Parallel Computing

2022/2023 1st semester

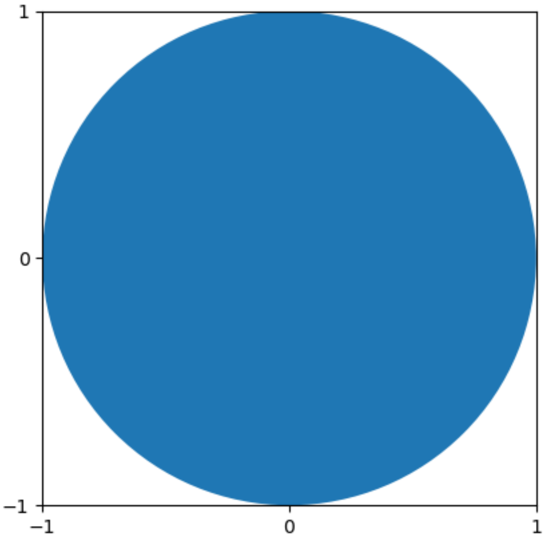
Exercise Sheet #2 28-09-2022

Monte-Carlo simulation to obtain the numerical value of π

Create a python program that creates n random points with $x, y \in [-1, 1]$, determines the fraction of points falling inside the inlying circle with radius 1, and calculates an approximated value for π. Do not use any trigonometric functions, as this would presuppose the knowledge of π's value. Use only basic arithmetic operations.
Compare the error of this approximation with the expected one.
The actual value of n should be taken as a parameter from the command line:

```
python3 myprog.py 1000000
```

For this to work you have to add, at the top of your python program, the following code:



```
import sys
n = int(sys.argv[1])
```

To get a random number $x \epsilon$ $[0, 1]$ you can use:

```
from random import random
x = random()
```

1. Use a Python program without any parallelization ("single core") to simulate $n = 10^9$ points, and compare the results with the ones expected by Estimation Theory. Measure the real time the program took to complete.
2. Create a parallel code in order to run the same exercise in p parallel processes, $n = 10^9$ points as well.

   Run it on your PC with the number of cores (i.e. threads) present, at least $p = 4$.

   n should again be obtained from the command line, and the program should determine the number of points $\frac{n}{p}$ to be created by each parallel process by obtaining the number of initiated processes through the corresponding method of `mpi4py`.

   Measure the real time the program took to complete, and compare it with the time needed for single core calculation on the same PC.

Submit the codes of your programs (1 até 2) at `https://trixi.coimbra.lip.pt/cap` **until the beginning of the next lectures (2022-10-19)**.

Put any comments and answers to the questions in a comment section of the source codes (free text inside a pair of three double quotes):

```
"""
Comment text with answers
"""
```