ETH zürich

High Performance Computing for Science and Engineering I

P. Koumoutsakos
S. Martin
ETH Zentrum, CLT E 13
CH-8092 Zürich

Fall semester 2021

Set 4 - Diffusion, PSE & OpenMP

Issued: November 12, 2021
Hand in (optional): November 26, 2021 12:00

Question 1: Diffusion (25 points)

The diffusion of a substance can be described by the equation

$$\frac{\partial c(x,y,t)}{\partial t} = D\left(\frac{\partial^2 c(x,y,t)}{\partial x^2} + \frac{\partial^2 c(x,y,t)}{\partial y^2}\right), \tag{1}$$

where c is the concentration of the substance at position $(x,\, y)$ and at time t, and D is the diffusion constant. The diffusion process happens in the domain $|x| < L/2$ and $|y| < L/2$. The concentration is zero on the boundaries of the domain for $t \geq 0$. The initial concentration is

$$c(x,y,0) = \begin{cases} 1, & \text{if } |x| < L/4 \text{ and } |y| < L/4, \\ 0, & \text{otherwise.} \end{cases}$$

a) Write down the 2-dimensional discretized diffusion process for eq. (1). Assume a uniform grid with spacing h and a central finite difference scheme in space and forward Euler time integration. For the forward Euler integration assume time intervals of size dt. Make sure that you annotate all variables.

In the following subquestions, you will work with the codes provided in `/ex04/skeleton_code/q1/`. Please have a look at the README file for further information. We recommend compiling and running the code on the Euler cluster `https://scicomp.ethz.ch/wiki/Main_Page`.

b) Based on the discretization found in the previous subquestion, find the maximal timestep dt using the von Neumann stability analysis. Replace the hardcoded timestep ($t = 0.0001$) in the code with your solution.

c) Based on the discretization found in the first subquestion, provide a cache-friendly implementation of the diffusion equation in the method `advance`. I.e. avoid copying of memory if possible and mind the access patterns of the memory. Blocking must not be implemented.

d) Plot the total concentration as a function of time for $t \in [0, 0.5]$ using $D = 1$, $L = 2$ and $N = 100$. The concentration can be read from the file `diagnostics.dat` (column 0 and 1). Qualitatively explain the behaviour of the graph in less than 3 sentences, is this result expected?

e) Parallelize the diffusion process (your implementation from subquestion 1c) in the method `advance` using OpenMP.

f) Parallelize the integration of the concentration (marked with `TODO` in `compute_diagnostics`) and the calculation of the histogram (marked with `TODO` in the method `compute_histogram`) using OpenMP.

Question 2: Particle Strength Exchange (20 points)

Consider the diffusion equation eq. (1) from the previous exercise. We want to utilize the particle strength exchange (PSE[1]) method to solve this diffusion problem. Instead of discretizing the field $c(x, y, t)$ on a grid, we will use a collection of N particles. A particle represents a small "volume" of the field and is defined by its position $\mathbf{x}_i$ and field value $\phi_i = \pi_i(t)$. In this exercise, we assume the volume of each particle is equal $V_i = V_{total}/N = L^2/N$. We rewrite eq. (1) as a system of equations on particles:

$$\frac{\partial \phi_i}{\partial t} = \frac{D}{\epsilon^2} \sum_{j=1}^{N} V_j(\phi_j - \phi_i)\eta_\epsilon(x_j - x_i)\,, \tag{2}$$

where $\eta_\epsilon(r)$ is a kernel representing the Laplacian operator, and ϵ a scale constant. In this exercise we consider the following kernel:

$$\eta_\epsilon(\mathbf{r}) = \frac{4}{\epsilon^2 \pi} e^{-\frac{1}{\epsilon^2}|r|^2}. \tag{3}$$

Assume the same initial and boundary conditions as in the previous exercise.

In the following subquestions, you will work with the codes provided in `/ex04/skeleton_code/q2/`. Please have a look at the README file for further information. We recommend compiling and running the code on the Euler cluster.

a) You are given a skeleton code that initializes the particle positions and values. Get familiar with the skeleton code. Use `make run` and `make plot` to run the code and the scripts.

b) Extend the provided skeleton code to compute the right-hand side of eq. (2) using eq. (3) for the kernel η_ϵ. Reuse pair-wise quantities and reduce the number of operations. Implement the forward Euler scheme for the integration of the eq. (2).

c) Considering the domain size and the number of particles, what is qualitatively the distance h between neighboring particles? Parameter ϵ determines the spread of the kernel. Run the code for different values of ϵ. Experiment with

- $\epsilon << h$,
- $\epsilon \approx h$,
- and $\epsilon >> h$.

What do you observe for different cases? You can visualize the output of your runs with `make plot`.

d) Parallelize the `timestep` method using OpenMP. For this, you need to modify the `Makefile` and include the library in your source code.

e) Measure and plot the runtimes of your programs (diffusion and PSE) using 1...12 threads. For benchmarking on the Euler cluster, you can run `make stat` in an interactive shell or launch a job with `make statjob` (or `make statjobfull`). For plotting the runtimes run `make plotstat`. Answer the following: Which code (diffusion or PSE) scales better and how do you measure that? Do you consider strong or weak-scaling?

[1]see lecture website for supplementary information on PSE