

Prof. Dr. P. Koumoutsakos
ETH Zentrum, CLT F12
CH-8092 Zürich

Project 3

Issued: November 26, 2013
Hand in: December 10, 2013

Particle Strength Exchange

Particle strength exchange (PSE) is a deterministic particle method to simulate the diffusion equation,

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

The method approximates the differential operator $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \dots$ with an integral operator:

$$\Delta u(x) = \frac{1}{\varepsilon^2} \int_{\Omega} (u(y) - u(x)) \eta_{\varepsilon}(y - x) dy + \mathcal{O}(\varepsilon^r). \quad (2)$$

Recalling the properties of η_{ε} from Project 2b, you can use the Taylor expansion and check, that the approximation is correct. We can thus write the diffusion equation as

$$\frac{\partial u}{\partial t}(x, t) \simeq \frac{D}{\varepsilon^2} \int_{\Omega} (u(y, t) - u(x, t)) \eta_{\varepsilon}(y - x) dy. \quad (3)$$

We now discretize Equation 3. We will introduce $N = n \times n$ particles placed in the nodes of a uniform mesh $\mathbf{x}_k = \mathbf{x}_{i,j} = (i \cdot \delta x, j \cdot \delta y)$ with concentrations $u_k^n = u(\mathbf{x}_k, n \cdot \delta t)$. 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:

$$\frac{u_k^{n+1} - u_k^n}{\delta t} = \frac{D}{\varepsilon^2} \sum_{\substack{p=1 \\ k \neq p}}^N \text{vol}_p (u_p^n - u_k^n) \eta_{\varepsilon}(\mathbf{x}_p - \mathbf{x}_k), \quad (4)$$

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

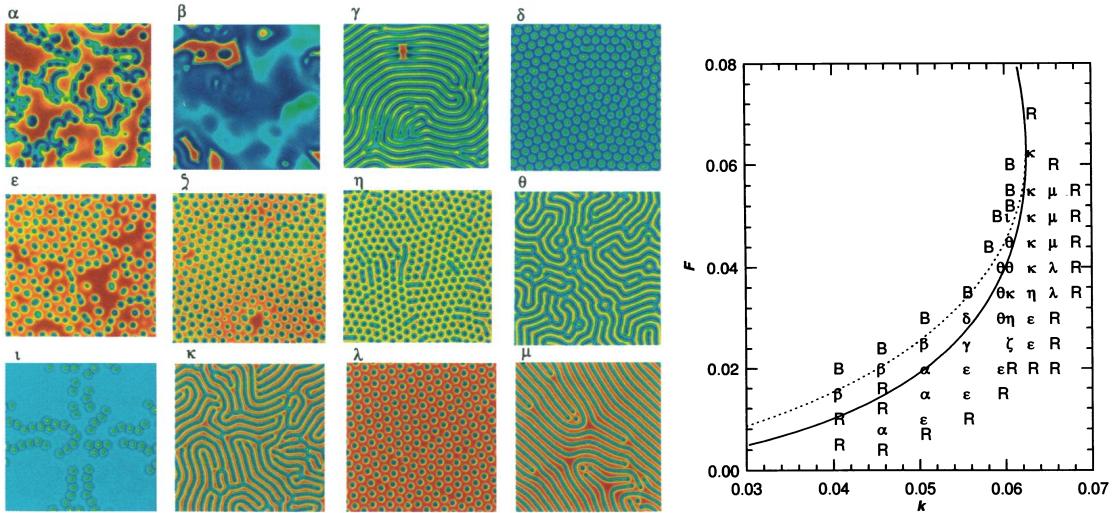


Figure 1: *On the left, different patterns obtained with Gray-Scott system. The Greek letters on the right figure indicate the locations in the parameter space which correspond to the patterns on the images on the left. Both figures are taken from "Complex Patterns in a Simple Systems" by J.E. Pearson.*

Reaction Diffusion

Reaction-Diffusion processes describe the evolution of concentrations of one or more substances distributed in a space. The substance concentrations are modified by chemical reactions and their distribution is governed by diffusion. The corresponding reaction-diffusion equation for M different species with respective concentration fields u_i and diffusion coefficients D_i has following form:

$$\frac{\partial u_i}{\partial t} = D_i \Delta u_i + f_i(\mathbf{u}), \quad \forall i = 1 \cdots M, \quad (5)$$

where $\mathbf{u} = (u_1, u_2, \dots, u_M)$. The terms f_i describe creation or consumption of species i caused by reactions. Note that since each concentration field u_i depends on the reaction term $f_i(\mathbf{u})$ (with possible different form for the term f_i), we are dealing with a system of coupled PDEs.

