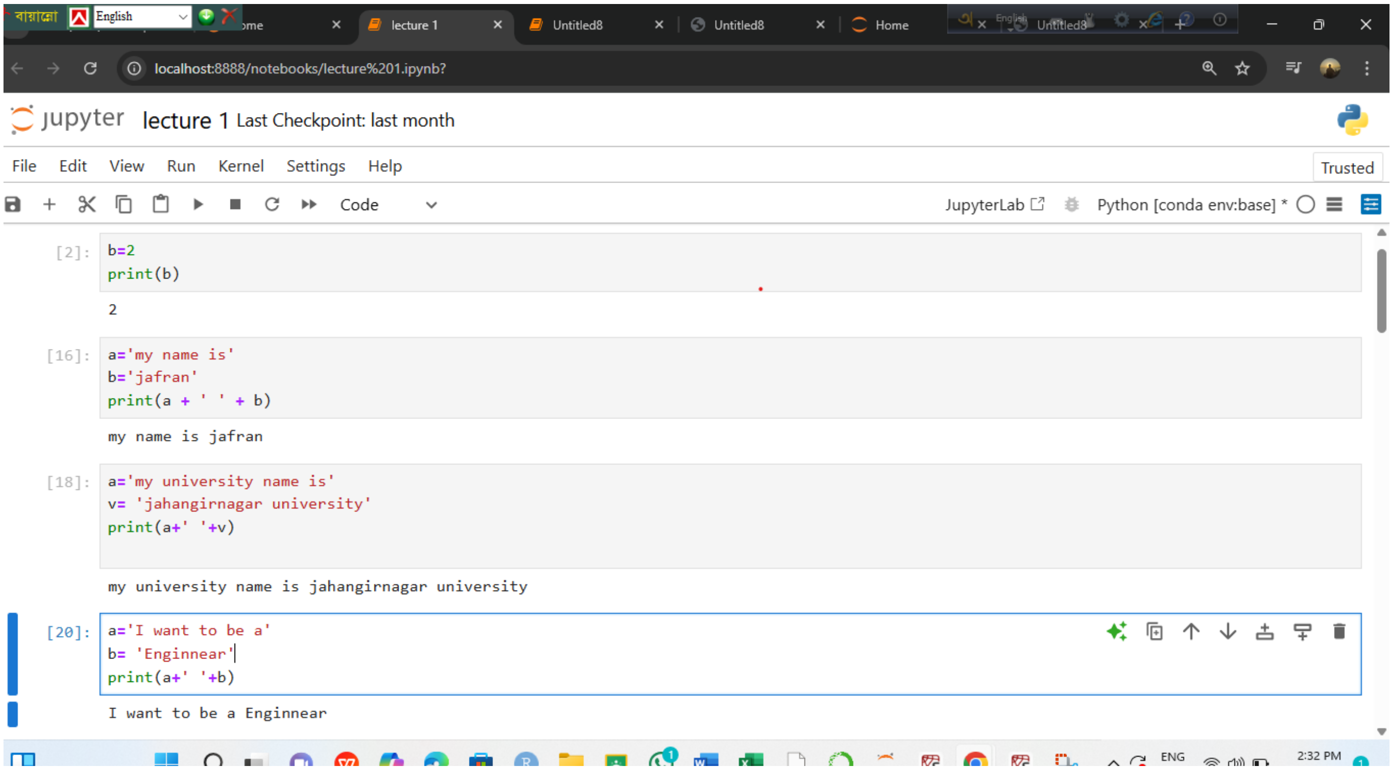
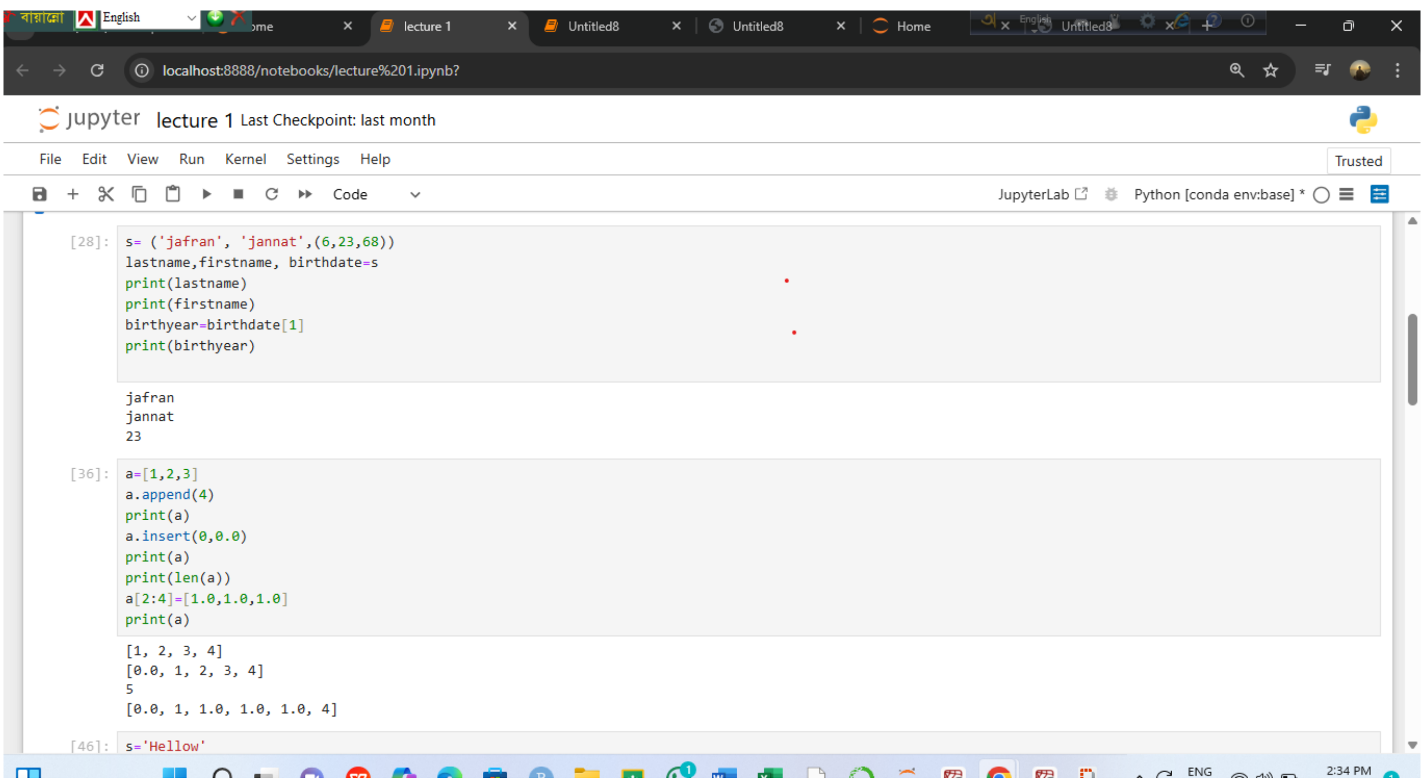
LECTUR 01:(Introduction to computing with python)



```
[2]: b=2
     print(b)
```
```
2
```
```
[16]: a='my name is'
      b='jafran'
      print(a + ' ' + b)
```
```
my name is jafran
```
```
[18]: a='my university name is'
      v= 'jahangirnagar university'
      print(a+' '+v)
```
```
my university name is jahangirnagar university
```
```
[20]: a='I want to be a'
      b= 'Enginnear'
      print(a+' '+b)
```
```
I want to be a Enginnear
```



```
[28]: s= ('jafran', 'jannat',(6,23,68))
      lastname,firstname, birthdate=s
      print(lastname)
      print(firstname)
      birthyear=birthdate[1]
      print(birthyear)
```
```
jafran
jannat
23
```
```
[36]: a=[1,2,3]
      a.append(4)
      print(a)
      a.insert(0,0.0)
      print(a)
      print(len(a))
      a[2:4]=[1.0,1.0,1.0]
      print(a)
```
```
[1, 2, 3, 4]
[0.0, 1, 2, 3, 4]
5
[0.0, 1, 1.0, 1.0, 1.0, 4]
```
```
[46]: s='Hellow'
```

