C and Python

G V V Sharma*

Abstract—**This manual shows how to generate data in a file using a C program and importing it in Python.**

Problem 1. Graphically show that the function

$$f(x) = \begin{cases} -x & x < 1 \\ a + \cos^{-1}(x+b) & 1 \le x \le 2 \end{cases} \tag{1.1}$$

is continuous at $x = 1$ for $b = -1, a - b = -\frac{\pi}{2}$.

Solution: The following python code yields Fig. 1 verifying the above result.

```
import numpy as np
import matplotlib.pyplot as plt
#Computation
b = -1
x2 = np.linspace(-1,1,100)
x3 = np.linspace(1,2,100)
a = -1 - np.pi/2.0
y = -x2
z = a + np.arccos(b+(x3))

#Plotting
plt.plot(x3,z, label = '$f(x)␣=␣-
    x$')
plt.plot(x2,y, label = '$f(x)␣=␣a␣
    +␣\cos^{-1}(x+b)$')

sol = np.zeros((2,1))
sol[0] = 1
sol[1] = -1

#Display solution
A = sol[0]
B = sol[1]

plt.plot(A,B,'o')
for xy in zip(A,B):
        plt.annotate('(%s,␣%s)' %
            xy, xy=xy, xytext=(30,0)
            , textcoords='offset␣
            points')

plt.grid()
plt.legend(loc='best',prop={'size'
    :11})
plt.xlabel('$x$')
plt.ylabel('$f(x)$')
#Comment the following line
#plt.savefig('../figs/ee16b1005.
    eps')
plt.show()
```

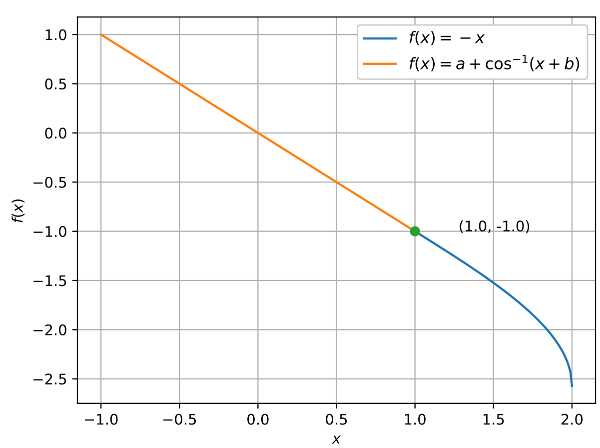

Fig. 1: Substituting the values of a and b in $f(x)$, the graph is smooth at $x = 1$. So $f(x)$ is continuous as well as differentiable $x = 1$.

Problem 2. Write a C program to generate an arithmetic progression with first term $a = -1$, last term $l = 1$ and number of terms $n = 100$ and print the numbers on the screen.

Solution:

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

```c
#include <stdio.h>

int main(void)
{
float a = -1.0, l = 1.0, d;
int n = 100, i;

//Common difference
d = (l-a)/(n-1);

for(i = 0; i < 100; i++)
{
printf("%f\n",a+i*d);
}

return 0;
}
```

Problem 3. Repeat the above exercise by using functions for finding the common difference and the $n$term given a, l and n.

Solution:

```c
#include <stdio.h>

float a_n(float,float,int);
float c_d(float,float,int);

int main(void)
{
float a = -1.0, l = 1.0, d;
int n = 100, i;
d = c_d(a,l,n);
for(i = 0; i < 100; i++)
{
printf("%f\n",a_n(a,d,i));
}

return 0;
}
//nth term of AP
float a_n(float a,float d,int i)
{
return a+i*d;
}

//Common difference
float c_d(float a,float l,int n)
{
float d;
d = (l-a)/(n-1);
return d;
}
```

Problem 4. Repeat the above exercise by printing the numbers in a file called test.dat

Solution:

```c
#include <stdio.h>

int main(void)
{
FILE *fp;
float a = -1.0, l = 1.0, d;
int n = 100, i;

//Common difference
d = (l-a)/(n-1);

//Open file for writing
fp = fopen("test.dat", "w");

for(i = 0; i < 100; i++)
{
fprintf(fp, "%f\n", a+i*d);
}
fclose(fp);
return 0;
}
```

Problem 5. Now run the following program. Comment.

```python
import numpy as np
import matplotlib.pyplot as plt
#Computation
b = -1
x2 = np.loadtxt('test.dat',dtype='float')
#x2 = np.linspace(-1,1,100)
x3 = np.linspace(1,2,100)
a = -1 - np.pi/2.0
y = -x2
z = a + np.arccos(b+(x3))

#Plotting
plt.plot(x3,z, label = '$f(x) = -x$')
```

```
plt.plot(x2,y, label = '$f(x) = a + \cos^{-1}(x+b)$')

sol = np.zeros((2,1))
sol[0] = 1
sol[1] = -1

#Display solution
A = sol[0]
B = sol[1]

plt.plot(A,B,'o')
for xy in zip(A,B):
        plt.annotate('(%s, %s)' % xy, xy=xy, xytext=(30,0), textcoords='offset points')

plt.grid()
plt.legend(loc='best',prop={'size':11})
plt.xlabel('$x$')
plt.ylabel('$f(x)$')
plt.show()
```

Problem 6. Compute $f(x)$ in (1.1) through a C program

Solution:

```
#include <stdio.h>
#include <math.h>

float f(float);


int main(void)
{

printf("%f\n",f(1.99));

return 0;
}

//Common difference
float f(float x)
{
float b = -1.0, a;

a = b-M_PI_2;

if(x < 1)
{
        return -x;

}
else if(x >= 1 && x <= 2)
{
return a+acos(x+b);
}
else

return 0;

}
```

Problem 7. Do all the computations in Problem 1 in C and verify your results by plotting in python.