We will consider Gray-Scott system, a reaction-diffusion system consisting of two chemical species with respective concentrations u and v and corresponding diffusion terms D_u , D_v , which are subjects to the following reaction-diffusion equation:

$$\frac{\partial u}{\partial t} = D_u \Delta u - uv^2 + F(1 - u), \quad (6)$$

$$\frac{\partial v}{\partial t} = D_v \Delta v + uv^2 - (F + k)v. \quad (7)$$

The terms F and k are model parameters. The Gray-Scott system results in a formation of various patterns, see Figure 1 for few examples, where different pattern modes are obtain by different combination of values for model parameters F and k .

Particle Strength Exchange for Reaction-Diffusion Equation

We can use PSE to simulate the spatial component of the reaction-diffusion equation. Since now we are solving a system of coupled PDEs (Equation 5), each particle will carry a vector of concentrations, with one entry for each chemical species. Assuming the Gray-Scott system with two species with respective concentrations u , and v , given in Equations 6-7, we will define for each particle a vector of corresponding concentrations as:

$$\mathbf{u}_k^n = (u_k^n, v_k^n) = (u(\mathbf{x}_k, n \cdot \delta t), v(\mathbf{x}_k, n \cdot \delta t)). \quad (8)$$

Each particle carries the concentration values of all species in vector \mathbf{u}_k^n .

The diffusion can be simulated as described in Equation 4. We also observe that the reaction terms $f_i(u_k^n, v_k^n)$, where $i = 1, 2$ in the Gray-Scott system, only depend on (u_k^n, v_k^n) at the location of the corresponding particles. Reactions are thus purely local operations and require no particle-particle interactions at all. They can be evaluated independently for each particle. Thus the Gray-Scott system given by Equations 6-7 can be formulated in terms of the PSE method as follows:

$$\frac{u_l^{n+1} - u_l^n}{\delta t} = \frac{D_u}{\varepsilon^2} \sum_{\substack{p=1 \\ l \neq p}}^N \left(\text{vol}_p (u_p^n - u_l^n) \eta_\varepsilon(\mathbf{x}_l - \mathbf{x}_p) \right) - u_l^n (v_l^n)^2 + F(1 - u_l^n), \quad (9)$$

$$\frac{v_l^{n+1} - v_l^n}{\delta t} = \frac{D_v}{\varepsilon^2} \sum_{\substack{p=1 \\ l \neq p}}^N \left(\text{vol}_p (v_p^n - v_l^n) \eta_\varepsilon(\mathbf{x}_l - \mathbf{x}_p) \right) + u_l^n (v_l^n)^2 - (F + k)v_l^n. \quad (10)$$

Particle-to-Mesh Operations

Particle-to-Mesh (P2M) and Mesh-to-Particle (M2P) interpolations are a fundamental component of many particle methods. These operations enable us to work with both particles and a grid (mesh). M2P interpolations are used to transfer quantities from a grid to the particles while P2M are used for the opposite operations. In this project we will focus on the P2M interpolation which can be used to compute concentrations on a regular grid that can then be directly saved for visualization, where the mesh nodes correspond to pixels.

We discretize our domain $[0, 1]^2$ with a regular mesh of M -by- M grid points. Let $u_{i,j}^n$ be the concentration on the mesh at time $t = n\delta t$ at position $(x_i, y_j) = (i \cdot h, j \cdot h)$ where $h = 1/M$. Furthermore, we define u_p^n to be the concentration that particle p , located at (x_p, y_p) , carries at time $t = n\delta t$ for N distinct particles.

