P. Koumoutsakos, W. Petersen, P. Hadjidoukas
ETH Zentrum, CLT E 13
CH-8092 Zürich

Fall semester 2017

Set 8 - 2D Diffusion and MPI

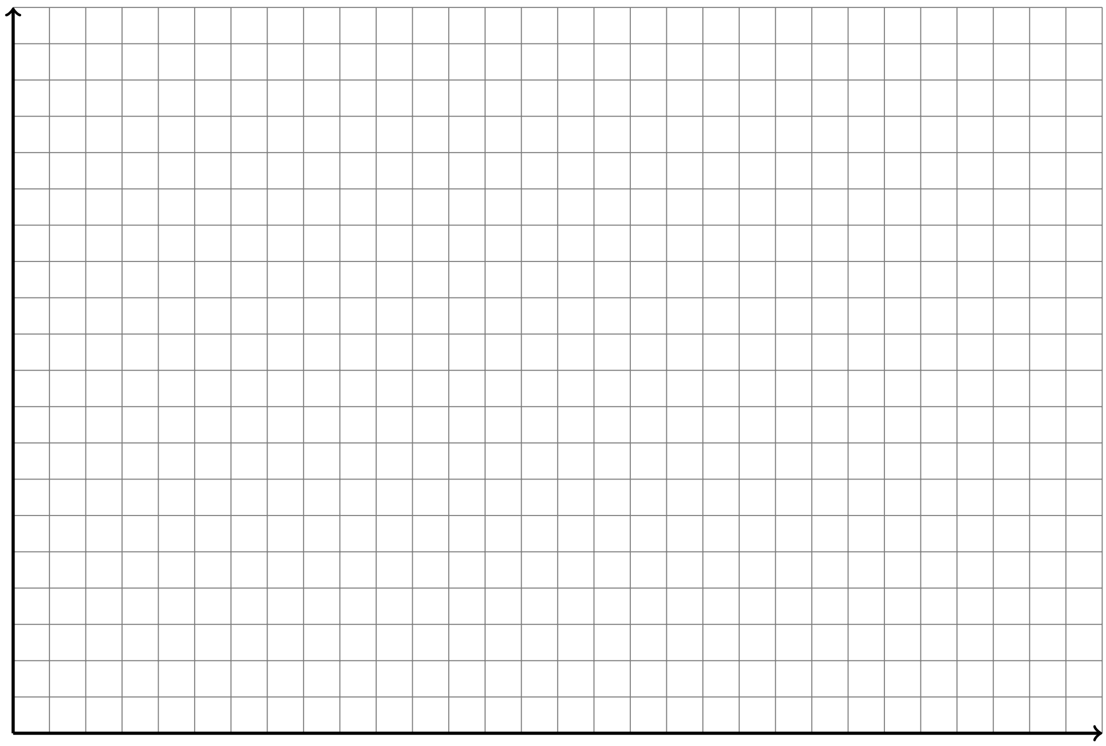
Issued: November 17, 2017
Hand in (optional): November 24, 2017 8:00am

