5])  
5. y = 2*x+1  
6.  
7. plt.plot(x,y,'-o')  
8.  
9. plt.show()
```



In [15]:

```
1. import matplotlib.pyplot as plt
2. import numpy as np
3.
4. x = np.arange(1,6.2,0.2)
5. y = np.sin(x)
6. z = np.cos(x)
7.
8. plt.plot(x,y,'--s',x,z,'-o')
9. plt.title('Sine and cosine')
10. plt.ylabel('Y values')
11. plt.xlabel('X values')
12.
13. plt.show()
```



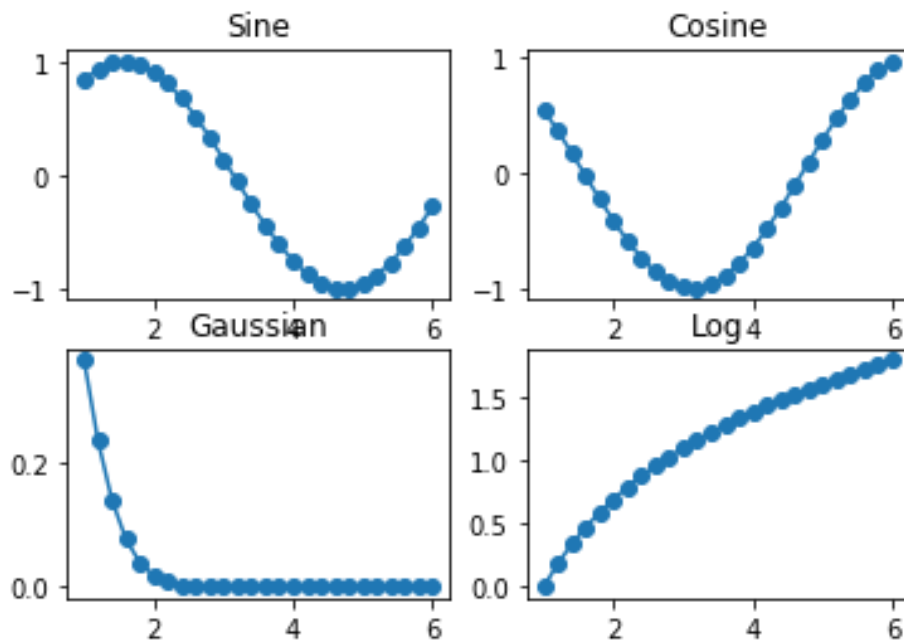
In [19]:

```
1. import matplotlib.pyplot as plt
2. import numpy as np
3.
4. x = np.arange(1,6.2,0.2)
5. y = [np.sin(x), np.cos(x), np.exp(-x**2.0), np.log(x)]
6. name = ['Sine', 'Cosine', 'Gaussian', 'Log']
7.
8. for i in range(0,4):
```

```

9.     plt.subplot(2,2,i+1)
10.    plt.plot(x,y[i],'-o')
11.    plt.title(name[i])
12.
13.    plt.show()

```



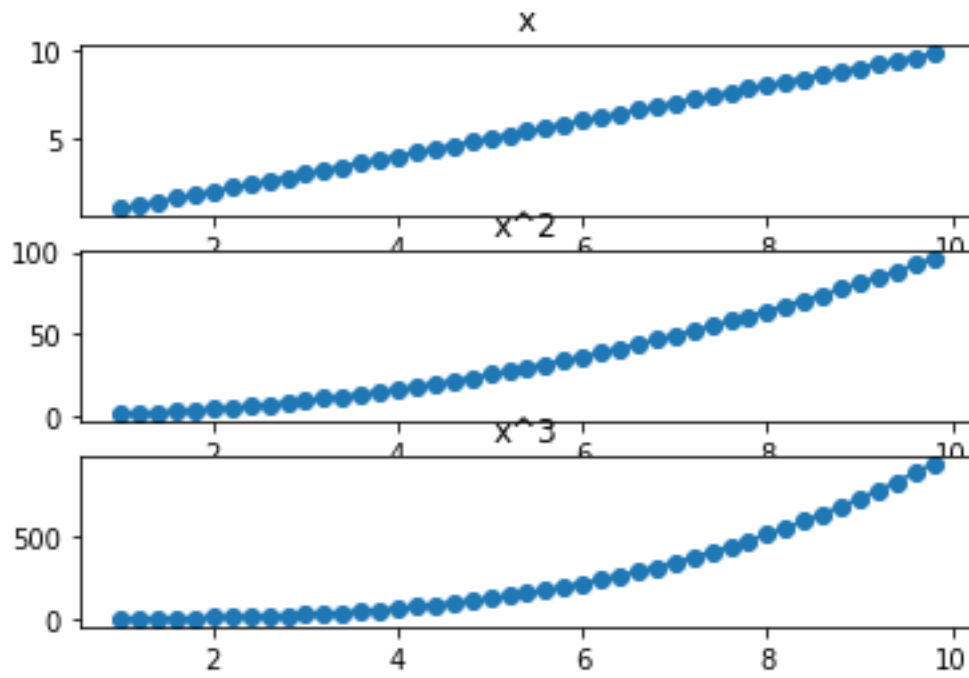
In [20]:

