Universität Zürich[UZH]

Exercise sheet I

Simon Scheidegger
simon.scheidegger@gmail.com
July 25th, 2017
Open Source Macroeconomics Laboratory – BFI/UChicago

Supplementary material for the exercises is provided in OSM_Lab/HPC_day1/code_day_1/supplementary_material

1. HPC basics

I have 100 CPU cores. My code is 0.4% serial code. What is maximum speed-up I can obtain?

2. Unix Basics

- Go to your **home** directory on MIDWAY (e.g. /home/simonsch)

- Create a folder named **Exercises_day1**

- in there, create a sub-folder denoted **Fortran** as well as a sub-folder named **CPP**

- go to your favourite programming language's folder.

- Write a little program in that language that reads in your name from the terminal and write “Hello YOURNAME, how are you”.

- copy this file to your local laptop/desktop by scp.

- compile the code, create an executable that is called “hidiho.exec”.

- Hard-code YOURNAME into your *.cpp/*.f90 file.

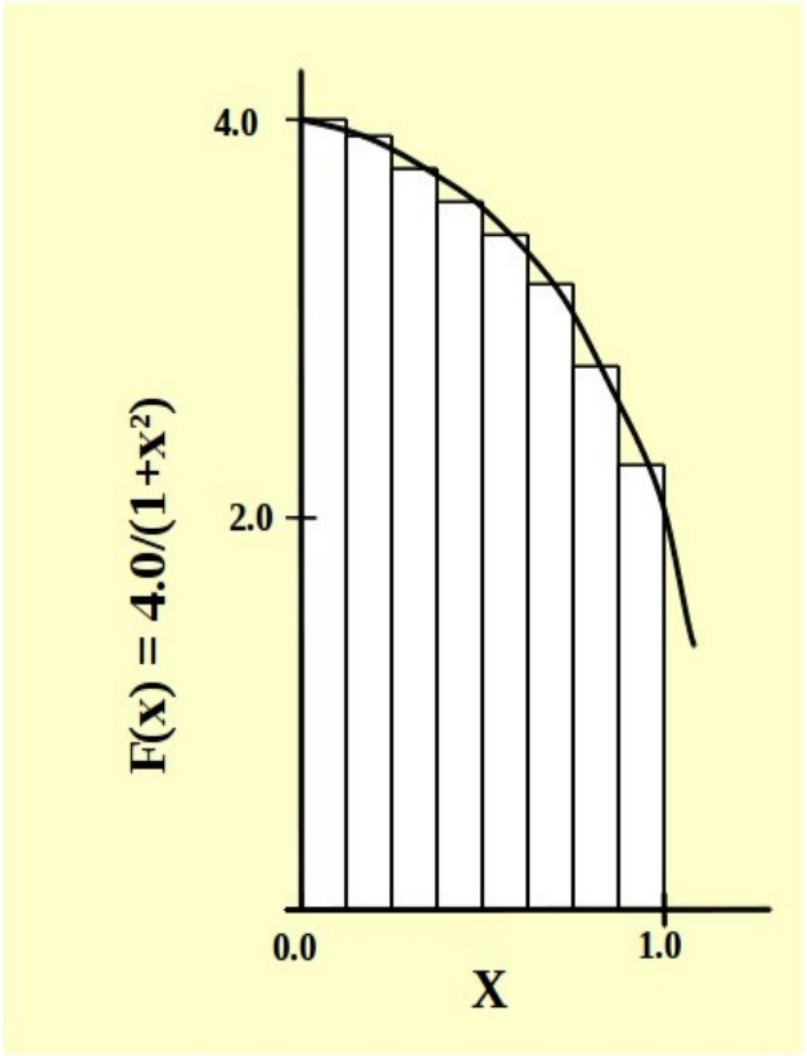
- Adjust a **slurm_job.sh** file such that you can submit the executable in batch mode.

3. Quadratic Equation

Write a program in Fortran or CPP that reads arbitrary values a, b, c from stdin and prints the solution to the quadratic Equation.

$$ax^2 + bx + c = 0$$

4. Compute Pi with finite differences



Mathematically, we know that:

$$\int_0^1 \frac{4.0}{(1+x^2)}\,dx = \pi$$

We can approximate the integral as a sum of rectangles:

$$\sum_{i=0}^{N} F(x_i)\Delta x \approx \pi$$

Where each rectangle has width Δx and height $F(x_i)$ at the middle of interval i.

Solution algorithm in CPP (2)

```
static long num_steps = 100000; double step;
void main ()
{
    int i;
    double x, pi, sum = 0.0;

    step = 1.0/(double) num_steps;

    for (i=0; i<num_steps; i++)
    {
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
    }
    pi = step * sum;
}
```

4a. Write a makefile & slurm file

Once you have written a running example that computes the value of Pi, write a makefile that compiles the source.

The files in the folder folder OSM_Lab/HPC_day1/code_day_1:

→ makefile_helloworld_cpp
→ makefile_helloworld_f90

provide some help.

Next, write a job submission script that runs this example with 1 CPU
on MIDWAY.

The file **submit_midway.sh** in the folder folder OSM_Lab/HPC_day1/code_day_1

5. Estimating Pi with Monte Carlo

Let's consider the problem of estimating Pi by utilizing the Monte Carlo method. Suppose you have a circle inscribed in a square (as in the figure). The experiment simply consists of throwing darts on this figure completely at random (meaning that every point on the dartboard has an equal chance of being hit by the dart). How can we use this experiment to estimate Pi?

The answer lies discovering the relationship between the geometry of the figure and the statistical outcome of throwing the darts.

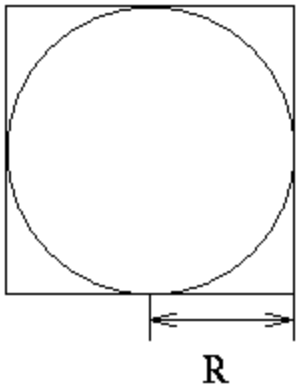
Let's first look at the geometry of the figure. Let's assume the radius of the circle is R, then the Area of the circle = Pi R^2 and the Area of the square = 4 R^2.

Now if we divide the area of the circle by the area of the square we get Pi / 4.

But, how do we estimate Pi by simulation? In the simulation, you keep throwing darts at random onto the dartboard. All of the darts fall within the square, but not all of them fall within the circle. Here's the key. If you throw darts completely at random, this experiment estimate the ratio of the area of the circle to the area of the square, by counting the number of darts in each.

Our study of the geometry tells us this ratio is Pi/4. So, now we can estimate Pi as

Pi = 4 x (Number of Darts in Circle) / (Number of Darts in Square).



→ Compute Pi according this algorithm.
→ Experiment with the number of random number you create (N = 100, 1,000, 10,000)
→ Run the code both in interactive as well as in batch mode.

6. Familiarize with the Projects

a) familiarize with the sample codes provided for project 1 and 3 (and re-visit 2 if needed)

→ You can form teams if you want.

b) compile both projects on MIDWAY.

c) write a slurm.sh submission file for MIDWAY.

d) Starting from the European option case, implement the ASIAN option.