Program-M10

Devansh Shukla
I18PH021

a) Explain mathematical strategy to estimate π value under Monte Carlo method.

b) Write & execute a Fortran program to find π value in MC method. Plot a merged figure with dirt's inside and outsidethe targeted region along with the function (circle).

1 Theory

1.1 π Estimation: MC Method

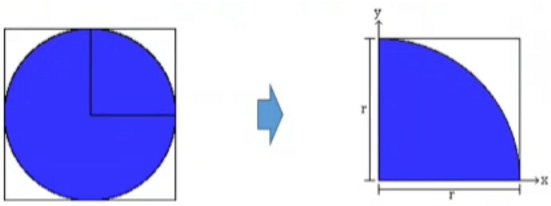
Here we utilize the "hit or miss" approach for estimating the value of pi. The hit-or-miss Monte Carlo method generates random points in a bounded rectangle and counts the number of 'hits' or points that are in the region whose area we want to evaluate.

π can be defined through the area of the circle: imagine a circle inscribed into the square with same diameter as the length of the square. We then calculate the ratio of no. of darts that hits inside the circle and total no. of darts.

Now for a very large number of generated points, this estimate approximates the value of π

$$\pi = \frac{\text{Area of the circle}}{r^2} \tag{1}$$

The denominator in Eq.(1) is simply the area of a square with side length '$2r$'.



Now, the same could be accompilished by considering only a quarter of the circle, the value of π is then approximated by Eq.(1.1)

$$\text{Probability, } P = \frac{\text{Area of the shaded region}(1/4\pi r^2)}{\text{Area of the square}(r^2)}$$
$$\pi = 4P$$
$$\boxed{\pi = 4\,\frac{\#\text{ of darts hitting shaded area}}{\#\text{ no of darts inside square}}} \tag{2}$$

2 Program Algorithm

NOTE: Blue-colored text represents variables in the algorithm, eg. variable.

1. Program open
2. Define x, y, pi, dist, i, hits, n, fmt.
3. Open a file "random_no.dat" with write access.
4. Get input from user for no of random numbers n
5. Open a do loop for index i from 0 to n with step 1.
6. Compute two random numbers, x and y between 0 and 1.
7. Check if $(x^2 + y^2) < 1$: if yes then increment the variable hits by 1.
8. Write i, x, y to file.
9. End-do loop
10. Compute the value of π using $\pi = 4\,\frac{\#\text{ of hits}}{\#\text{ of darts}}$ and write it to stdout.
11. Close file.
12. Program close

3 Program

3.1 Fortran program:

Using subroutine RANDOM_NUMBER, it returns a single pseudorandom number or an array of pseudorandom numbers from the uniform distribution over the range $0 \leq x \leq 1$ and implements a superior algorithm than the function RAND.

```
!==========================================
! pi_calc.f90
! Author: Devansh Shukla
!------------------------------------------
program pi_estimate

   implicit none
   ! Defining the variables
   real*8 :: x=0.0, y=0.0, pi=0.0, dist=0.0
   integer :: i=0, hits=0, n=0
   character(len=*), parameter :: fmt = "(I4 F10.3 F10.3)"

   ! Opening the data file
   open(unit=8, file="random_no.dat")

   ! Getting input from the user for total no. of darts
   print *, "Enter the total no. of darts (n)"
   read *, n
   print *, "------------"

   ! Do-loop for computing
   do i=0, n, 1
      ! compute the random numbers
      call RANDOM_NUMBER(x)
      call RANDOM_NUMBER(y)
      ! compute the distance
      dist = sqrt(x**2 + y**2)
      ! Write the computed random numbers to the data file
      write (8,fmt) i, x, y
      ! Check if the distance is less than or equal to one
      ! if yes then increment `hits` by 1
      if (dist .le. 1.0) then
         hits = hits + 1
      endif
   enddo

   ! Compute the value of `pi` and write it to `stdout` for the user
   pi = 4.0 * hits/n
   print "(xA,F6.4)", "pi=", pi
   print *, "------------"

   ! Close the data file
   close(8)

end program pi_estimate
```