Then, given an interpolation kernel $W(\lambda)$, the value $u_{i,j}^n$ is given by

$$u_{i,j}^n = \sum_{p=0}^N u_p^n W(\lambda_i) W(\lambda_j), \quad (11)$$

where $\lambda_i = \frac{x_p}{h} - i$ and $\lambda_j = \frac{y_p}{h} - j$.

In this work you will implement and use the M'_4 kernel, which is given as

$$W(\lambda) = \begin{cases} 1 - \frac{5}{2}\lambda^2 + \frac{3}{2}|\lambda|^3 & \text{if } 0 \leq |\lambda| < 1, \\ \frac{1}{2}(2 - |\lambda|)^2(1 - |\lambda|) & \text{if } 1 \leq |\lambda| < 2, \\ 0 & \text{if } |\lambda| \geq 2. \end{cases} \quad (12)$$

Note that given the finite support of the interpolation kernel $W(\lambda)$ it is not computationally efficient to loop over all grid points and accumulate the contribution of each particle (gather operation). A more efficient implementation would iterate over all particles, compute their interaction range and only distribute their carried concentration on the affected mesh nodes (scatter operation).

Question 1: Particle Strength Exchange for Diffusion Equation

In this question you will solve the two-dimensional diffusion equation from Project 2. Consider the following two-dimensional kernel:

$$\eta(\mathbf{x}) = \frac{16}{\pi^2} \frac{1}{|\mathbf{x}|^8 + 1}. \quad (13)$$

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:

$$\eta_\varepsilon(\mathbf{x}_p - \mathbf{x}_k) = \begin{cases} 0, & |\mathbf{x}_p - \mathbf{x}_k| > 5\varepsilon = 10\delta h, \\ \eta_\varepsilon(\mathbf{x}_p - \mathbf{x}_k), & \text{otherwise.} \end{cases} \quad (14)$$

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

- a) We define the domain Ω in two dimensions as $x, y \in [0, 1]$. We will use periodic boundary conditions

$$\rho(0, y, t) = \rho(x, 0, t) = \rho(1, y, t) = \rho(x, 1, t) = 0 \quad \forall t > 0 \quad (15)$$

and an initial density distribution

$$\rho(x, y, 0) = \sin(x \cdot 2\pi) \cdot \sin(y \cdot 2\pi), \quad (16)$$

for which the analytical solution is given by

$$\rho(x, y, t) = \sin(x \cdot 2\pi) \cdot \sin(y \cdot 2\pi) \cdot e^{-8D\pi^2 t}. \quad (17)$$

Implement the PSE method (Equation 4) for this problem. Choose your time step δt size such as:

$$0.5 = \frac{D\delta t}{\delta h^2}. \quad (18)$$

- b) Report the spatial and temporal convergence of the solution. Compare the results with the ADI method and your implementation. In case you do not possess a correct implementation, use the solutions provided. Which one is more expensive? What are the advantages/disadvantages of the two methods with respect to each other?

In the spatial convergence (Figure 2) we observe that 2nd order is only reached for smaller problem sizes. Increasing ε from $2\delta x$ to $4\delta x$ we can observe that even for higher problem sizes we approximate the 2nd order slope. This deviation from the expected solution is due to the operator approximation error which depends on ϵ and is related to the choice of the kernel $\eta(x)$. In order to expose the spatial discretization error therefore we need to reduce both time discretization and operator approximation errors.

The more computationally expensive method is clearly PSE. It uses 20×20 values to update one, while ADI or any sort of method built upon second order finite difference approximation of the diffusion operator uses only 5. The operational intensity of PSE is usually much higher though, so using some high order compact kernel one might end up with a more accurate but not much more expensive method. Another disadvantage of the PSE is the stability condition, while ADI is unconditionally stable.

A major advantage of PSE is that you don't have to use the grid. You can place more particles in more "interesting" regions and less – where nothing really happens. Also you can approximate curved boundary conditions with higher order of accuracy than finite difference can provide you with.