localhost:8888/notebooks/lecture%201.ipynb?

jupyter lecture 1 Last Checkpoint: last month

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[46]:
```python
s='Hellow'
t='to you'
b=2
print(b+2)
print(b-2)
print(b*2)
print(b/2)
print(b**3)
print(b%3)
a=[1,2.3]
print(3*s)
print(3*a)
print(s+t)
print(3+s)
```

```
4
0
4
1.0
8
2
HellowHellowHellow
[1, 2.3, 1, 2.3, 1, 2.3]
Hellowto you
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
```

localhost:8888/notebooks/lecture%201.ipynb?

jupyter lecture 1 Last Checkpoint: last month

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[50]:
```python
a=2
b=1.99
c='2'
print(a>b)
print(a==c)
print((a>b) and(a!=c))
print((a>b)or(a==b))
```

```
True
False
True
True
```

[56]:
```python
def sign_of_a(a):
    if a<0.0:
        sign='negative'
    elif a>0.0:
        sign='positive'
    else:
        sign='zero'
    return sign
a=-1.5
print('a  is '+sign_of_a(a))
```

```
a  is negative
```

localhost:8888/notebooks/lecture%201.ipynb?

jupyter lecture 1 Last Checkpoint: last month

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[58]:

```python
nmax=5
n=1
a=[]
while n<nmax:
    a.append(1/n)
    n=n+1
print(a)
```

```
[1.0, 0.5, 0.3333333333333333, 0.25]
```

[60]:

```python
nmax=5
a=[]
for n in range(1,nmax):
    a.append(1/n)
    print(a)
```

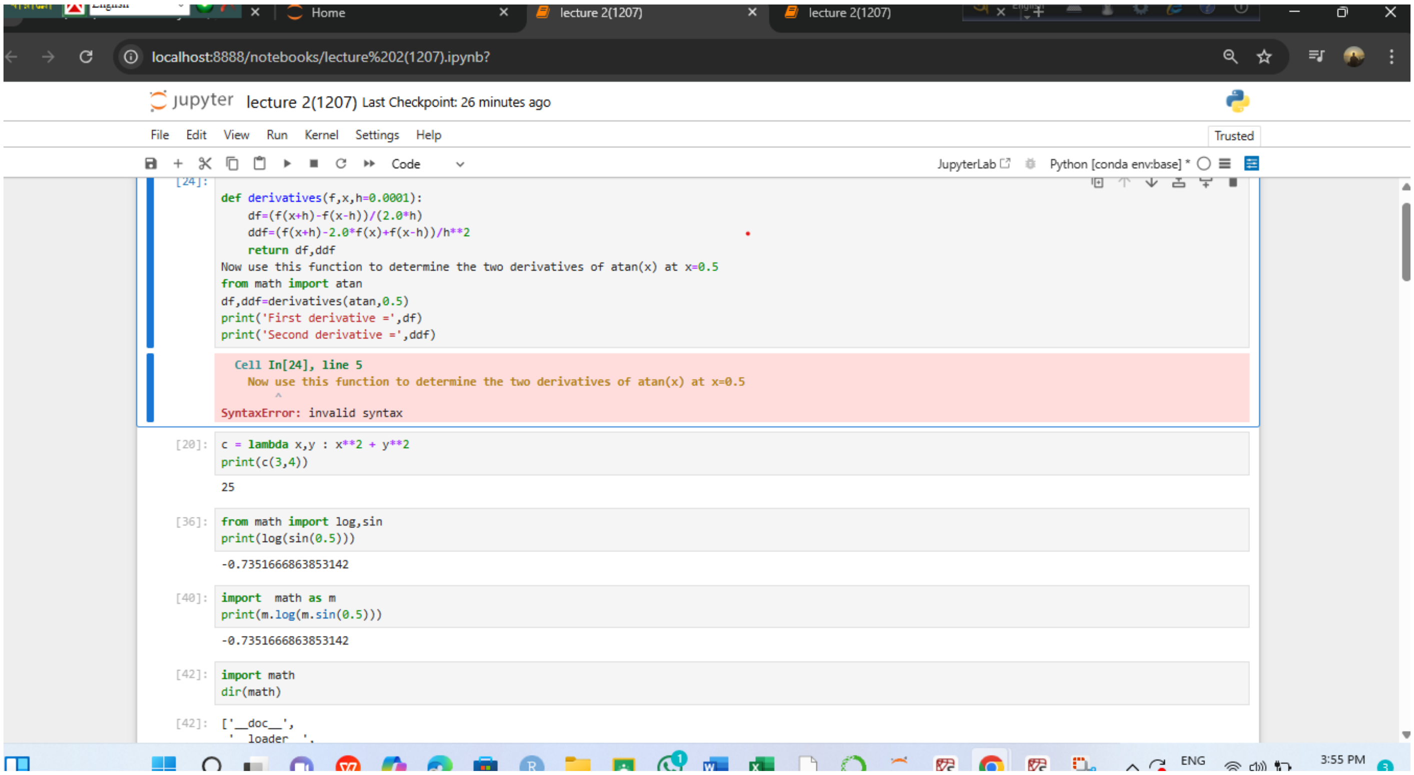
```
[1.0]
[1.0, 0.5]
[1.0, 0.5, 0.3333333333333333]
[1.0, 0.5, 0.3333333333333333, 0.25]
```

