P. Koumoutsakos, M. Troyer
ETH Zentrum, CLT F 12
CH-8092 Zürich

Fall semester 2014

Set 10 - Particle Strength Exchange with MPI

Issued: November 28, 2014
Hand in: December 5, 2014, 8:00am

We want to use the Particle Strength Exchange (PSE) method to simulate the diffusion equation,

$$\frac{\partial u(\mathbf{x},t)}{\partial t} = D\Delta u(\mathbf{x},t), \tag{1}$$

in a two dimensional domain $\Omega = [0,1]^2$ with periodic boundary conditions and $N \times N$ particles. The PSE method approximates the differential operator $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ with an integral operator:

$$\Delta u(\mathbf{x}) = \frac{1}{\varepsilon^2} \int_\Omega (u(\mathbf{y}) - u(\mathbf{x}))\, \eta_\varepsilon(\mathbf{y}-\mathbf{x})\, d\mathbf{y} + \mathcal{O}(\varepsilon^r), \tag{2}$$

where

$$\eta_\epsilon(\mathbf{x}) = (1/\epsilon^d)\eta(\mathbf{x}/\epsilon) \tag{3}$$

is a d-dimensional kernel, and the parameter r is called the order of the kernel.

We can thus write the diffusion equation as

$$\frac{\partial u}{\partial t}(\mathbf{x},t) \simeq \frac{D}{\varepsilon^2} \int_\Omega (u(\mathbf{y},t) - u(\mathbf{x},t))\, \eta_\varepsilon(\mathbf{y}-\mathbf{x})\, d\mathbf{y}. \tag{4}$$

We now discretize Equation **??**. We will introduce $N = n \times n$ particles placed in the nodes of a uniform mesh $\mathbf{x}_k = \mathbf{x}_{i,j} = (i \cdot \delta x + 0.5\delta x, j \cdot \delta y + 0.5\delta y)$ with concentrations $u_k^n = u(\mathbf{x}_k, n \cdot \delta t)$ and mesh widths $\delta x = \delta y = 1/n$. We then approximate the integral of the right-hand side with the quadrature sum over all the particles and use the Euler explicit time integration scheme:

$$\frac{u_k^{n+1} - u_k^n}{\delta t} = \frac{D}{\varepsilon^2} \sum_{\substack{p=1 \\ p \neq k}}^{N} \mathsf{vol}_p \left(u_p^n - u_k^n\right) \eta_\varepsilon(\mathbf{x}_p - \mathbf{x}_k), \tag{5}$$

where $\mathsf{vol}_p = \delta x \cdot \delta y$. Note that $\eta_\epsilon(\mathbf{x}) = \frac{1}{\epsilon^2}\eta\left(\frac{|\mathbf{x}|}{\epsilon}\right)$ for a two-dimensional settings.

Question 1: PSE with MPI

a) Complete the skeleton code provided in `skeleton.zip` such that the program uses MPI to simulate the diffusion with $M = M_x \times M_y$ processes, where each process works on its part of the domain. In particular, you should implement the PSE kernel evaluation method

which computes $\eta_\epsilon(\mathbf{x})$, the Diffusion class constructor and destructor as well as the `setIC` and `simulate` methods. You can add member variables and methods to help yourself.
Alternatively, you can implement the program from scratch and ignore the skeleton code.
For parallelization, you will need perform 2D domain decomposition on $M = M_x \times M_y$ MPI processes. Each process (r, s) is therefore responsible for the subdomain

$$\left[\frac{r}{M_x}, \frac{(r+1)}{M_x}\right] \times \left[\frac{s}{M_y}, \frac{(s+1)}{M_y}\right]. \tag{6}$$