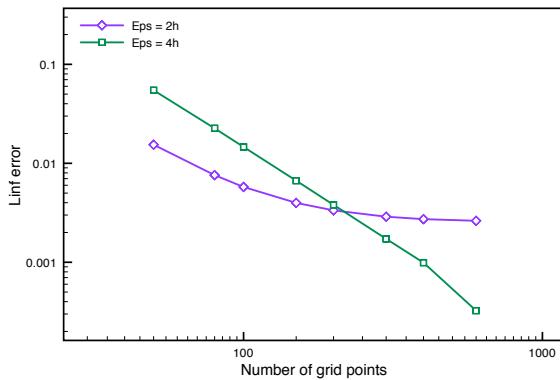


Figure 2: Convergence of the solution with different ε of the kernel. $\delta t = \frac{0.5}{N_{particles}}$, ran up to $t = 0.01$

- c) How would you employ the PSE method for Dirichlet boundary conditions? Discuss briefly.

For zero boundary condition you have to place so-called ghost particles outside the domain such that the particle strengths on the boundary vanish. For a flat boundary (linear) the ghost particles are mirrored with respect to the boundaries and their strengths are set to opposite of the original. This ensures the zero boundary conditions.

Question 2: Particle-to-Mesh for Visualization

Implement and use the P2M interpolation for visualization purposes. More specifically, interpolate the concentrations of each particle to a high-resolution regular grid of M -by- M points.

Extend your solver with a method that uses the P2M operator to interpolate the values from the particles to a grid with $M = 2n$, where n is the number of particles in a given direction. You can then use the method we provide in `write_png.c` to store your M -by- M grid in a PNG

image (you can look at the sample code provided for an example on how to interface with the function).

The interpolation kernel is used to interpolate the values to the grid cells directly surrounding the particle. In our case where the particles are also placed on a regular grid this means we might only get a sparse set of non zero points in case the higher resolution grids resolution is a multiple of the particle grid. If it is not a multiple of the particle grid the distribution will seem more even but probably still show "banding artifacts". These properties, while visually not very pleasing, are expected. The interpolation's important property is that it preserves concentration which is the case here. An example of the visualization is on the fig. 3.

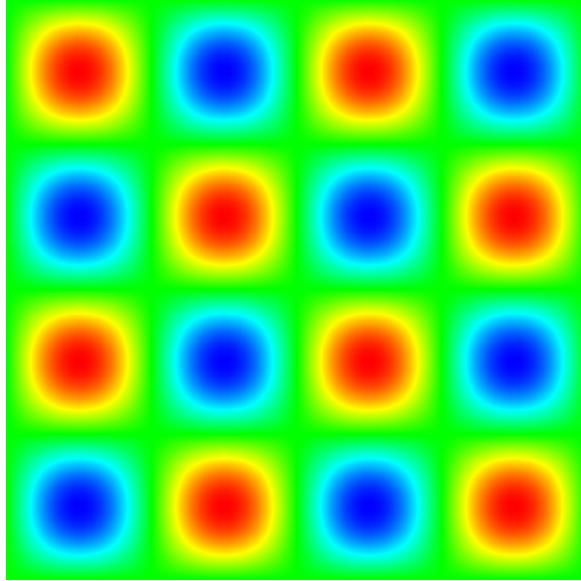


Figure 3: Visualization of the concentration in the domain $[-1, 1] \times [-1, 1]$. 1000 particles per dimension, 500 points per grid dimension

Question 3: Particle Strength Exchange for Reaction-Diffusion

Extend your implementation of PSE to handle the Gray-Scott system. Consider the simulation domain $[-1, 1]^2$, with periodic boundary conditions and initial conditions $(u, v) = (1, 0)$ everywhere except within the square $A = [-0.2, 0.2]^2$, which should be set to $(\frac{1}{2}, \frac{1}{4})$ and perturbed with $\pm 1\%$ random noise in order to break the square symmetry, i.e:

$$u(x, y, 0) = (1 - \chi_A(x, y)) + \chi_A(x, y) \left(\frac{1}{2} + \frac{r_1}{100} \right), \quad (19)$$

$$v(x, y, 0) = \chi_A(x, y) \left(\frac{1}{4} + \frac{r_2}{100} \right), \quad (20)$$

where r_1, r_2 are random numbers from a normal distribution $\mathcal{N}(0, 1)$ and χ_A is the characteristic function:

$$\chi_A(x, y) = \begin{cases} 1 & \text{for } (x, y) \in [-0.2, 0.2]^2, \\ 0 & \text{otherwise.} \end{cases}$$