Question 1: Parallel Scaling

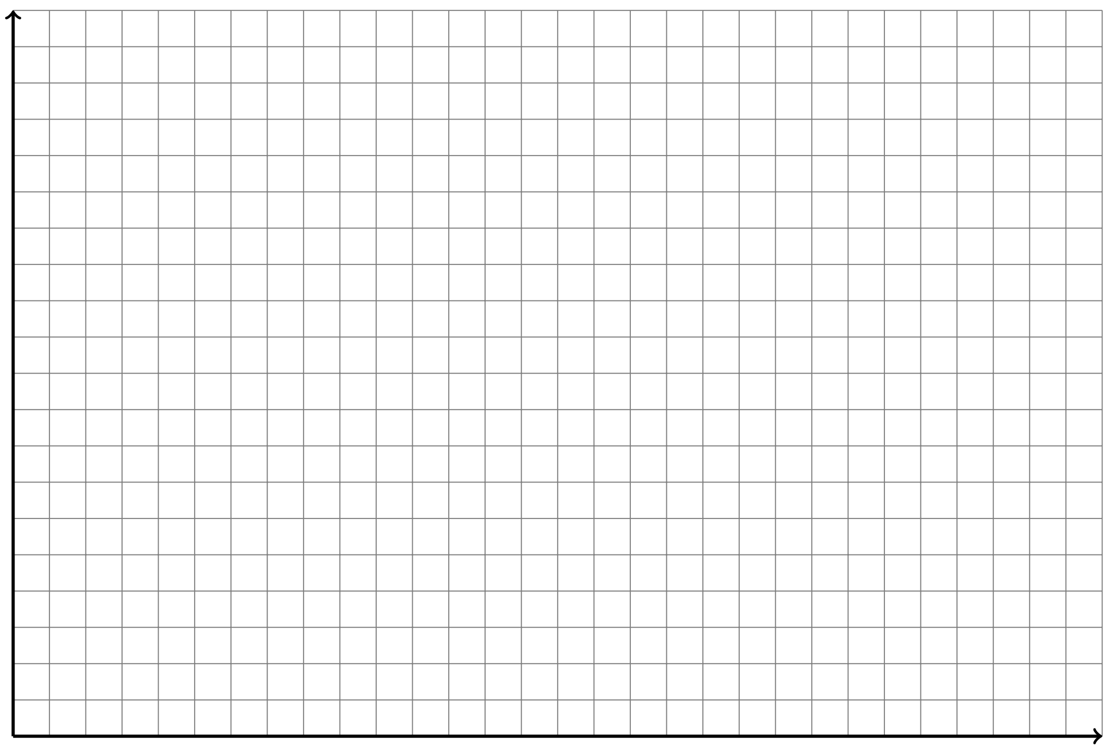
a) Assume we implemented a simple solver for simulating N particles. All particles interact with each other and, thus, the total computational cost at each time step is $O(N^2)$. The following table reports timing results for the solver for various number of particles N on P processor cores with a fixed number of time steps:

	runtime [s]			
P	$N = 500$	$N = 1000$	$N = 1500$	$N = 2000$
1	6.00	30.00	72.00	120.00
4	1.50	7.50	18.00	30.00
9	0.75	3.50	9.00	20.00
16	0.50	2.15	6.00	12.00
24	0.40	1.50	4.50	10.00

i) Draw in the figure below at least four points of the strong scaling plot for this program. On the solution sheet show all steps of your calculations. Do *not* forget to label the axes!



ii) Draw in the figure below at least four points of the weak scaling plot for this program, using the value for $N = 500$ at $P = 1$ as reference to estimate the parallelization overhead. On the solution sheet show all steps of your calculations. Do *not* forget to label the axes!



Question 2: Diffusion in 2D

Heat flow in a medium can be described by the diffusion equation of the form

$$\frac{\partial \rho(x,y,t)}{\partial t} = D\nabla^2 \rho(x,y,t) \tag{1}$$

where $\rho(x,y,t)$ is a measure for the amount of heat at position $\boldsymbol{r}$ and time t and the diffusion coefficient D is constant. Lets define the domain Ω in two dimensions as $x, y \in [-1, 1]$. We will use the boundary condition:

$$\rho(x,y,t) = 0 \quad \forall\, t \geq 0 \,\text{and}\, (x,y) \notin \Omega \tag{2}$$

and an initial distribution:

$$\rho(x,y,0) = \begin{cases} 1 & |x,y| < 1/2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

a) Discretize equation (1) using forward Euler in time and central differences in space and fill in all parts of the provided template code that are marked with `TODO:DIFF`, to implement the space and time evolution of $\rho(x,y,t)$.

b) Implement the MPI parallelization of the 2D diffusion equation by filling in all parts of the code marked by `TODO:MPI`. Decompose the domain using tiling decomposition scheme (described in the lecture notes). (i.e. distribute the rows evenly to the MPI processes).
   - *Note 1:* Study and become familiar with the provided OpenMP version of the code.
   - *Note 2:* Do not use non-blocking communication (which has not been discussed yet).

c) Compute an approximation to the integral of ρ over the entire domain in `compute_diagnostics`. Compare your result after 10000 iterations using the result of the provided OpenMP code that solves equation (1). To run the code use the parameters in Table 1.

d) Suggest other ways to divide the real-space domain between processes with the aim of minimizing communication overhead. Prove your argument by computing the message communication size for the tiling domain decomposition and for your suggestion.

e) (Optional) Make a strong and weak scaling plot up to 48 cores (two compute nodes on Euler).

Table 1: Example parameters.

	D	Ω	Δt
Set 1	1	128×128	0.00001
Set 2	1	256×256	0.000001
Set 3	1	1024×1024	0.00000001