

Project Chemical Species Transport

Sam Nees, Paul Mildenberger, Siegfried Schöfer
February 3rd 2014
Chair for System Simulation



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Outline

Discretization of the concentration convection-diffusion equation

Fundamental equation

Boundary conditions

Discretization

Timestep Control

Implementation

Results

Discretization of the concentration convection-diffusion equation



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Fundamental equation I

$$\frac{\partial c}{\partial t} + \vec{u} \cdot \nabla c = \lambda \Delta c + Q(t, x, y, \vec{u}, c) \quad (1)$$

- λ diffusion coefficient
- Q source term (location, time, velocity, concentration), multi-species-systems are modeled here
- the chemical processes under consideration have no effect on the density and hence produce no buoyancy forces \Rightarrow no coupling with the momentum equations
- convective transport: $\vec{u} \cdot \nabla c$
- diffusive spreading: $\lambda \Delta c$

Fundamental equation II

- time step control necessary
- for very stiff equations, implicit procedures might be necessary, we do an explicit time stepping

Boundary conditions

- Dirichlet BC when injecting concentration
- Neumann BC for impermeable walls

Discretization I

$$\left[\frac{\partial c}{\partial t} \right]_{ij}^{n+1} + \left[\frac{\partial uc}{\partial x} \right]_{ij} + \left[\frac{\partial vc}{\partial y} \right]_{ij} = \lambda \left[\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right]_{ij} + Q_{ij} \quad (2)$$

$$\left[\frac{\partial c}{\partial t} \right]_{ij}^{n+1} = \frac{1}{dt} (c_{ij}^{n+1} - c_{ij}^n) \quad (3)$$

$$\left[\frac{\partial uc}{\partial x} \right]_{ij} = \frac{1}{dx} \left(u_{ij} \frac{c_{ij} + c_{i+1,j}}{2} - u_{i-1,j} \frac{c_{i-1,j} + c_{ij}}{2} \right) + \frac{\gamma}{dx} \left(|u_{ij}| \frac{c_{ij} - c_{i+1,j}}{2} - |u_{i-1,j}| \frac{c_{i-1,j} - c_{ij}}{2} \right) \quad (4)$$

Discretization II

$$\left[\frac{\partial v c}{\partial y} \right]_{ij} = \frac{1}{dy} \left(v_{ij} \frac{c_{ij} + c_{i,j+1}}{2} - v_{i,j-1} \frac{c_{i,j-1} + c_{ij}}{2} \right) + \frac{\gamma}{dy} \left(|v_{ij}| \frac{c_{ij} - c_{i,j+1}}{2} - |v_{i,j-1}| \frac{c_{i,j-1} - c_{ij}}{2} \right) \quad (5)$$

$$\left[\frac{\partial^2 c}{\partial x^2} \right]_{ij} = \frac{c_{i+1,j} - 2c_{ij} + c_{i-1,j}}{(dx)^2} \quad (6)$$

$$\left[\frac{\partial^2 c}{\partial y^2} \right]_{ij} = \frac{c_{i,j+1} - 2c_{ij} + c_{i,j-1}}{(dy)^2} \quad (7)$$

- concentration c lies on the center of the staggered grid cell
- standard central differences of the Laplacian
- add timestep control

Timestep Control

$$dt < \frac{1}{2\lambda_i} \left(\frac{1}{(dx)^2} + \frac{1}{(dy)^2} \right)^{-1} \quad (8)$$

- has to be added for every new species
- depends on the diffusion constant

Implementation I

- array of Arrays for multiple concentrations
- array of reals for the lambdas
- extended parameter reader
- extended initialization

Implementation II

Main loop:

- `determineNextDT`: added support for the concentration convection diffusion equation
- `refreshBoundaries`: Dirichlet and Neumann BC according to input for every species (e.g. iterate over all Arrays)
- `computeFG`: no change
- `composeRHS`: no change
- `calculateConcentrations`: explicit concentration update
- `updateVelocities`: no change
- `vtk`: support for multiple concentrations

Results I

- two species come together at an obstacle and create a new one
- one species is coming from west and east
- the other from the south

Results II