The diffusion coefficients are $D_u = 2 \times 10^{-5}$, $D_v = 10^{-5}$ and time step dt is chosen according the stability condition

$$dt \leq \frac{dh^2}{2 \max\{D_u, D_v\}} \quad (21)$$

where dh is a spacing between particles.

- a) Consider the reaction rates $F = 0.03, k = 0.062$, which produce a pattern with spots. Extend and run your code with these parameters. Visualize the results with the method developed in the previous question.

Figure 4 shows concentration of chemical species u and v at time 20000 obtained by Gray-Scott system with parameters $F = 0.03, k = 0.062$. Evolution of this system over time is shown on movies `spotsU.mov` and `spotsV.mov`. As expected, selected values of model parameters lead to formation of patterns with spots for both species. In the Gray-Scott model, chemical u acts as an activator and v as an inhibitor, it means that areas with high concentration of chemical u correspond to areas with low concentration of chemical v and vice versa.

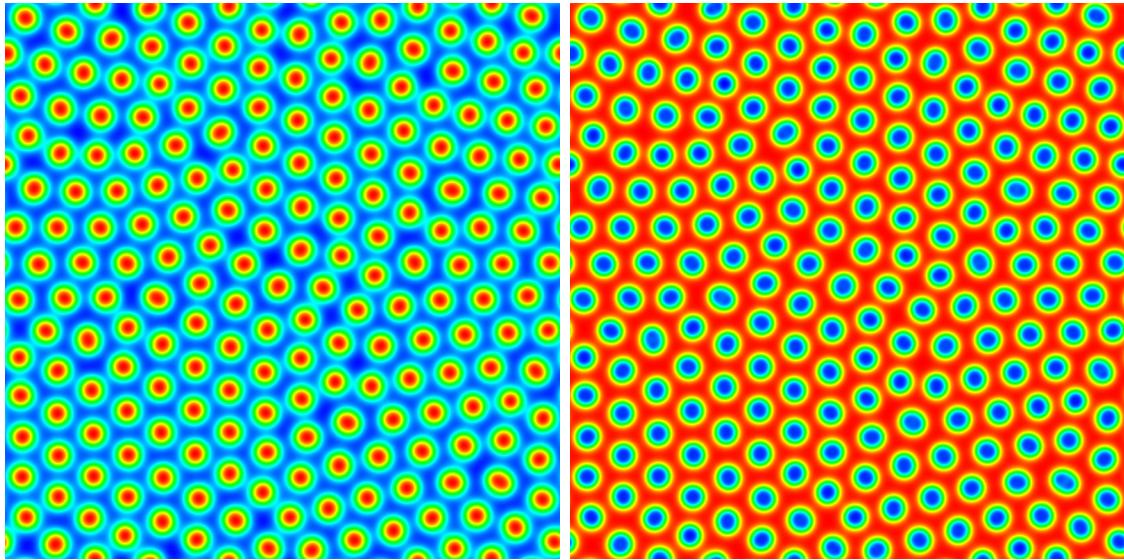


Figure 4: Gray-Scott with $F = 0.03, k = 0.062$ at time 20000. Concentration of u on the left, concentration of v on the right.

- b) Chose any combination of parameters F and k based on the parameter space in Figure 1. Visualize the results and report the used parameters. Can you get the expected pattern?

Figure 5 shows state of the Gray-Scott system with parameters $F = 0.03, k = 0.056$ at time 20000. This parameters corresponds to the pattern denoted as δ in the Pearson parameter map. Time evolution of this system is shown on movies `deltaU.mov` and `deltaV.mov`.