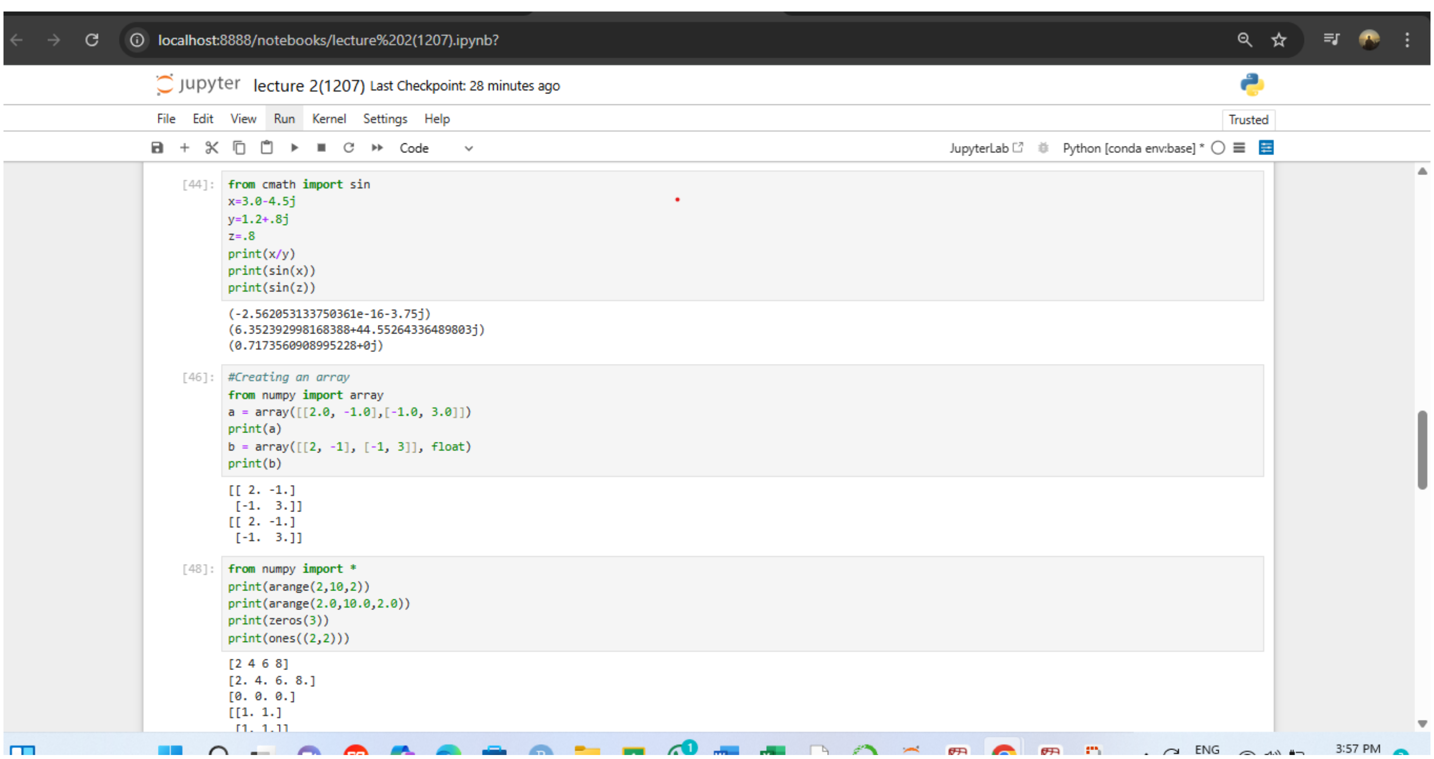
[64]:

```python
x=[]
for i in range(1,100):
    if i%7!=0:continue
    x.append(i)
print(x)
```

```
[7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98]
```

