Ujjwal

April 21, 2022

Monte Carlo Integration and convergence of error with increasing number of sample

Objectives: 1. calculate follwing integral using monte carlo method with number of samples 1, 10, 100, 1000, 10000, 100000
2. plot calculated integral against number of sample
3. plot error against number of sample

$$\int_0^{\pi} sin(x)dx$$

————-Julia implimentation —————-

Importing Library
Distrubutions : for generating random number with uniform probability distribution
Plots : for generating plots

[17]:
```
using Distributions
using Plots
```

[18]:
```
lowerLimit = 0
upperLimit = pi
```

[18]:
```
 = 3.1415926535897...
```

[19]:
```
# creating function with  imput: n(number of sample) and output : ans (approx␣
↪integral)
function approxIntegral(n)
        v = rand(Uniform(lowerLimit,upperLimit),n) # create random vector v of␣
↪n elements within given limits
        f_appliedTo_v = sin.(v)# apply sin function elementwise to vector v    ␣
↪
        integral = sum(f_appliedTo_v) # sum the elements of vector obtained␣
↪after applying function
        ans = (upperLimit - lowerLimit)/n * integral # calclulate ans
        return ans
end
```

[19]:
```
approxIntegral (generic function with 1 method)
```

actual value of integral is : 2.0

[20]:
```
trueVal = 2.0 # represents actual value of integral
function error(n) # function with input : n number of sample output error
        approxVal = approxIntegral(n) # generate approxval by calling functon␣
 ↪approxIntegral
        error = abs(approxVal - trueVal)/trueVal # calculate error
        return error
end
```

[20]:
```
error (generic function with 1 method)
```

1 . calclulate approx integral using Monte Carlo Method for the following samples 1, 10, 100 , 1000, 10000, 100000

[21]:
```
# creating sample vector N which store number of sample as components
N =[1 10 100 1000 10000 100000]
```

[21]:
```
1×6 Matrix{Int64}:
 1  10  100  1000  10000  100000
```

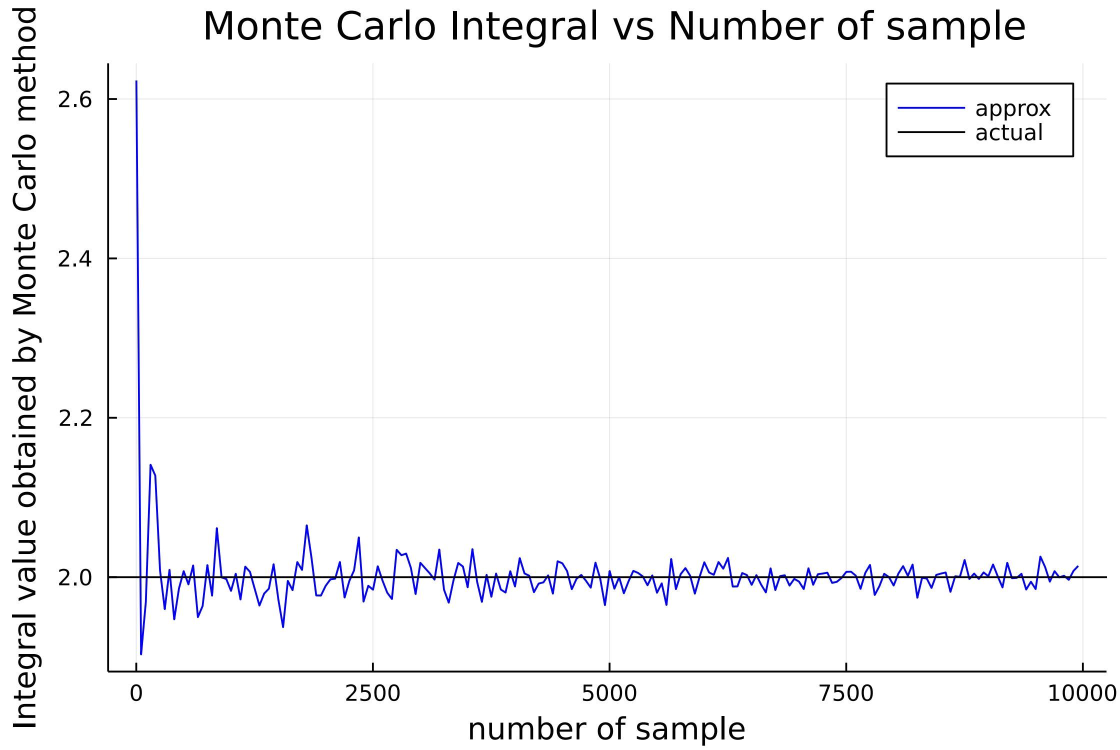
[22]:
```
#calculate integral at sample  elements N using approxIntegral method
 #display result in form of vector
approxIntegral.(N)
```