3.2 Python program: Plots

```
#!/usr/bin/env python
"""
Author: Devansh Shukla
"""
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec

plt.style.use("rcStyleSheet.mplstyle")
mpl.use("pgf")
plt.ioff()

n=1000
df = pd.read_csv(f"random_no_{n}.dat", engine="python", delimiter=" ", header=None, skipinitialspace=True, comment="#")

fig = plt.figure(figsize=(4,4))
gs = gridspec.GridSpec(1, 1)
ax = fig.add_subplot(gs[0, 0])

ax.grid(True)
ax.set_aspect("equal")
```

```
ax.plot(df[1], df[2], "x", markersize=2.5, color="blue", label="dart")
ax.add_patch(plt.Circle((0, 0), 1, color='r'))
ax.set_xlim(0, 1)
ax.set_ylim(0, 1)
ax.set_xlabel(r"$x$")
ax.set_ylabel(r"$y$")
ax.legend(loc="upper right")
plt.title(rf"n={n}")

plt.suptitle(r"$\pi$-Estimation--Hit or miss")
fig.savefig(f"outputs/pi_est_dart_{n}.pdf")

fig = plt.figure(figsize=(4,4))
ax = fig.add_subplot(gs[0, 0])

n=2000
df = pd.read_csv(f"random_no_{n}.dat", engine="python", delimiter=" ", header=None, skipinitialspace=True, comment="#")

ax.grid(True)
ax.set_aspect("equal")
ax.plot(df[1], df[2], "x", markersize=2.5, color="blue", label="dart")
ax.add_patch(plt.Circle((0, 0), 1, color='r'))
ax.set_xlim(0, 1)
ax.set_ylim(0, 1)
ax.set_xlabel(r"$x$")
ax.set_ylabel(r"$y$")
ax.legend(loc="upper right")
plt.title(rf"n={n}")

plt.suptitle(r"$\pi$-Estimation--Hit or miss")
fig.savefig(f"outputs/pi_est_dart_{n}.pdf")

fig = plt.figure(figsize=(4,4))
ax = fig.add_subplot(gs[0, 0])

n=5000
df = pd.read_csv(f"random_no_{n}.dat", engine="python", delimiter=" ", header=None, skipinitialspace=True, comment="#")

ax.grid(True)
ax.set_aspect("equal")
ax.plot(df[1], df[2], "x", markersize=2.5, color="blue", label="dart")
ax.add_patch(plt.Circle((0, 0), 1, color='r'))
ax.set_xlim(0, 1)
ax.set_ylim(0, 1)
ax.set_xlabel(r"$x$")
ax.set_ylabel(r"$y$")
ax.legend(loc="upper right")
plt.title(rf"n={n}")

plt.suptitle(r"$\pi$-Estimation--Hit or miss")
fig.savefig(f"outputs/pi_est_dart_{n}.pdf")
```

4 Results

4.1 Terminal Output

4.1.1 n=1000

```
 ------------
 Enter the total no. of darts (n)
1000
 ------------
 pi=3.1440
 ------------
```

4.1.2 n=2000

```
 Enter the total no. of darts (n)
2000
 ------------
 pi=3.1360
 ------------
```

4.1.3 n=5000

```
 Enter the total no. of darts (n)
5000
 ------------
 pi=3.1488
 ------------
```

4.2 Data file

The data files have three columns: index i, x and y.
First 15 lines from the data files:

4.2.1 n=1000

```
 0    0.102    0.712
 1    0.666    0.043
 2    0.106    0.291
 3    0.909    0.583
 4    0.236    0.790
 5    0.222    0.442
 6    0.476    0.147
 7    0.455    0.884
 8    0.924    0.047
 9    0.060    0.509
10    0.995    0.535
11    0.629    0.272
12    0.032    0.288
13    0.736    0.882
14    0.404    0.265
```

4.2.2 n=2000

```
 0    0.039    0.253
 1    0.260    0.426
 2    0.216    0.395
 3    0.042    0.940
 4    0.143    0.370
 5    0.753    0.451
 6    0.097    0.145
 7    0.972    0.340
 8    0.844    0.503
 9    0.068    0.196
10    0.618    0.727
11    0.107    0.745
12    0.229    0.195
13    0.610    0.065
14    0.829    0.648
```

4.2.3 n=5000

```
 0    0.428    0.130
 1    0.751    0.329
 2    0.882    0.599
 3    0.282    0.800
 4    0.798    0.127
 5    0.470    0.005
 6    0.767    0.450
 7    0.508    0.611
 8    0.885    0.952
 9    0.673    0.145
10    0.564    0.373
11    0.563    0.633
12    0.202    0.398
13    0.282    0.968
14    0.027    0.385
```

4.3 Plots

The red region shows the quarter circle of radius $R = 1$ and darts are represented by $\times$.

