Using antidiffusion to solve the advection equation more accurately

Case study: Smolarkiewicz' iterative approach

Gerhard Burger Jelmer Wolterink

Scientific Computing
Department of Mathematics
Utrecht University

Project presentations SOAC, 31 October 2011

Universiteit Utrecht

Outline

1. The advection equation in climate modelling
   - Importance
   - Diffusion
   - Antidiffusion methods
2. Case study: Smolarkiewicz
   - the scheme and method of lines
   - implementation
   - numerical results

Advection

Transport mechanism of a **substance** by a *fluid*, due to the fluids motion in a particular direction.

Examples in ocean, atmosphere and climate modelling

- Transport of **trace gasses** by *air* due to wind
- Transport of **heat** by *ocean water* due to currents
- Transport of **warm and moist air** over a colder surface by *air* due to wind: advection fog

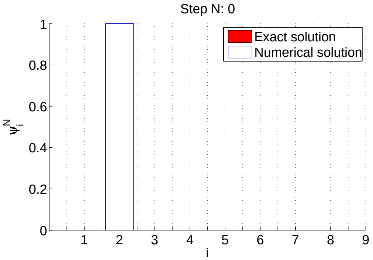
Advection simulation

Constraints on advection simulation [?]

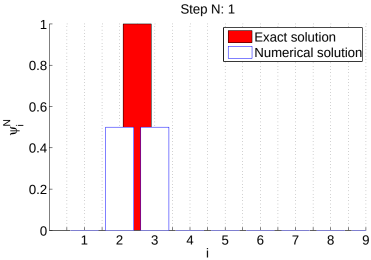
- Solutions should contain no unphysical overshoot or undershoot: positive definite
- Methods should be volume preserving. No loss of matter
- The solutions should be local: the solution at any one point should not be influenced by what is going on far away from that point
- The numerical solution should not introduce new extrema in the solution, because the continuous form of the solution would not
- The method should be cost effective: memory and computational requirements should be sufficiently small so that practical problems may be solved

When simulating an advective process, we need to discretize space. When doing so, we introduce numerical constraints on the calculation and we get diffusion.

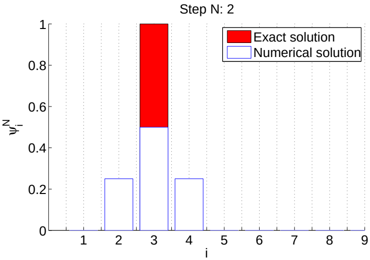
Example of diffusion with an upstream scheme



Example of diffusion with an upstream scheme

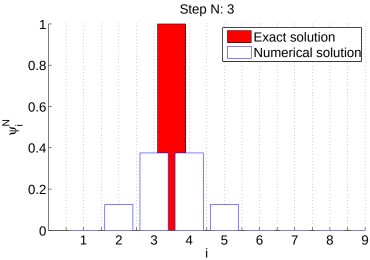


Example of diffusion with an upstream scheme

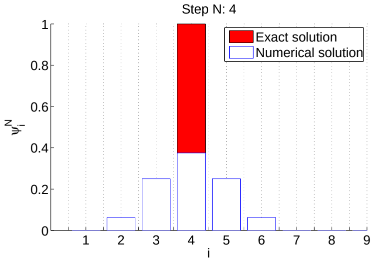


Universiteit Utrecht

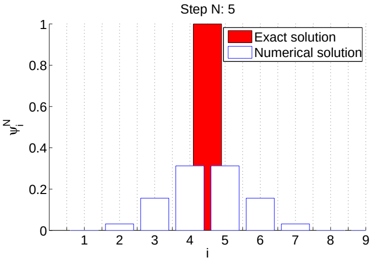
Example of diffusion with an upstream scheme



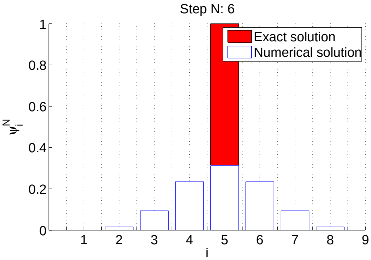
Example of diffusion with an upstream scheme



Example of diffusion with an upstream scheme

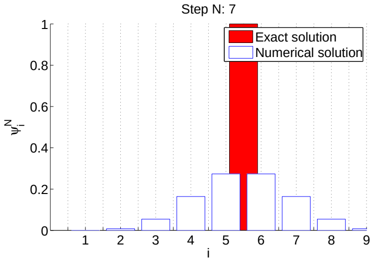


Example of diffusion with an upstream scheme



Universiteit Utrecht

Example of diffusion with an upstream scheme



Antidiffusion methods 1

- Flux-corrected transport (FCT) method (Boris and Book, 1973),
- self-adjusting hybrid scheme (SAHS) (Harten en Zwas, 1972),

both can be very accurate but require excessive computing time, better is the

- hybrid-type scheme based on a Crowley advection scheme (Clark and Hall, 1979),

with more diffusion but half the computation time.