Lecture 02:(Function and Modules)





localhost:8888/notebooks/lecture%202(1207).ipynb?

jupyter lecture 2(1207) Last Checkpoint: 32 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[50]:
```python
from numpy import *
a = zeros((3,3),int)
print(a)
a[0] = [2,3,2]
a[1,1] = 5
a[2, 0:2] = [8,-3]
print(a)
```

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]
[[ 2  3  2]
 [ 0  5  0]
 [ 8 -3  0]]
```

[52]:
```python
from numpy import array
a = array([0.0, 4.0, 9.0, 16.0])
print(a/16.0)
print(a - 4.0)
```

```
[0.     0.25   0.5625 1.    ]
[-4.  0.  5. 12.]
```

[54]:
```python
from numpy import array,sqrt,sin
a = array([1.0, 4.0, 9.0, 16.0])
print(sqrt(a))
print(sin(a))
```

```
[1. 2. 3. 4.]
[ 0.84147098 -0.7568025   0.41211849 -0.28790332]
```

[65]:
```python
from numpy import *
x = array([7,3])

y = array([2,1])
```

4:01 PM

Home | lecture 2(1207) | lecture 2(1207)

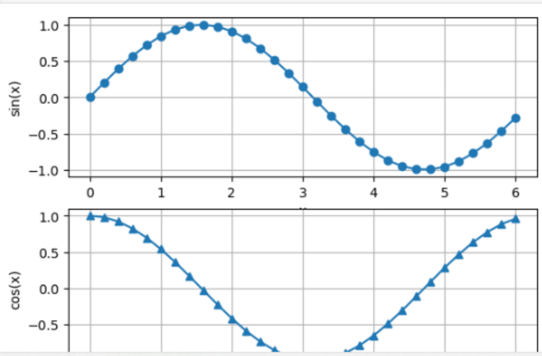
localhost:8888/notebooks/lecture%202(1207).ipynb?

jupyter lecture 2(1207) Last Checkpoint: 34 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[67]:
```python
import matplotlib.pyplot as plt
from numpy import arange,sin,cos
x = arange(0.0, 6.2, 0.2)
plt.subplot(2,1,1)
plt.plot(x,sin(x),'o-')
plt.xlabel('x');plt.ylabel('sin(x)')
plt.grid(True)
plt.subplot(2,1,2)
plt.plot(x,cos(x),'^-')
plt.xlabel('x');plt.ylabel('cos(x)')
plt.grid(True)
plt.show()
```



4:03 PM

Lecture 03:(Bisection Method)

Example1: A root of $x^3 - 10x^2 + 5 = 0$ lies in the interval (0, 1). Use rootsearch to compute this root with four-digit accuracy.

localhost:8888/notebooks/Untitled9.ipynb?

jupyter Untitled9 Last Checkpoint: 28 minutes ago

```python
def f(x):
    return x**3-10*x**2+5
def find_root(a, b):
    for i in range(100):
        mid = (a + b) / 2
        if f(mid) == 0 or abs(b - a) < 0.0001:
            return mid
        if f(a) * f(mid) < 0:
            b = mid
        else:
            a = mid
    return mid
root = find_root(0, 1)
print("Root is:", root)
```

```
Root is: 0.734588623046875
```

Class Task:

1. Show that $f(x) = x^3 + 4x^2 - 10 = 0$ has a root in [1,2] and use the Bisection method to determine an approximation to the root that is accvurate to at least within 10^{-4}.

localhost:8888/notebooks/Untitled9.ipynb?

jupyter Untitled9 Last Checkpoint: 1 hour ago

```python
def f(x):
    return x**3+4*x**2-10
def find_root(a, b):
    for i in range(100):
        mid = (a + b) / 2
        if f(mid) == 0 or abs(b - a) < 0.0001:
            return mid
        if f(a) * f(mid) < 0:
            b = mid
        else:
            a = mid
    return mid
root = find_root(1,2)
print("Root is:", root)
```

```
Root is: 1.365203857421875
```

2. Use the Bisection method to find p_3 for $f(x) = \sqrt{x} - cosx = 0$ on $[0,1]$.