State of the Gray-Scott system with parameters $F = 0.04, k = 0.06$ at time 20000 is shown at figure 6. Chosen combination of model parameters correspond to the pattern denoted as θ in the Pearson parameter map. Time evolution of this system is shown on movies `thetaU.mov` and `thetaV.mov`.

Run your simulation sufficiently long so that you reach a steady state, meaning that you will have a pattern that is not changing with time. Final time $T_{end} = 200\,000$ should be a reasonable choice.

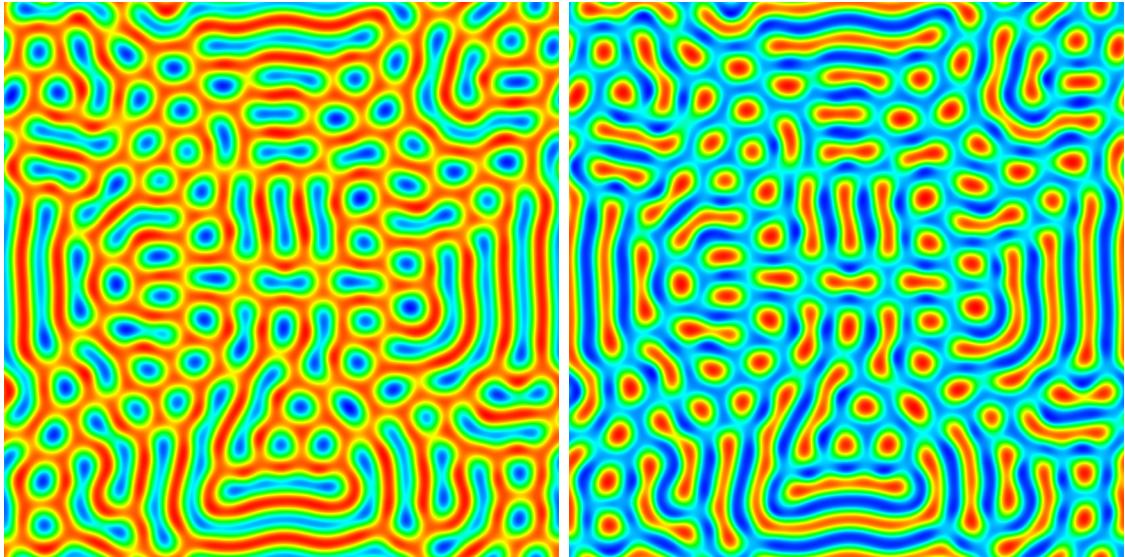


Figure 5: Gray-Scott with $F = 0.03$, $k = 0.056$ at time 20000. Concentration of u on the left, concentration of v on the right.

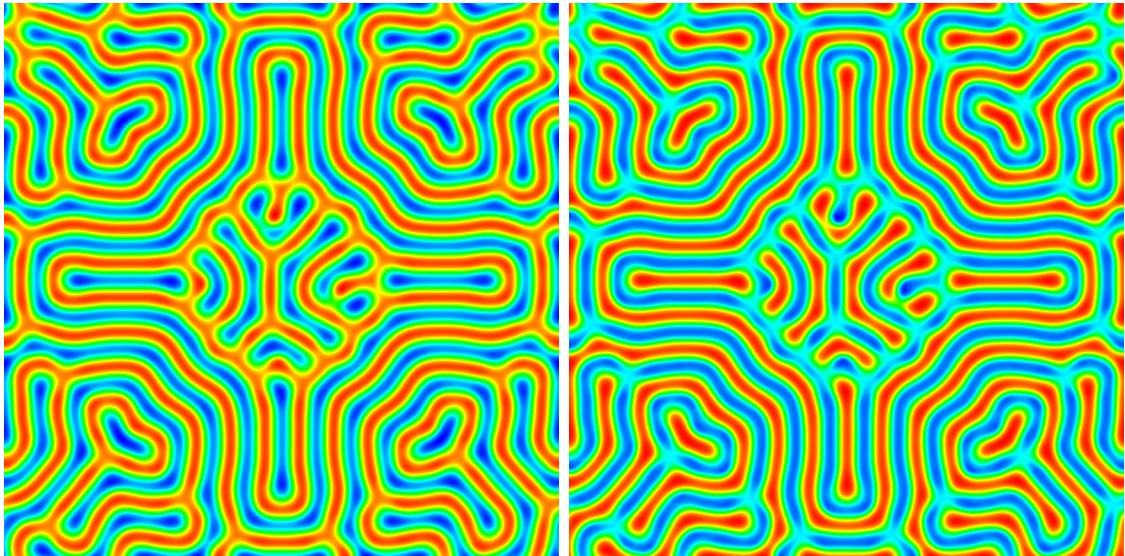


Figure 6: Gray-Scott with $F = 0.04$, $k = 0.06$ at time 20000. Concentration of u on the left, concentration of v on the right.

Question 4: PSE for Reaction-Diffusion with MPI

Implement a parallel version of your previously developed code using MPI. You will perform 2D domain decomposition on $M = M_x \times M_y$ MPI processes. Each process (r, s) is therefore responsible for the subdomain $[-1 + \frac{2r}{M_x}, -1 + \frac{2(r+1)}{M_x}] \times [-1 + \frac{2s}{M_y}, -1 + \frac{2(s+1)}{M_y}]$. The MPI processes can be organized in a two-dimensional Cartesian topology and every process must exchange data with its eight neighbors.

The 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.

- Following the above guidelines, extend your solver to do Reaction-Diffusion with MPI parallelism. You can validate your code by running diffusion only with the settings of Project 2.
 - Comment on your parallelization strategy. Suggest and discuss other ways to divide the real-space domain between processes with the aim of minimising communication overhead.
- The solution provided divides the domain into rectangular blocks and assigns one to each MPI process. An alternative is to split the domain into stripes, which reduces the number of communications and changes the amount of data communicated. This approach however does not scale as with the increasing of the problem size the amount of work per MPI rank also increases.
- Run your code with reaction rates $F = 0.03$ and $k = 0.062$, for $M = 1$ and for some value of M with both $M_x > 1$ and $M_y > 1$. Compare the results of the two cases.
 - Make a strong and weak scaling plot up to 96 cores (2 Brutus nodes).

The scaling plots are available in the fig. 7. The significant performance drop for more than 24 processes is probably due to the bandwidth contention, although additional study is required.

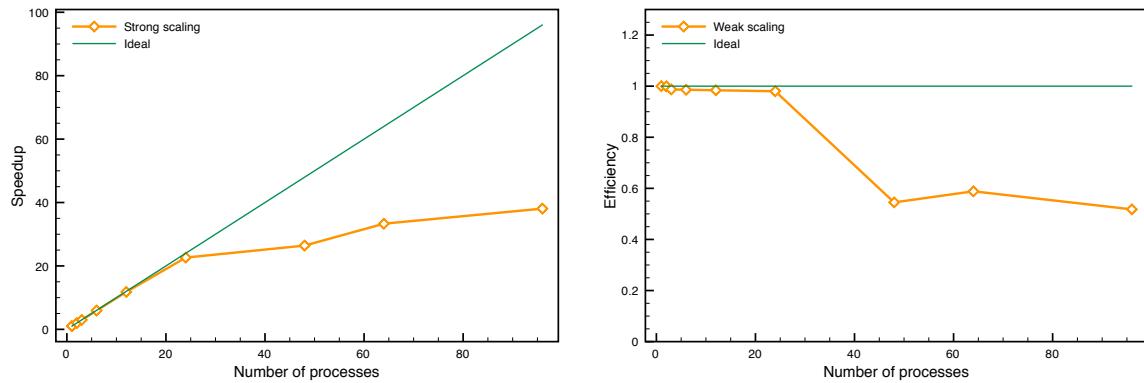


Figure 7: Strong scaling (left) and weak scaling (right). Used 1, 2, 3, 6, 12, 24, 48, 64 and 96 cores. $\delta t = 10^{-7}$, 100 steps, 1000 particles per dimension.

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, source code and related movies to your assigned teaching assistant.