Antidiffusion methods 2

- Smolarkiewicz' iterative correction

less time consuming while results are comparable to those of the more complex hybrid schemes

The scheme

We start with the following upstream advection equation on staggered grid:

$$\psi_i^{N+1} = \psi_i^N - \left(F\left(\psi_i^N, \psi_{i+1}^N, u_{i+1/2}^N\right) - F\left(\psi_{i-1}^N, \psi_i^N, u_{i-1/2}^N\right) \right),$$

where

$$F\left(\psi_i^N, \psi_{i+1}^N, u_{i+1/2}^N\right) = \\ \left(\left(u_{i+1/2}^N + u_{i+1/2}^N\right) \psi_i^N + \left(u_{i+1/2}^N - u_{i+1/2}^N\right) \psi_{i+1}^N \right) \frac{\Delta t}{2\Delta x}.$$

Writing it out Inserting this and collecting terms gives us

$$
\begin{aligned}
\psi_i^{N+1} = {} & \frac{\Delta t}{2\Delta x}\left(u^N_{i-1/2} + u^N_{i-1/2}\right)\psi^N_{i-1} \\
& + \left(1 - \frac{\Delta t}{2\Delta x}\left(u^N_{i+1/2} + u^N_{i+1/2} - u^N_{i-1/2} + u^N_{i-1/2}\right)\right)\psi^N_i \\
& - \frac{\Delta t}{2\Delta x}\left(u^N_{i+1/2} - u^N_{i+1/2}\right)\psi^N_{i+1}
\end{aligned}
$$

Writing it out We can write this as

$$\psi_i^{N+1} = \alpha_i \psi_{i-1}^N + \beta_i \psi_i^N + \gamma_i \psi_{i+1}^N, \quad \text{for } i = 1, \dots, M-1,$$

where we have that

$$\begin{aligned}
\alpha_i &= \frac{\Delta t}{2\Delta x}\left(u_{i-1/2}^N + u_{i-1/2}^N\right), \\
\beta_i &= \left(1 - \frac{\Delta t}{2\Delta x}\left(u_{i+1/2}^N + u_{i+1/2}^N - u_{i-1/2}^N + u_{i-1/2}^N\right)\right), \\
\gamma_i &= -\frac{\Delta t}{2\Delta x}\left(u_{i+1/2}^N - u_{i+1/2}^N\right).
\end{aligned}$$

Matrix form We can also write this in matrix form using periodic boundary conditions

$$\begin{bmatrix} \psi_1^{N+1} \\ \psi_2^{N+1} \\ \vdots \\ \psi_{M-2}^{N+1} \\ \psi_{M-1}^{N+1} \end{bmatrix} = \begin{bmatrix} \beta_1 & \gamma_1 & & & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha_{M-2} & \beta_{M-2} & \gamma_{M-2} \\ 0 & & & \alpha_{M-1} & \beta_{M-1} \end{bmatrix} \begin{bmatrix} \psi_1^{N} \\ \psi_2^{N} \\ \vdots \\ \psi_{M-2}^{N} \\ \psi_{M-1}^{N} \end{bmatrix}$$

Matrix form We can also write this in matrix form using periodic boundary conditions

$$\begin{bmatrix} \psi_1^{N+1} \\ \psi_2^{N+1} \\ \vdots \\ \psi_{M-2}^{N+1} \\ \psi_{M-1}^{N+1} \end{bmatrix} = \begin{bmatrix} \beta_1 & \gamma_1 & & & \alpha_1 \\ \alpha_2 & \beta_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha_{M-2} & \beta_{M-2} & \gamma_{M-2} \\ \gamma_{M-1} & & & \alpha_{M-1} & \beta_{M-1} \end{bmatrix} \begin{bmatrix} \psi_1^{N} \\ \psi_2^{N} \\ \vdots \\ \psi_{M-2}^{N} \\ \psi_{M-1}^{N} \end{bmatrix}$$