```
[18]: import math
      def f(x):
          return math.sqrt(x)-math.cos(x)
      def find_root(a, b):
          for i in range(100):
              mid = (a + b) / 2
              if f(mid) == 0 or abs(b - a) < 0.0001:
                  return mid
              if f(a) * f(mid) < 0:
                  b = mid
              else:
                  a = mid
          return mid
      root = find_root(0,1)
      print("Root is:", root)

      Root is: 0.641693115234375
```

Homework

1. Use the Bisection method to find solutions accurate to within 10^{-2} for $x^3 - 7x^2 + 14x - 6 = 0$ on each interval.

```
[20]: def f(x):
          return x**3-7*x**2+14*x-6
      def find_root(a, b):
          for i in range(100):
              mid = (a + b) / 2
              if f(mid) == 0 or abs(b - a) < 0.0001:
                  return mid
              if f(a) * f(mid) < 0:
                  b = mid
              else:
                  a = mid
          return mid
      root = find_root(0,1)
      print("Root is:", root)

      Root is: 0.585784912109375
```

Lecture 04:(Newton-Raphson Method)

Use Newton's method to find solutions accurate to within 10^{-4} for the following problems.

1. $x^3 + 3x^2 - 1 = 0$, [-3,-2]

```
[1]: import math
def f(x):
    return x**3+3*x**2-1
def df(x):
    return 3*x**2+6*x
def newton_raphson(x0, tol, max_iter):
    for i in range(max_iter):
        x1 = x0 - f(x0) / df(x0)
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
    return x0
x0 = 1
tol = 0.0001
max_iter = 20
root = newton_raphson(x0, tol, max_iter)
print("Approximate root:", round(root, 4))
```

```
Approximate root: 0.5321
```

3. The equation $x^3 - 1.2x^2 - 8.19x + 13.23 = 0$ has a double root close to $x = 2$. Determine this root with the Newton-Raphson method within four decimal places.

```
def f(x):
    return x**3-1.2*x**2-8.19*x+13.23
def df(x):
    return 3*x**2-2.4*x-8.19
def newton_raphson(x0, tol, max_iter):
    for i in range(max_iter):
        x1 = x0 - f(x0) / df(x0)
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
    return x0
x0 = 1
tol = 0.0001
max_iter = 20
root = newton_raphson(x0, tol, max_iter)
print("Approximate root:", round(root, 4))
```

```
Approximate root: 2.0999
```

2. Determine the two roots of $sinx + 3cosx - 2 = 0$ lies in the interval (-2,2). Use the Newton-Raphson method.

localhost:8888/notebooks/2.ipynb?

jupyter 2 Last Checkpoint: 11 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[8]:
```python
import math
def f(x):
    return math.sin(x)+3*math.cos(x)-2
def df(x):
    return math.cos(x)-3*math.sin(x)
def newton_raphson(x0, tol, max_iter):
    for i in range(max_iter):
        x1 = x0 - f(x0) / df(x0)
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
    return x0
x0 = 1
tol = 0.0001
max_iter = 20
root = newton_raphson(x0, tol, max_iter)
print("Approximate root:", round(root, 4))
```

```
Approximate root: 1.2078
```

2. $x - cosx = 0, \ [0, \pi/2]$

jupyter 2 Last Checkpoint: 11 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[12]:
```python
import math
def f(x):
    return x-math.cos(x)
def df(x):
    return 1+math.sin(x)
def newton_raphson(x0, tol, max_iter):
    for i in range(max_iter):
        x1 = x0 - f(x0) / df(x0)
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
    return x0
x0 = 1
tol = 0.0001
max_iter = 20
root = newton_raphson(x0, tol, max_iter)
print("Approximate root:", round(root, 4))
```

```
Approximate root: 0.7391
```

Lecture 05:(Newton's Interpolation)

Example: The following table given the population of a town during the last six censuses. Using the Newtown's interpolation formula estimate the population in 1923.

Year (x)	1911	1921	1931	1941	1951	1961
Population (y) (in thousands)	12	15	20	27	39	52