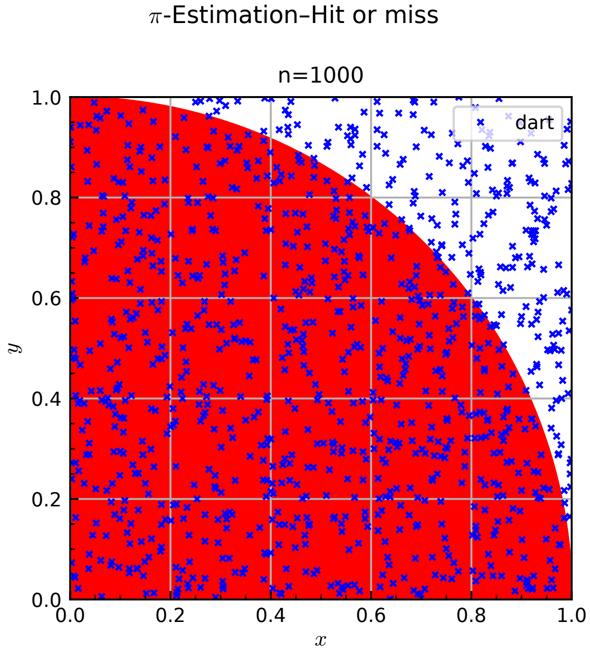


Figure 1: $n = 1000;\;\; \pi = 3.1440$

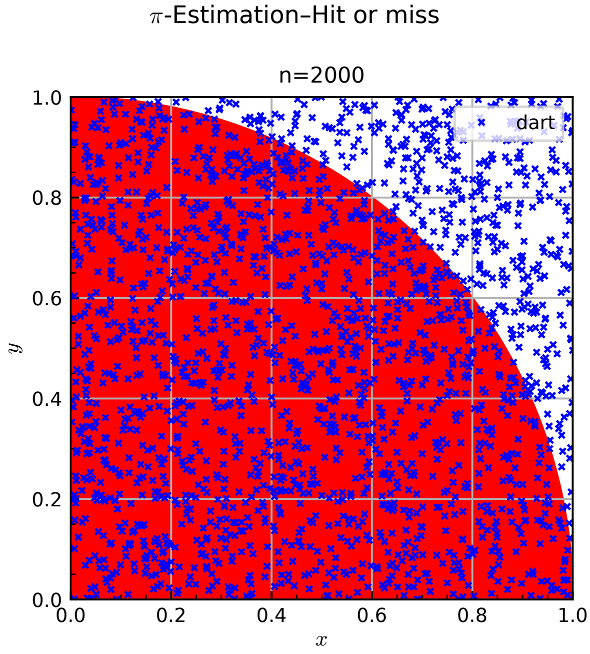


Figure 2: $n = 2000;\;\; \pi = 3.1360$

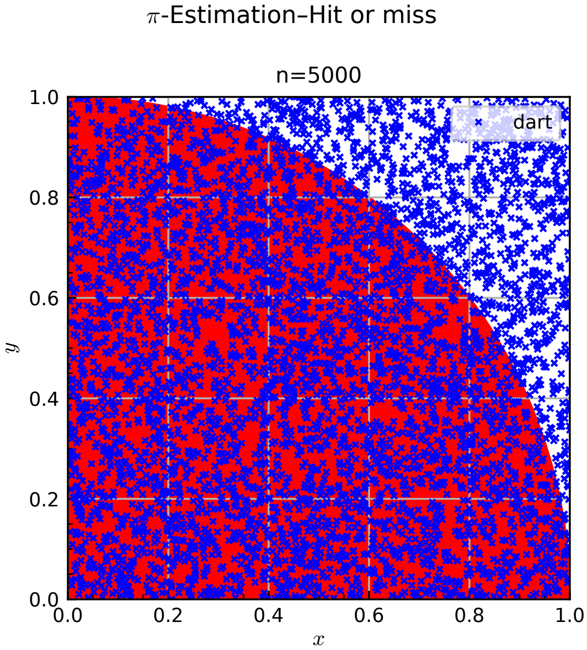


Figure 3: $n = 5000;\;\; \pi = 3.1488$

5 Remarks

The programs can be used to numerically estimate the value of π.

The parameters computed numerically are in agreement with the actual value of π.