ME522: High-Performance Scientific Computing - Assignment 1

Dr Gaurav Bhutani

4th March 2024
(Due: 18th March 2024)

Problem: Monte Carlo Estimation of π using Fortran

You have to write a Fortran code to estimate the value of π using a Monte Carlo simulation

This is how the Monte Carlo method works:

Your program should generate a large number of random points within a unit square centered at the origin. It should then count the number of points that fall within a circle of radius 0.5 centred at the origin. The ratio of the number of points in the circle to the total number of points in the square will converge to the ratio of the area of the circle to the area of the square (which is equal to $\pi/4$) as the number of points increases.

To solve this problem, you will need to use the Pythagorean theorem to check whether each point falls within the circle. You can use the *random_number*() function to generate random points.

The algorithm will work as follows: suppose you generate 10,000 random points within the unit square centred at the origin. You count 7,854 points that fall within the circle of radius 0.5 centred at the origin. The ratio of the number of points in the circle to the total number of points is 0.7854, which is approximately equal to $\pi/4$. Therefore, your estimate of π is $4 \times 0.7854 = 3.1416$.

Proceed as follows:

1. Write a Fortran function called *monte_carlo_pi(num_points)* that takes the number of points as an argument and returns the π estimate using the Monte Carlo method.

2. Write a subroutine $test\_pi(n, input, output)$ that takes in an array of values of the number of points ranging between 10 and 10^6 , then calls the *monte_carlo_pi*() for each of the number of points, and returns an array of π estimates. In addition, calculate the error for each case and print a message if the error is larger than 20% of the actual value of pi.

3. Write a main code called *monte_carlo.f90* that calls the subroutine *test_pi*() with a concrete array of values.

4. Each of the above cases should be included in separate .f90 files.

5. Create a Makefile with the following targets - (*) *output.txt* that is generated by running *monte_carlo.f90* with the output redirected to *output.txt*. (*) *monte.exe*, (*) Phony targets called clean, clobber and help with the expected output. (*) Other targets as needed.

6. Comment the code thoroughly.

7. Write another main code called *main_shell.f90* that when executed into *main_shell.exe*, takes the number of points as a command line argument and prints the estimate of π using *monte_carlo_pi*() function. It will be executed on the terminal as follows: '*main_shell.exe* 1000' which will print the π estimate for 1000 Monte Carlo points.

8. Write a Python script that plots a graph with the π estimate on the y-axis and number of Monte Carlo points on the x-axis. Your python script should call the above *main_shell.exe* executable many times to generate this data. You should notice a clear convergence in the value of π as n increases. The plots should be exported to a pdf file called *plots.pdf*.

9. Add two more targets to the Makefile: (*) main_shell.exe and (*) plots.

You will have to upload the following files on Moodle:

1. monte_carlo.f90

2. monte_carlo_pi.f90

3. test_pi.f90

4. Makefile

5. main_shell.f90

6. plot_python.py