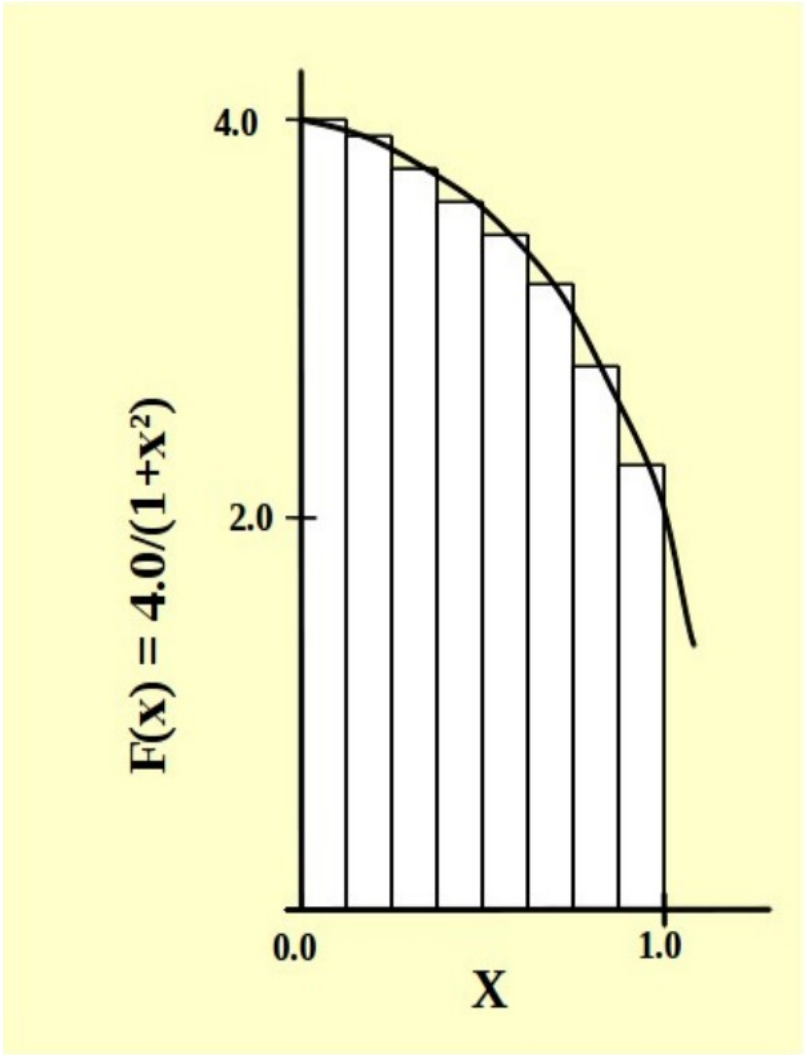
Universität Zürich[UZH]

Exercise sheet 4

Simon Scheidegger
simon.scheidegger@gmail.com
August 3rd, 2017
Open Source Macroeconomics Laboratory – BFI/UChicago

Supplementary material for the exercises is provided in OSM_Lab/HPC_day4/supplementary_material

1. 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.

1. Hybrid computaton of Pi

1. go to OSM_Lab/HPC_day4/supplementary_material

> cd OSM_Lab/HPC_day4/supplementary_material

2. Have a look at the source code in Fortran/CPP that compute Pi OpenMP parallel.

3. Make the code hybrid parallel by adding MPI.

4. write a makefile that compiles the code in hybrid.

5. run the code with a slurm file in hybrid mode. Experiment with the settings/distribution of MPI processes and OpenMP threads.

   → **2 MPI processes & 10 OMP threads, 4 MPI & 5 Openmp, ...**

2. Scaling test – Discrete State DP

Go to the massively parallelized DSDP code (set nk = 36,000 – a big number)

/project2/osmlab/DP_MultComms

→ perform a fully-fledged hybrid scaling test.
→ 1 thread, 1 MPI process.
→ 1 node, 1 MPI process – max. threads/node (hybrid – comm not split).
→ 5 nodes (comm split).
→ 10 nodes (comm split.
→ 20 nodes (comm split).
(see how much you can request – be careful with the wall time requested)

→ **all on MIDWAY – please use the slurm file that allows for hybrid jobs.**

→ **Generate scaling plots, normalized to**
a) one CPU.
b) one Node.

3. Scaling test – Continuous State DP

Go to the massively parallelized growth model code

/project2/osmlab/growth_model (set the dimension of the model to 10d)

→ perform a fully-fledged hybrid scaling test.
→ 1 MPI process.
→ 1 node, max. # MPI processes/node (comm not split).
→ 5 nodes (comm split – one group per state).
→ 10 nodes (comm split – one group per state).
→ 20 nodes (comm split – one group per state).
(see how much you can request – be careful with the wall time requested)

→ **all on MIDWAY – please use the slurm file that allows for hybrid jobs.**

→ **Generate scaling plots, normalized to**
a) one CPU.
b) one Node.
c) how does scaling vary if you vary the sparse grid level / dimension)

4. Hybridize the options pricer

- Can you parallelize BS.cpp (both option flavours) in hybrid (mixed OpenMP/MPI)?

- If yes, perform a scaling experiment
  - 1 thread per process/ 1 MPI
  - 8 threads per process/ 1 MPI
  - 8 threads per process/ 2 MPI
  - 8 threads per process/ 4 MPI
  - 8 threads per process/ 8 MPI

- Vary the number of MC samples and generate scaling plots.
- How do the number of MC samples influence scalabilty?