Diffusion Cross-Diffusion equations to model student distribution in a three type school system

20. August 2018

1 Model

1.1 equations

Let u_1 be the *density of privileged students*, while u_2 denotes the *density of non-privileged students.*

In this model three type of schools are considered: private-, public-, and a-third-kind-of schools. To model the long-term student-school distribution the schools are thought to have 'attracting fields', which spread over the domain and attract the two types of students. The variable v_{ij} correspond to the signal sent out by school type $j \in \{1, 2, 3\}$ to attract students of type $i \in \{1, 2\}$. Here $j = 1, 2, 3$ correspond to private-, public, and the third-kind-of school respectively.

In model 1 a diffusion cross-diffusion equation is used for the u_i while the v_{ij} spread due a diffusion equation:

$$\frac{\partial u_i}{\partial t} = c_i \Delta u_i - c_{i1} \Delta v_{i1} - c_{i2} \Delta v_{i2} - c_{i3} \Delta v_{i3} \tag{1}$$

$$\frac{\partial v_{ij}}{\partial t} = \tilde{c}_{ij} \Delta v_{ij} + g\left(u_1, u_2, u_3\right). \tag{2}$$

In model 2 the attracting fields are described by a Laplace-equation

$$-\Delta v_{ij} = g\left(u_1, u_2, u_3\right). \tag{3}$$

2 Numerical Approximation

2.1 explicit scheme using multiple threads

The equations in both models are discretized using a finite-difference approach. **Using a finite Volume approach results in the same method, but is more natural when using this type of equations. It also allows grid refinement and non-quadratic grid-cells; I will try to explain it to you in the upcoming days. It uses a little bit more advanced mathematics but results in (almost) the same numerical method.**

Using

$$\begin{aligned}\Delta v_{ij}(t,x,y) \approx & \frac{v_{ij}(t,x+\Delta x,y) - 2v_{ij}(t,x,y) + v_{ij}(t,x-\Delta x,y)}{(\Delta x)^2} \\ &+ \frac{v_{ij}(t,x,y+\Delta y) - 2v_{ij}(t,x,y) + v_{ij}(t,x,y-\Delta y)}{(\Delta y)^2}\end{aligned}$$

in equation (2) yields

$$\begin{aligned}\frac{\partial v_{ij}}{\partial t}(t,x,y) \approx & \frac{v_{ij}(t,x+\Delta x,y) - 2v_{ij}(t,x,y) + v_{ij}(t,x-\Delta x,y)}{(\Delta x)^2} \\ &+ \frac{v_{ij}(t,x,y+\Delta y) - 2v_{ij}(t,x,y) + v_{ij}(t,x,y-\Delta y)}{(\Delta y)^2}.\end{aligned}$$

Using an explicit euler method in time brings us to the scheme

$$\begin{aligned}\frac{u_i(t+\Delta t,x,y) - u_i(t,x,y)}{\Delta t} \approx & \frac{v_{ij}(t,x+\Delta x,y) - 2v_{ij}(t,x,y) + v_{ij}(t,x-\Delta x,y)}{(\Delta x)^2} \\ &+ \frac{v_{ij}(t,x,y+\Delta y) - 2v_{ij}(t,x,y) + v_{ij}(t,x,y-\Delta y)}{(\Delta y)^2}.\end{aligned}$$

3 Stefan's remarks

Here are a few remarks Stefan gave me today:

1. He suggested to use a grid refinement where the estimated error is high.

2. Using multiple threads we should control the synchronization between the threads and compare the results using different synchronization-rates, e.g. synchronize after every time-step, after every n time steps, don't synchronize.