The MPI processes can be organized in a two-dimensional Cartesian topology and every process must exchange data with its eight neighbors.
This particle code requires the notion of "ghost cells", where each MPI process sends the values of the boundary cells to its eight neighbors and receives the corresponding values from them respectively. The "ghost cells" form a halo, the depth of which is determined by the cutoff radius. Data exchange should be performed with non-blocking communication between the processes, implementing thus a computation/transfer overlap mechanism where ghost data are transferred while processes work on the inner part of their domain.
Use the following polynomial PSE kernel:

$$\eta(\mathbf{x}) = \frac{16}{\pi^2}\frac{1}{|\mathbf{x}|^8 + 1}. \tag{7}$$

Set the number of particles $N = n \times n$ to 120×120 and the time step as

$$dt = \frac{dh^2}{2D}. \tag{8}$$

Use $D = 1$ and $\varepsilon = 2\delta x = 2\delta y = 2\delta h$, δh being the interparticle distance. The cutoff for kernel evaluation should be set to 5ε. This implies that:

$$\eta_\varepsilon(\mathbf{x}_p - \mathbf{x}_k) = \begin{cases} 0, & |\mathbf{x}_p - \mathbf{x}_k| > 5\varepsilon = 10\delta h, \\ \eta_\varepsilon(\mathbf{x}_p - \mathbf{x}_k), & \text{otherwise.} \end{cases} \tag{9}$$

In other words, at most 20×20 particles around the observed particle affect its concentration.
As initial conditions, set

$$u(x, y, 0) = \sin(x \cdot 2\pi) \cdot \sin(y \cdot 2\pi), \tag{10}$$

such that the analytic solution is

$$u(x, y, t) = \sin(x \cdot 2\pi) \cdot \sin(y \cdot 2\pi) \cdot \exp(-8D\pi^2 t). \tag{11}$$

See code in `pse_mpi.cpp`.

b) Report the convergence of the solution - i.e. report L^1, L^2 and L^∞ errors (use analytic solution). For this you will need to use collective communication routine `MPI_Reduce`.
Errors are given by

$$L^\infty = 0.0760936, \qquad L^1 = 0.030825, \qquad L^2 = 0.0380468.$$

c) Make a strong and weak scaling plot using 1x1, 3x2, 6x4, and 6x6 cores.

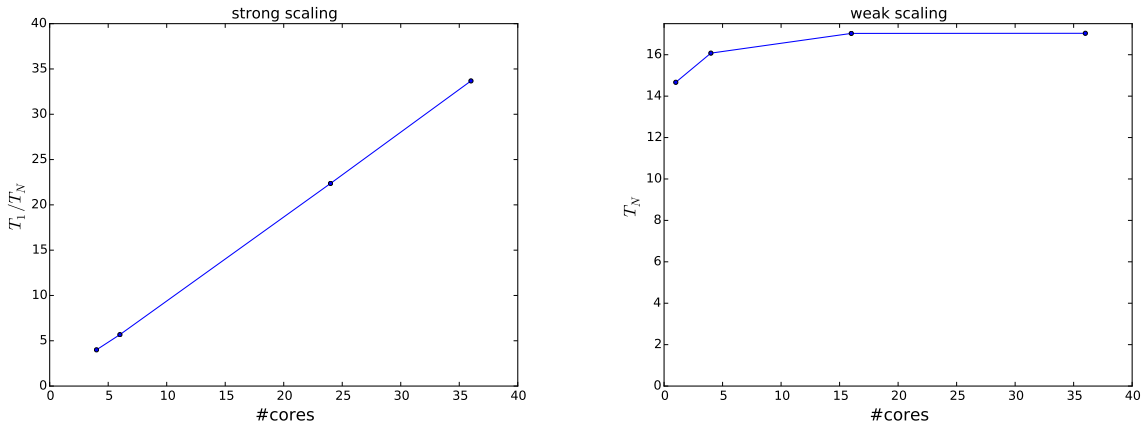

Figure 1: Strong and weak scaling for 1x1, 3x2, 6x4, and 6x6 cores.

Summary

Summarize your answers, results and plots into a PDF document. Furthermore, elucidate the main structure of the code and report possible code details that are relevant in terms of accuracy or performance. Send the PDF document and source code to your assigned teaching assistant.