

EXERCISE 10 PARTICLE TRACKING

An alternative approach to solving the Convection-Dispersion equation

$$\begin{aligned} \partial_t c + \nabla \cdot (-\tilde{D} \nabla c + c \vec{v}) &= 0 & \vec{x} \in \Omega \\ c &= f & \vec{x} \in \partial\Omega_D \\ \vec{n} \cdot (-\tilde{D} \nabla c + c \vec{v}) &= J_N & \vec{x} \in \partial\Omega_N \end{aligned}$$

numerically is particle tracking:

For a given set of particles $P = \{p_0, \dots, p_N\}$ located within the computational domain resolved by a grid, we define a piecewise constant concentration on a grid cell E . Let $P_E \subset P$ be the subset of all particles contained in E , then we define the concentration c_E in E as

$$c_E = c_0 \frac{|P_E|}{|P|}.$$

Here c_0 corresponds to the maximum concentration reached when all particles are within the same element.

For a velocity field $\vec{v}(\vec{x})$ and an effective dispersion $\tilde{D}(\vec{x})$ which are constant in time, the movement of a particle which is initially at position $\vec{x}(t)$ at time t is computed via discrete time steps of size τ :

$$\vec{x}(t + \tau) = \vec{x}(t) + \int_t^{t+\tau} \vec{v}(\vec{x}(t')) dt' + \sqrt{2\tilde{D}(\vec{x}(t))} \tau \cdot \vec{Z}.$$

To represent the stochastic nature of dispersion we employ a random vector \vec{Z} with unit length and a mean value of zero. The integral, that approximates the convective movement in each step, is calculated by applying the midpoint rule:

$$\int_t^{t+\tau} \vec{v}(\vec{x}(t')) dt' \approx \vec{v}(\vec{x}(t)) + \vec{v}(\vec{x}(t)) * \left(\frac{\tau}{2}\right) \tau$$

It is advantageous to choose the time step individually for each particle based on a constant spatial step length Δx . Then - in addition to its position - the current time of each particle must be computed and stored in memory.

The individual time step of particle p at time t_p and position $\vec{x}_p(t_p)$ is given by:

$$\tau_p = \min \left(\frac{\Delta x}{\vec{v}(\vec{x}_p(t_p))}, \frac{(\Delta x)^2}{2\tilde{D}(\vec{x}_p(t_p))} \right).$$

To compute the concentration at a given time \tilde{t} all particle positions have to be known at this time. Therefore, if $\tau_p + t_p > \tilde{t}$ we choose $\tau_p := \tilde{t} - t_p$ instead.

The particle tracking algorithm is realized by a class `ParticleTracker` with the following public interface:

- The constructor:

```
ParticleTracker(
    const Grid & _grid,
    const ConvectionField & _convection_field,
    const DispersionField & _dispersion_field,
    std::vector<Particle> & _particles
)
```

The constructor receives the grid and objects representing the convection and dispersion field. Furthermore it receives the array of particles represented by objects of type `Particle` (see below)

- Implement a method

```
size_t step( const double step_dx, const double target_time )
```

that propagates particles with time steps based on the constant spatial step `step_dx`. All particles are propagated to a time $t_p \leq \text{target_time}$. The return value gives the number of the still moving particles that have not reached the target time yet.

The class `Particle` used to store particle information is given by:

```
class Particle
{
public:
    Vector position;
    double time;

    Particle()
    : position(Grid::dimensions,0),
      time(0)
    {}
};
```

Both the classes `DispersionField` and `ConvectionField` provide an operator `double operator() (Vector position) const` to access field values at given global coordinates in the domain.

Apply the particle tracker to the contamination example of exercise 8. To avoid dealing with ugly particle tracking boundary conditions, a uniform initial contribution of particles within the area

$$I = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \mid 0m \leq x \leq 3m, \quad 3m \leq y \leq 6, m \right\}.$$

is chosen. A simulation length of twenty days will be sufficient.

If particles leave the domain, you are free to let them go or otherwise let them stick to the point of collision with the boundary.

Start with a small number of particles (≈ 100) and increase their number (`./solution10 n`, where `n` is the number of particles) until the solution does not change significantly anymore.

5 Points