```

1. #problem
2. #Create three subplots,
3. #and plot respectively y=x, y=x^2 and y=x^3
4. #for x ranging between 0 to 10.
5. #Distinguish the graphs by different symbols.
6.
7. import matplotlib.pyplot as plt
8. import numpy as np
9.
10. x = np.arange(1,10,0.2)
11. y = [x, x**2, x**3]
12. name = ['x', 'x^2','x^3']
13.
14. for i in range(0,3):
15.     plt.subplot(3,1,i+1)
16.     plt.plot(x,y[i],'-o')

```

```
17. plt.title(name[i])  
18.  
19. plt.show()
```



2. Writing to a file. Write a program to create an array consisting of numbers ranging from -10 to 10, with a spacing of 0.1. Calculate the functions of $\exp(-x^2)$, and $\sin(x)/x$. Create a file by the name of 'results.txt' and write three columns consisting of the sampled points, and the two functions. Use integer, floating point, and exponential formats for printing each one of the columns. (Figure out a solution for the singular point at zero).

HW2-2.py

```
1. import numpy as np
2.
3. x = []
4.
5. trigger = -100
6. while (trigger <= 100):
7.     x.append(trigger/10.0)
8.     trigger = trigger + 1
9.
10. result1, result2 = [], []
11. for i in x:
12.     result1.append(np.exp(-1*i**2))
13.     if (i == 0):
14.         result2.append('div 0')
15.     else:
16.         result2.append(np.sin(i)/i)
17.
18. f = open('results.txt','w')
19.
20. j = 0
21. for i in x:
22.     if (i == 0):
23.         f.write('{:5.1f}    {:10.4g}    '.format(i,result1[j]))
24.         f.write(' undefined\n')
25.     else:
26.         f.write('{:5.1f}    {:10.4g}    {:10.6f}\n'.format(i,result1[j],result2[j]))
27.     j = j+1
28.
29. f.close()
```

results.txt

-10.0	3.72e-44	-0.054402
-9.9	2.721e-43	-0.046216
-9.8	1.951e-42	-0.037396
-9.7	1.372e-41	-0.028017
-9.6	9.45e-41	-0.018159
-9.5	6.382e-40	-0.007911
-9.4	4.224e-39	0.002636
-9.3	2.741e-38	0.013382
-9.2	1.743e-37	0.024227
-9.1	1.087e-36	0.035066
-9.0	6.64e-36	0.045791
-8.9	3.977e-35	0.056294
-8.8	2.335e-34	0.066468
-8.7	1.344e-33	0.076203
-8.6	7.578e-33	0.085395
-8.5	4.19e-32	0.093940
-8.4	2.271e-31	0.101738
-8.3	1.206e-30	0.108695
-8.2	6.281e-30	0.114723
-8.1	3.206e-29	0.119739
-8.0	1.604e-28	0.123670
-7.9	7.865e-28	0.126448
-7.8	3.78e-27	0.128018
-7.7	1.781e-26	0.128334
-7.6	8.225e-26	0.127358
-7.5	3.723e-25	0.125067
-7.4	1.652e-24	0.121447
-7.3	7.185e-24	0.116498
-7.2	3.063e-23	0.110232
-7.1	1.28e-22	0.102672
-7.0	5.243e-22	0.093855
-6.9	2.105e-21	0.083832
-6.8	8.284e-21	0.072664
-6.7	3.195e-20	0.060425
-6.6	1.208e-19	0.047203