[22]:
```
1×6 Matrix{Float64}:
 2.21235  2.1986  1.85922  2.04067  1.99882  1.99775
```

2. plot approx integral vs number of sample

[23]:
```
n = 1:50:10000 # generate integer from 1 to 10000 with step size 50
plot(approxIntegral,n,title = " Monte Carlo Integral vs Number of sample",color␣
 ↪= "blue",label = "approx")
hline!([2],label = "actual",color = "black")
xlabel!("number of sample")
ylabel!("Integral value obtained by Monte Carlo method")
```

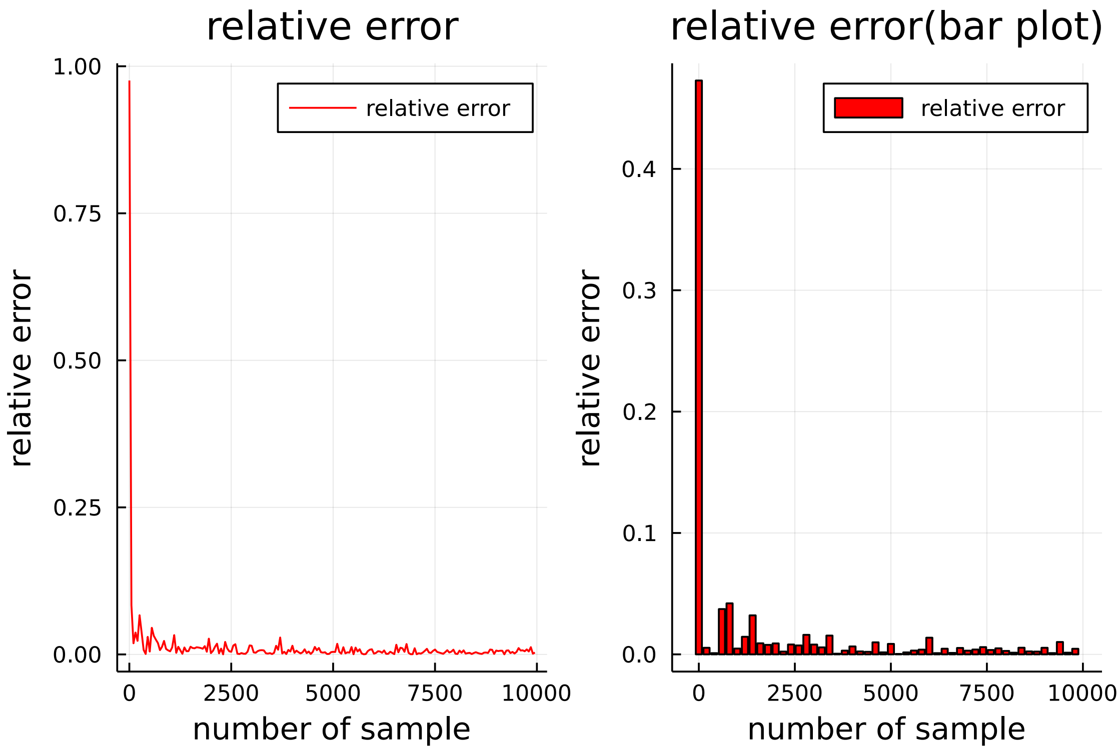
[23]:



3. plot error against number of sample

[24]:
```
n1 = 1:50:10000  # generate integer from 1 to 10000 with step size 50
p1  = plot(error,n1,title = " relative error ",color = "red",label = "relative error",xlabel = "number of sample",ylabel= "relative error")
n2 = 1:200:10000 # to generate clear bar chart
p2  = bar(error,n2,color = "red",xlabel = "number of sample",ylabel= "relative error",label = " relative error",title = "relative error(bar plot)")
plot(p1,p2,layout=(1,2))
```

[24]:



Created by Ujjwal