Project #5: Computing π Numerically

Submit your program and results via email.

1. Task 1 tegrating the fol

```
student15@ittl-15:~/Documents/Project5$ ./project5_1
Enter the number of sub-intervals: 100
Approximate π using 100 sub-intervals: 3.141600986923125
Actual π: 3.141592653589793
Absolute Error: 0.000008333333332
student15@ittl-15:~/Documents/Project5$ ./project5_1
Enter the number of sub-intervals: 10000
Approximate π using 10000 sub-intervals: 3.141592654423134
Actual π: 3.141592653589793
Absolute Error: 0.000000000833341
student15@ittl-15:~/Documents/Project5$
```

   (a) t

   (b) t

   (c) c accurate r

2. Task 2: parallelize your sequential code using MPI.

   (a) Put together the code pieces in my slides to compute the value of π in parallel using MPI. Requirements:

   (b) Understand the usage of the four must-have MPI functions in the code.

   (c) Understand the usage of `MPI_Bcast` and `MPI_Reduce`.

   (d) Compile the code using `mpicc`. Make sure your code compiles.

   (e) Try to run the code using `mpirun`. Make sure your code runs.

   (f)

```
Sub-intervals (n): 10
Computed π     = 3.14242598500110
```

```
Sub-intervals (n): 100
Computed π     = 3.14160098692312
```

```
Sub-intervals (n): 1000
Computed π     = 3.14159273692313
Actual π       = 3.14159265358979
```

erical integration) 10, 10^2, 10^3, 10^4, 10^5, 10^6. Tabulate t after

```
Sub-intervals (n): 10000
Computed π     = 3.14159265442312
Actual π       = 3.14159265358979
```

```
Sub-intervals (n): 100000
Computed π     = 3.14159265359812
Actual π       = 3.14159265358979
```

```
Sub-intervals (n): 1000000
Computed π     = 3.14159265358990
Actual π       = 3.14159265358979
```

solutions should be more accurate with more sub-intervals.

```
student15@ittl-15:~/Documents/Project5$ mpirun -np 1 ./project5_2 1000000
Sub-intervals (n): 1000000
Computed π     = 3.14159265358976
Actual π       = 3.14159265358979
Absolute Error = 0.00000000000003
Time Elapsed   = 0.006023 seconds
```

```
student15@ittl-15:~/Documents/Project5$ mpirun -np 2 ./project5_2 1000000
Sub-intervals (n): 1000000
Computed π     = 3.14159265358994
Actual π       = 3.14159265358979
Absolute Error = 0.00000000000015
Time Elapsed   = 0.003888 seconds
```

```
student15@ittl-15:~/Documents/Project5$ mpirun -np 3 ./project5_2 1000000
Sub-intervals (n): 1000000
Computed π     = 3.14159265358991
Actual π       = 3.14159265358979
Absolute Error = 0.00000000000012
Time Elapsed   = 0.002311 seconds
```

be the same up to the machine error regardless of the

   (h) Use the same $n = 10^6$ with different number of processes (np = 1, 2, 3, 4). Use `MPI_Wtime` to obtain the time cost of your program. The `MPI_Wtime` function should not include the I/O statements. Draw a speed-up curve similar to the one in my slides.