-6.5	4.478e-19	0.033095
-6.4	1.627e-18	0.018211
-6.3	5.792e-18	0.002669
-6.2	2.022e-17	-0.013402
-6.1	6.917e-17	-0.029863
-6.0	2.32e-16	-0.046569
-5.9	7.624e-16	-0.063369
-5.8	2.457e-15	-0.080104
-5.7	7.758e-15	-0.096611
-5.6	2.402e-14	-0.112726
-5.5	7.288e-14	-0.128280
-5.4	2.168e-13	-0.143105
-5.3	6.319e-13	-0.157032
-5.2	1.806e-12	-0.169895
-5.1	5.058e-12	-0.181532
-5.0	1.389e-11	-0.191785
-4.9	3.738e-11	-0.200501
-4.8	9.86e-11	-0.207534
-4.7	2.549e-10	-0.212750
-4.6	6.461e-10	-0.216020
-4.5	1.605e-09	-0.217229
-4.4	3.909e-09	-0.216273
-4.3	9.33e-09	-0.213062
-4.2	2.183e-08	-0.207518
-4.1	5.006e-08	-0.199580
-4.0	1.125e-07	-0.189201
-3.9	2.48e-07	-0.176350
-3.8	5.355e-07	-0.161015
-3.7	1.134e-06	-0.143199
-3.6	2.353e-06	-0.122922
-3.5	4.785e-06	-0.100224
-3.4	9.54e-06	-0.075159
-3.3	1.864e-05	-0.047802
-3.2	3.571e-05	-0.018242
-3.1	6.705e-05	0.013413
-3.0	0.0001234	0.047040

-2.9	0.0002226	0.082500
-2.8	0.0003937	0.119639
-2.7	0.0006823	0.158289
-2.6	0.001159	0.198270
-2.5	0.00193	0.239389
-2.4	0.003151	0.281443
-2.3	0.005042	0.324220
-2.2	0.007907	0.367498
-2.1	0.01216	0.411052
-2.0	0.01832	0.454649
-1.9	0.02705	0.498053
-1.8	0.03916	0.541026
-1.7	0.05558	0.583332
-1.6	0.0773	0.624734
-1.5	0.1054	0.664997
-1.4	0.1409	0.703893
-1.3	0.1845	0.741199
-1.2	0.2369	0.776699
-1.1	0.2982	0.810189
-1.0	0.3679	0.841471
-0.9	0.4449	0.870363
-0.8	0.5273	0.896695
-0.7	0.6126	0.920311
-0.6	0.6977	0.941071
-0.5	0.7788	0.958851
-0.4	0.8521	0.973546
-0.3	0.9139	0.985067
-0.2	0.9608	0.993347
-0.1	0.99	0.998334
0.0	1	undifined
0.1	0.99	0.998334
0.2	0.9608	0.993347
0.3	0.9139	0.985067
0.4	0.8521	0.973546
0.5	0.7788	0.958851
0.6	0.6977	0.941071

0.7	0.6126	0.920311
0.8	0.5273	0.896695
0.9	0.4449	0.870363
1.0	0.3679	0.841471
1.1	0.2982	0.810189
1.2	0.2369	0.776699
1.3	0.1845	0.741199
1.4	0.1409	0.703893
1.5	0.1054	0.664997
1.6	0.0773	0.624734
1.7	0.05558	0.583332
1.8	0.03916	0.541026
1.9	0.02705	0.498053
2.0	0.01832	0.454649
2.1	0.01216	0.411052
2.2	0.007907	0.367498
2.3	0.005042	0.324220
2.4	0.003151	0.281443
2.5	0.00193	0.239389
2.6	0.001159	0.198270
2.7	0.0006823	0.158289
2.8	0.0003937	0.119639
2.9	0.0002226	0.082500
3.0	0.0001234	0.047040
3.1	6.705e-05	0.013413
3.2	3.571e-05	-0.018242
3.3	1.864e-05	-0.047802
3.4	9.54e-06	-0.075159
3.5	4.785e-06	-0.100224
3.6	2.353e-06	-0.122922
3.7	1.134e-06	-0.143199
3.8	5.355e-07	-0.161015
3.9	2.48e-07	-0.176350
4.0	1.125e-07	-0.189201
4.1	5.006e-08	-0.199580
4.2	2.183e-08	-0.207518

4.3	9.33e-09	-0.213062
