Q#1

June 24, 2022

[1]:
```
# importing the modules
import math
from scipy import random
import numpy as np

# limits of integration
a = 0
b = 2 # gets the value of pi
N = 1000

# array of zeros of length N
ar = np.zeros(N)

# iterating over each Value of ar and filling
# it with a random value between the limits a
# and b
for i in range (len(ar)):
        ar[i] = random.uniform(a,b)

# variable to store sum of the functions of
# different values of x
integral = 0.0

# function to calculate the sin of a particular
# value of x
def f(x):
        return math.sqrt((math.cos(x) ** 2) + 1)

# iterates and sums up values of different functions
# of x
for i in ar:
        integral += f(i)

# we get the answer by the formula derived adobe
ans = (b-a)/float(N)*integral

# prints the solution
```

```
print ("The value calculated by monte carlo integration is {}.".format(ans))
```

```
The value calculated by monte carlo integration is 2.3538522114799214.
```

[2]:
```
# importing the modules
import math
from scipy import random
import numpy as np

# limits of integration
a = 2
b = 3 # gets the value of pi
N = 1000

# array of zeros of length N
ar = np.zeros(N)

# iterating over each Value of ar and filling
# it with a random value between the limits a
# and b
for i in range (len(ar)):
        ar[i] = random.uniform(a,b)

# variable to store sum of the functions of
# different values of x
integral = 0.0

# function to calculate the sin of a particular
# value of x
def f(x):
        return x**2

# iterates and sums up values of different functions
# of x
for i in ar:
        integral += f(i)

# we get the answer by the formula derived adobe
ans = (b-a)/float(N)*integral

# prints the solution
print ("The value calculated by monte carlo integration is {}.".format(ans))
```

```
The value calculated by monte carlo integration is 6.295701207999134.
```

[3]:
```
# importing the modules
import math
```

```
from scipy import random
import numpy as np

# limits of integration
a = 2
b = 3 # gets the value of pi
N = 1000

# array of zeros of length N
ar = np.zeros(N)

# iterating over each Value of ar and filling
# it with a random value between the limits a
# and b
for i in range (len(ar)):
        ar[i] = random.uniform(a,b)

# variable to store sum of the functions of
# different values of x
integral = 0.0

# function to calculate the sin of a particular
# value of x
def f(x):
        return math.sin(x ** 2)

# iterates and sums up values of different functions
# of x
for i in ar:
        integral += f(i)

# we get the answer by the formula derived adobe
ans = (b-a)/float(N)*integral

# prints the solution
print ("The value calculated by monte carlo integration is {}.".format(ans))
```

```
The value calculated by monte carlo integration is -0.08423574167271496.
```

[ ]: