Question 1: 1D-Diffusion problem

Consider the following problem:

\[
\frac{\partial u(x, t)}{\partial t} = \nu \cdot \frac{\partial^2 u(x, t)}{\partial x^2}
\]

\[
u (x, t = 0) = x e^{-x^2} \quad x \in [0, \infty)
\]

\[
u (x = 0, t) = 0 \quad t \in [0, \infty)
\]

To solve the diffusion equation you have been introduced to two pure particle methods, Random Walk (RW) and Particle Strength Exchange (PSE). The objective of this exercise is to implement the two methods and to validate your code. The exact (analytic) solution of the problem is:

\[
u^{ex}(x, t) = \frac{x}{(1 + 4 \nu t)^{3/2}} \cdot e^{-x^2/(1+4\nu t)}.
\]

Random Walk for diffusion in space

Please implement the Random Walk particle method in 1D. Test your code with the problem stated above! At \(x = 0\) we have the Dirichlet boundary condition \(u(x = 0, t) = 0\). By computing the solution on \(x \in [-X, X]\) this boundary condition is implicitly satisfied for all \(t\) due to the point symmetric nature of the initial concentration. At \(x = X\) we have an open boundary. Discuss the problems and effects of the boundary conditions!

PSE for diffusion in space

Please implement the Particle Strength Exchange (PSE) method in 1D! Solve the diffusion problem for the same initial and boundary conditions as above. Do a comparison of both methods as outlined in the slides and check the convergence of both methods.

Parameters:

- particle number: \(N = \{50, 100, 200, 400, 800\}\)
- domain: \(x \in [-X, X], \; X = 4\)
- interparticle spacing: \(h = 2X/(2N - 1)\)
- diffusion constant: \(\nu = 0.0001\)
- time step: \(\Delta t = 0.1\)
- integration time: \(T = t_{\text{max}} = 10\)
- kernel size: \(\epsilon = h\)
In the second exercise we simulate the diffusion process in 2D. For this purpose you have to extend your PSE implementation to the 2D case.

**Question 2: Isotropic Diffusion in 2D**

Write a program that simulates isotropic 2D diffusion with the PSE operator. The code should produce particles on a grid in the 2D unit square computational domain with 26 particles per dimension. The initial condition for the concentration $u$ is a Dirac’s delta function at $(x, y) = (\frac{1}{4}, \frac{1}{2})$ (and zero elsewhere). The initial set-up is depicted in Fig. 1.

![Initial conditions](image)

Figure 1: Initial set-up of the simulation

Simulate isotropic, homogeneous and normal diffusion with the diffusion constant $D = 2$.

a) The key function that you need to implement is `applyPSE.m`. The function reads like this:

```matlab
function pseSum = applyPSE(particleMat, verletList, epsilon, numStren)

% Code for Exercise 6 - 2D PSE Operator
% Input
% particleMat: (numParticles x (dim+1+numStren)) - Matrix of
% particle positions, cell indices and particle strengths
% verletList: Verlet list of particles
% epsilon: Kernel parameter epsilon (standard deviation)
% numStren: Number of different strengths a particle carries
%
% Output
% pseSum: ((numParticles x numStren) - Matrix of
% updated particle strengths
%
% function pseSum = applyPSE(particleMat, verletList, epsilon, numStren)
```
Read the script chapter on PSE again, and implement the PSE operator. For this, you need a kernel function $\eta$ that works in 2D. It reads:

$$\eta(x) = \frac{4}{\pi \epsilon^2} e^{-\frac{x^2}{\epsilon^2}}$$  \hspace{1cm} (1)

Use periodic boundary conditions, $\epsilon = h$, $\nu = 2$ and $dt = h^2/(3\nu) = 2.6667e^{-04}$ and $T = 1$.

When your code runs smoothly, increase the number of particles to $51^2$ and adapt the time step accordingly. Plot the evolution of the 2D concentration.

b) As soon as you have a running implementation, play with the time step parameter $dt$. For the forward Euler time stepping scheme the stability condition that relates the space discretization to the time step is $dt < h^2/(2\nu)$. Vary the $dt$ parameter such that the stability conditions are not satisfied any more. What effect has this on your simulation?