4.4	3.909e-09	-0.216273
4.5	1.605e-09	-0.217229
4.6	6.461e-10	-0.216020
4.7	2.549e-10	-0.212750
4.8	9.86e-11	-0.207534
4.9	3.738e-11	-0.200501
5.0	1.389e-11	-0.191785
5.1	5.058e-12	-0.181532
5.2	1.806e-12	-0.169895
5.3	6.319e-13	-0.157032
5.4	2.168e-13	-0.143105
5.5	7.288e-14	-0.128280
5.6	2.402e-14	-0.112726
5.7	7.758e-15	-0.096611
5.8	2.457e-15	-0.080104
5.9	7.624e-16	-0.063369
6.0	2.32e-16	-0.046569
6.1	6.917e-17	-0.029863
6.2	2.022e-17	-0.013402
6.3	5.792e-18	0.002669
6.4	1.627e-18	0.018211
6.5	4.478e-19	0.033095
6.6	1.208e-19	0.047203
6.7	3.195e-20	0.060425
6.8	8.284e-21	0.072664
6.9	2.105e-21	0.083832
7.0	5.243e-22	0.093855
7.1	1.28e-22	0.102672
7.2	3.063e-23	0.110232
7.3	7.185e-24	0.116498
7.4	1.652e-24	0.121447
7.5	3.723e-25	0.125067
7.6	8.225e-26	0.127358
7.7	1.781e-26	0.128334
7.8	3.78e-27	0.128018

7.9	7.865e-28	0.126448
8.0	1.604e-28	0.123670
8.1	3.206e-29	0.119739
8.2	6.281e-30	0.114723
8.3	1.206e-30	0.108695
8.4	2.271e-31	0.101738
8.5	4.19e-32	0.093940
8.6	7.578e-33	0.085395
8.7	1.344e-33	0.076203
8.8	2.335e-34	0.066468
8.9	3.977e-35	0.056294
9.0	6.64e-36	0.045791
9.1	1.087e-36	0.035066
9.2	1.743e-37	0.024227
9.3	2.741e-38	0.013382
9.4	4.224e-39	0.002636
9.5	6.382e-40	-0.007911
9.6	9.45e-41	-0.018159
9.7	1.372e-41	-0.028017
9.8	1.951e-42	-0.037396
9.9	2.721e-43	-0.046216
10.0	3.72e-44	-0.054402

3. Making figures.

Read the file 'results.txt' from the previous exercise and plot the two figures as side by side subplots. Use dashed lines for the left figure and dotted lines for the figure on the right. Create an appropriate title, and the figure labels for the two plots.

HW2-3.py

```
1. import matplotlib.pyplot as plt
2.
3. results = open('results.txt','r')
4.
5. aa, bb= [], []
6.
7. for ii in results.readlines():
8.     aa.append(ii)
9.
10. for ii in aa:
11.     bb.extend(ii.split())
12.
13. x, f1, f2 = [], [], []
14. f = [x, f1, f2]
15. i = 0
16.
17. for ii in bb:
18.     try:
19.         f[i%3].append(float(ii))
20.         i += 1
21.     except:
22.         f[i%3].append(1) # sinc(0) is not defined. but limit is 1.
23.         i += 1
24.
25. title = [0,'Gaussian', 'Sinc']
26. linestyle = [0, '--', ':']
27. color = [0, 'blue', 'red']
28.
29. for i in [1,2]:
30.     plt.subplot(1,2,i)
31.     plt.plot(x,f[i],linestyle[i],color=color[i])
32.     plt.title(title[i])
```

```
33. plt.ylabel('Y value')
34. plt.xlabel('X value')
35. plt.tight_layout()
36.
37. plt.savefig('results.png')
```

results.png