```
[12]:
x = [1911,1921,1931,1941,1951,1961]
y = [12,15,20,27,39,52]
xp = 1923

n = len(x)
h = x[1] - x[0]
u = (xp - x[0]) / h
for i in range(1, n):
    for j in range(n-1, i-1, -1):
        y[j] -= y[j-1]
res = y[0]
term = 1
for i in range(1, n):
    term *= (u - i + 1) / i
    res += term * y[i]

print("Value at x =", xp, "is", round(res, 6))
```

```
Value at x = 1923 is 15.96384
```

Class Work:

Exercise 01: The population of a town in the census is given below. Estimate the increase in population during the year 1895 to 1925.

Year	1891	1901	1911	1921	1931
Population (in thousands)	46	66	81	93	101

```
[4]:
x = [1891,1901,1911,1921,1931]
y = [46,66,81,93,101]
xp = 1895

n = len(x)
h = x[1] - x[0]
u = (xp - x[0]) / h
for i in range(1, n):
    for j in range(n-1, i-1, -1):
        y[j] -= y[j-1]
res = y[0]
term = 1
for i in range(1, n):
    term *= (u - i + 1) / i
    res += term * y[i]

print("Value at x =", xp, "is", round(res, 6))
```

```
Value at x = 1895 is 54.8528
```

Exercise 05: Use appropriate interpolation formula to calculate the value of $e^{1.75}$ from the following data.

x	1.7	1.8	1.9	2.0
$y = e^x$	5.474	6.050	6.686	7.389

localhost:8888/notebooks/check.ipynb?

jupyter check Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:

[2]:
```python
x = [1.7,1.8,1.9,2.0]
y = [5.474,6.050,6.686,7.389]
xp = 1.75

n = len(x)
h = x[1] - x[0]
u = (xp - x[0]) / h
for i in range(1, n):
    for j in range(n-1, i-1, -1):
        y[j] -= y[j-1]
res = y[0]
term = 1
for i in range(1, n):
    term *= (u - i + 1) / i
    res += term * y[i]

print("Value at x =", xp, "is", round(res, 6))
```

```
Value at x = 1.75 is 5.754938
```

Exercise 05: Use appropriate interpolation formula to calculate the value of $e^{1.75}$ *and* $e^{1.96}$ from the following data.

x	1.7	1.8	1.9	2.0
$y = e^x$	5.474	6.050	6.686	7.389

jupyter check Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Code

[10]:
```python
x = [1.7,1.8,1.9,2.0]
y = [5.474,6.050,6.686,7.389]
xp = 1.96

n = len(x)
h = x[1] - x[0]
u = (xp - x[-1]) / h
for i in range(1, n):
    for j in range(n-1, i-1, -1):
        y[j] -= y[j-1]
res = y[-1]
term = 1
for i in range(1, n):
    term *= (u + i -1) / i
    res += term * y[-(i+1)]

print("Value at x =", xp, "is", round(res, 6))
```

```
Value at x = 1.96 is -0.436456
```

Lecture 06:(Gauss Forward and Backward)

Example: Find the value of $e^{-1.7425}$ by Gauss Forward formula, given that

x	1.72	1.73	1.74	1.75	1.76
e^{-x}	0.17907	0.17728	0.17552	0.17377	0.17204

jupyter check Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

```
value at x = 1895 is 54.8528
```

[14]:
```python
x = [1.72,1.73,1.74,1.75,1.76]
y = [0.17907,0.17728,0.17552,0.17377,0.17204]
xp = 1.7425
u = xp - x[1]

for i in range(1, len(x)):
    for j in range(len(x)-1, i-1, -1):
        y[j] -= y[j-1]

r = y[1]; p = 1; f = 1
for i in range(1, len(x)):
    f *= i
    k = (i+1)//2 if i%2 else i//2
    p *= (u - k + 1) if i%2 else (u + k - 1)
    r += p * y[1 - k] / f

print(round(r, 6))
```

```
0.000462
```

Homework:

Exercise 03: Apply Gauss forward formula to find the value of $f(x)$ at $x = 3.75$ given

x	2.5	3.0	3.5	4.0	4.5	5.0
$f(x)$	24.145	22.043	20.225	18.644	17.262	16.047

jupyter check Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

