ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

High Performance Computing for Science and Engineering II

Spring semester 2017

Dr. P. Hadjidoukas, Prof. C. Papadimitriou
ETH Zentrum, CLT E 13
CH-8092 Zürich

Project # 2

Issued: 15.05.2017

- In these scripts, we outline five computational engineering problems. Choose one and work either individually or in a group of two people. Communication is allowed to the extent that you do not copy the work of other groups.
- You are encouraged to contact one of the TAs in order to arrange a meeting to discuss your chosen project. These meetings are meant for clarifying and detailing the projects, give early feedback on your approach, *not* for correcting your code.
- In evaluating your work, we will consider your ability to analyse the problem and the hardware at your disposal, to appropriately apply the principles taught during the whole HPCSE class, and to report your reasoning and findings.
- The report (including text, code, figures) needs to be emailed before the day of the exam. If working in a group of two, each student has to write an *individual* report.

Diffusion

Diffusion is the process that describes the spreading of a quantity of interest driven by its concentration gradient towards regions with lower density. In this project we will consider diffusion in a two-dimensional domain medium that can be described by the diffusion equation of the form:

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

where the diffusion coefficient D is a constant and $\rho(\mathbf{r},t)$ is a scalar quantity of interest in position $\mathbf{r}$ at time t. We will use Dirichlet boundary conditions:

$$\rho(0,y,t) = \rho(x,0,t) = \rho(1,y,t) = \rho(x,1,t) = 0 \qquad \forall t \geq 0 \tag{2}$$

and an initial density distribution:

$$\rho(x,y,0) = \sin(\pi x)\cdot\sin(\pi y) \tag{3}$$

for which the analytical solution is given by

$$\rho(x,y,t) = \sin(\pi x)\cdot\sin(\pi y)\cdot e^{-2D\pi^2 t} \tag{4}$$

In the following, we denote with n the index of time $t_n = n \cdot \delta t$ and by i and j the indices of the nodes of a uniform mesh $x_i = i \cdot \delta x$, $y_j = j \cdot \delta y$. Then the quantity of interest at position $(x_i,\ y_j)$ and time t_n is denoted as $\rho^n_{i,j}$. We consider a uniform grid such that $\delta h = \delta x = \delta y$.

In this project you will solve the diffusion equation with **2 out of 3** introduced different methods. For each, first prioritize time-to-code and use the analytical solution to validate your implementation. Then, analyse each method and implement the best parallelization strategy to reduce the time-to-solution.

Finite differences with Alternating Direction Implicit method (ADI)

Alternating Direction Implicit (ADI) is a finite difference scheme based on the idea of operator splitting. The ADI method applied to the diffusion equation (equation 1) is summarized in the following two steps:

Step 1 Implicit Euler in x direction; Explicit Euler in y direction:

$$\rho^{n+\frac{1}{2}}_{i,j} = \rho^n_{i,j} + \frac{D\delta t}{2}\left[\frac{\partial^2 \rho^{n+\frac{1}{2}}_{i,j}}{\partial x^2} + \frac{\partial^2 \rho^n_{i,j}}{\partial y^2}\right] \tag{5}$$

Step 2 Explicit Euler in x direction; Implicit Euler in y direction:

$$\rho^{n+1}_{i,j} = \rho^{n+\frac{1}{2}}_{i,j} + \frac{D\delta t}{2}\left[\frac{\partial^2 \rho^{n+\frac{1}{2}}_{i,j}}{\partial x^2} + \frac{\partial^2 \rho^{n+1}_{i,j}}{\partial y^2}\right] \tag{6}$$

The algorithm takes advantage of both implicit and explicit schemes and results in an integration scheme which is unconditionally stable and second order in space and time.

Moreover, the implicit integration step, which usually requires iterative methods in order to invert a matrix, consists in a tridiagonal system of equations which can be efficiently solved by the Thomas algorithm **?**. In order to use the latter method for the implicit step in x direction, we need to rewrite for each row j the implicit step:

$$\rho^*_{i,j} = \rho^n_{i,j} + \frac{D\delta t}{2\delta x^2}(\rho^*_{i-1,j} - 2\rho^*_{i,j} + \rho^*_{i+1,j}) \tag{7}$$

in the form $\mathbf{A}\rho^* = \rho^n$ where $\mathbf{A}$ is a tridiagonal matrix, ρ^* is the unknown vector of $\rho^*_{i,j}$ with $i = 0 \dots n$ and ρ^n is the solution vector of $\rho^n_{i,j}$ at time n and row j.

An implicit step of ADI in x direction therefore consists in solving m tridiagonal systems of equations, where m is the size of the grid in y direction.

For the implicit Euler step in y direction we proceed analogously by writing m systems of equations, where m is the size of the grid in x direction.

Particle Strength Exchange

Particle strength exchange (PSE) is a deterministic particle method to simulate the diffusion equation. PSE is based on two main components: function approximation and operator approximation. The function approximation smoothly approximates a field function $\rho(\mathbf{r})$ with a finite

set of discrete particles using an interpolation kernel $\zeta_\epsilon(\mathbf{r})$:

$$\rho(\mathbf{r}) \approx \sum_p w_p \zeta_\epsilon(\mathbf{r} - \mathbf{r}_p) \tag{8}$$

where p is the particle index, w_p is the particle weight and $\zeta_\epsilon(\mathbf{r}) = (1/\epsilon^d)\zeta(\mathbf{r}/\epsilon)$, with d being the number of dimensions of the problem. The kernel $\zeta(\mathbf{r})$ is selected such that a certain number of moments of $\delta(\mathbf{r})$ function are conserved, i.e:

$$\int \mathbf{r}^\alpha \zeta(\mathbf{r}) \mathrm{d}\mathbf{r} - \int \mathbf{r}^\alpha \delta(\mathbf{r}) \mathrm{d}\mathbf{r} \qquad \forall \alpha \in \{0 \dots p-1\} \tag{9}$$

where p is the order of the interpolation kernel.

The method approximates the differential operator (the Laplacian $\Delta\rho = \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2}$) with an integral operator using a kernel $\eta_\epsilon(\mathbf{r})$

$$\Delta_\epsilon \rho(\mathbf{r}) = \frac{1}{\epsilon^2} \int_\Omega (\rho(\mathbf{r}', t) - \rho(\mathbf{r}, t))\ \eta_\epsilon(\mathbf{r}' - \mathbf{r}) d\mathbf{r}' + \mathcal{O}(\epsilon^p) \tag{10}$$

where $\eta(\mathbf{r})$ is a kernel such that $\eta_\epsilon(\mathbf{r}) = (1/\epsilon^d)\eta(\mathbf{r}/)\epsilon$. We can thus write the diffusion equation as

$$\frac{\partial \rho(\mathbf{r}, t)}{\partial t} \approx \frac{D}{\epsilon^2} \int_\Omega (\rho(\mathbf{r}', t) - \rho(\mathbf{r}, t))\ \eta_\epsilon(\mathbf{r}' - \mathbf{r}) d\mathbf{r}' \tag{11}$$

We introduce $N = n \times n$ particles placed on the nodes of an uniform mesh. We then approximate the integral of the right-hand side of the quadrature sum over all the particles and use the Euler explicit time integration scheme:

$$\rho_{i,j}^{n+1} = \rho_{i,j}^n + \frac{D\delta t}{\epsilon^2} \sum_{p=1, k \neq p}^{N} \delta x \cdot \delta y \cdot (\rho_p^n - \rho_{i,j}^n)\ \eta_\epsilon(\mathbf{r}_p - \mathbf{r}_{i,j}) \tag{12}$$

Consider the following two-dimensional kernel:

$$\eta(\mathbf{r}) = \frac{16}{\pi^2} \frac{1}{|\mathbf{r}|^8 + 1} \tag{13}$$

moreover, $\epsilon = 2\delta x = 2\delta y$. The cutoff for the kernel evaluation should be 5ϵ, in other words if $|\mathbf{r}_p - \mathbf{r}_{i,j}| > 5\epsilon$ then the kernel $\eta_\epsilon(\mathbf{r}_p - \mathbf{r}_{i,j}) = 0$.

Hint: Take special care to enforce the boundary conditions.

Random walks

Perform the same simulation by tracking the Browninan motion of a set of M equally weighted particles without inertia.

Initial conditions look as follows:

$$m_{i,j}^0 = \left\lfloor \rho(x_i, y_i, 0) \cdot \frac{M}{\iint_\Omega \rho(x, y, 0) \mathrm{d}x \mathrm{d}y} \right\rfloor \tag{14}$$

At each time step, with probability $1-4\lambda$ each particle remains at the same position, otherwise the particle moves in the 4 possible directions with equal probability λ for each. Here, $\lambda = D\delta t/\delta x^2$ and, by the CFL condition, $\lambda < 1/2$. Thus each particle moves around the grid by at most one grid-point per time-step.
Hint: you can derive a random variable describing how many particles jump in a certain direction.

After each time-step, the approximation of the solution can be computed as:

$$\rho_{i,j}^n = \frac{m_{i,j}^n}{M} \iint_\Omega \rho(x, y, 0) \mathrm{d}x \mathrm{d}y \tag{15}$$

where $m_{i,j}^n$ is the number of particles on the node.

Apply the appropriate boundary conditions to compare the result with the previous two methods. Analyse the convergence of the method by varying the grid size and the number of particles M.