```
value at x = 1895 is 54.8528
```

[12]:
```python
x = [2.5,3.0,3.5,4.0,4.5,5.0]
y = [24.145,22.043,20.225,18.644,17.262,16.047]
xp = 3.75
u = xp - x[1]

for i in range(1, len(x)):
    for j in range(len(x)-1, i-1, -1):
        y[j] -= y[j-1]

r = y[1]; p = 1; f = 1
for i in range(1, len(x)):
    f *= i
    k = (i+1)//2 if i%2 else i//2
    p *= (u - k + 1) if i%2 else (u + k - 1)
    r += p * y[1 - k] / f

print(round(r, 6))
```

```
22.797655
```

Example: Apply Stirling's and Bessel's formula to find the value of $f(1.22)$ from the following table which gives the values of $f(x)=\frac{1}{\sqrt{2\pi}}\int_0^x e^{-x^2/2}dx$ at intervals of $h=0.5$ from $x=0$ to 2.5.

x	0.0	0.5	1.0	1.5	2.0	2.5
$f(x)$	0.0	0.19146	0.34134	0.43319	0.47725	0.49379

jupyter check Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

[10]:
```
import numpy as np

x_values = [0.0, 0.5, 1.0, 1.5, 2.0, 2.5]
y_values = [0.0, 0.19146, 0.34134, 0.43319, 0.47725, 0.49379]

n = len(x_values)
diff_table = np.zeros((n, n))
diff_table[:, 0] = y_values

for j in range(1, n):
    for i in range(n - j):
        diff_table[i][j] = diff_table[i + 1][j - 1] - diff_table[i][j - 1]
x = 1.22
h = 0.5
p = (x - x_values[2]) / h

result = diff_table[2][0]
```

jupyter check Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] *

```
factorial = 1
p_term = 1
sign = 1

for k in range(1, 5):  # up to 4th order
    factorial *= k
    if k % 2 == 1:
        term = (diff_table[2 - k//2][k] + diff_table[2 - k//2 + 1][k]) / 2
        p_term *= (p ** k)
    else:
        term = diff_table[2 - k//2][k]
        p_term *= (p ** k)
    result += p_term * term / factorial

print(f"Estimated f(1.22) using Stirling's Interpolation: {result:.6f}")
```

```
Estimated f(1.22) using Stirling's Interpolation: 0.368787
```

Lecture 07:(Lagranges's Interpolation)

Class Work

Example2: Using Lagrange's interpolation formula find $y(2)$ from the following data

x	0	1	3	4	5
y	0	1	8	265	625

jupyter 2 Last Checkpoint: 4 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

[13]:
```
x=[0,1,3,4,5]
y=[0,1,8,265,625]
xv=2
def lagrange_interpolation(x,y,xv):
    n=len(x)
    result=0.0
    for i in range(n):
        term=y[i]
        for j in range(n):
            if i!=j:
                term *= (xv-x[j])/(x[i]-x[j])
        result += term
    return result
estimated_value=lagrange_interpolation(x,y,xv)
print(f"Estimated value of f(4): {estimated_value:.2f}")
```

```
Estimated value of f(4): -61.50
```

Example 5: By Lagrange's formula for inverse interpolation, determine the value of t when $A = 85$ given that.

x	2	5	8	4
y	94.8	87.9	81.3	68.7

```
[15]: x=[94.8,87.9,81.3,68.7]
y=[2,5,8,4]
xv=85
def lagrange_interpolation(x,y,xv):
    n=len(x)
    result=0.0
    for i in range(n):
        term=y[i]
        for j in range(n):
            if i!=j:
                term *= (xv-x[j])/(x[i]-x[j])
        result += term
    return result
estimated_value=lagrange_interpolation(x,y,xv)
print(f"Estimated value of f(4): {estimated_value:.2f}")
```

```
Estimated value of f(4): 6.47
```

Homework:

Exercise 1: Using Lagrange's formula Fit the polynomial of data, also find the value of y when $x = 2$.

x	0	1	2	4
y	-12	0	6	12

```
[17]: x=[0,1,2,4]
y=[-12,0,6,12]
xv=2
def lagrange_interpolation(x,y,xv):
    n=len(x)
    result=0.0
    for i in range(n):
        term=y[i]
        for j in range(n):
            if i!=j:
                term *= (xv-x[j])/(x[i]-x[j])
        result += term
    return result
estimated_value=lagrange_interpolation(x,y,xv)
print(f"Estimated value of f(4): {estimated_value:.2f}")
```

```
Estimated value of f(4): 6.00
```