

**University of Vienna
Faculty of Physics**

Bachelor Thesis

Modelling Atmospheric Flows In Cartesian And
Spherical Geometry

Author:	Peter Wirnsberger
Matriculation number:	0805471
Supervisor:	Dr. Nikos Nikiforakis Prof. Christoph Dellago
Begin:	July 2010
End:	May 2011

Contents

Contents	2
1 Introduction	4
1.1 Conservation Law And Advection Equation	5
1.2 Accuracy And Truncation Error Analysis	5
1.3 The Riemann Problem	6
1.4 Conservative Methods	7
2 First- And Second-Order Numerical Methods	10
2.1 Godunov’s First-Order Upwind Scheme	10
2.2 Lax-Friedrichs Scheme	11
2.3 First-Order Centered Scheme (FORCE)	12
2.4 Lax-Wendroff Scheme	13
2.5 Warming-Beam Scheme	14
2.6 Flux Limiter Centered Scheme (FLIC)	14
2.7 Slope Limiter Centered Scheme (SLIC)	15
2.8 Weighted Average Flux Scheme (WAF)	17
2.9 Comparison For Scalar Advection	19
2.10 Discussion	26
3 Multidimensional Advection In Cartesian Geometry	27
3.1 Dimensional Splitting Method	27
3.2 Two-dimensional and three-dimensional advection	28
3.2.1 Discussion	29
3.3 Variable Coefficient Extension	33
3.3.1 Kinematic Cyclogenesis	35
3.3.2 Discussion	37
4 Atmospheric Model With Adaptive Mesh Refinement	38
4.1 Overview	38
4.2 Governing Equations In Spherical Geometry	38
4.3 Adaptive Mesh Refinement (AMR)	39
4.3.1 Nesting	40

<i>CONTENTS</i>	3
4.3.2 Integration	40
4.3.3 Regridding	42
4.4 Cubed Sphere Method	43
4.5 Test problems for the latitude-longitude and cubed sphere representation .	46
4.5.1 Solid Body Rotation	46
4.5.2 Kinematic Cyclogenesis	46
4.6 Discussion	54
List of Figures	56
List of Tables	59
Bibliography	60

Chapter 1

Introduction

This thesis focusses on numerical schemes for the solution of the advection equation. The advection equation is of special interest, because it serves as a model equation for hyperbolic conservation laws. Understanding its solutions and suitable numerical schemes allows, for example, a simple model of a concentration bulb being propagated around the sphere by the wind. A more interesting set of equations are the so called time-dependent Euler equations. It is a set of non-linear hyperbolic conservation laws that govern the dynamics of a fluid neglecting the effects of body forces, viscous stresses and heat fluxes. Furthermore, the Euler equations can be written as a system of conservation laws. Therefore, all the following numerical schemes would also apply for this set of equations, although not all are equally well suited.

In the first two chapters the governing equations and different-order schemes are introduced and their features discussed using continuous and discontinuous initial conditions. Since an analytical solution is available for the advection equation, the deviation and properties of each scheme can be studied. Afterwards the advection equation is solved for two and three dimensions with the dimensional splitting method. Finally, at the end of chapter 3, a variable coefficient extension is added to solve the kinematic cyclogenesis test problem.

Chapter 4 discusses the solid body advection and the kinematic cyclogenesis in spherical geometry using a technique called Adaptive Mesh Refinement (AMR). Time step restrictions on the regular latitude-longitude grid led to the investigation of alternative mapping techniques one of which is the "cubed sphere". After presenting its key properties, the test cases are applied to this geometry and eventually compared to the solutions for the latitude-longitude grid for various resolutions. With regard to the programming, for chapter 4 I used the CAMR framework and added the test problems, whereas I did the programming for chapters 1 to 3 on my own.

1.1 Conservation Law And Advection Equation

Following [1], conservation laws are systems of partial differential equations of the form

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{0} \quad (1.1)$$

where \mathbf{U} is the vector of conserved variables and $\mathbf{F}(\mathbf{U})$ the vector of fluxes. Since

$$\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{x}} = \mathbf{A}(\mathbf{U}) \frac{\partial \mathbf{U}}{\partial \mathbf{x}} \quad (1.2)$$

where $\mathbf{A}(\mathbf{U})$ is the Jacobi-Matrix, (1.1) can always be written in a quasi-linear form. For m conserved variables this system is said to be *hyperbolic* at a point (x, t) , if \mathbf{A} has m real eigenvalues and a set of m linearly independent eigenvectors.

For a flux $f(u) = a \cdot u$, where a is a constant, one obtains

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (1.3)$$

which is the one dimensional *linear advection equation*. The hyperbolic PDE given by (1.3) describes the transport of a quantity u by a velocity a . The solution for a domain $-\infty < x < \infty$, $t \geq 0$ and initial condition

$$u(x, 0) = u_0(x) \quad (1.4)$$

is

$$u(x, t) = u_0(x - at) \quad (1.5)$$

which means, that the initial data simply propagates with velocity a . The shape of the initial profile remains unchanged and moves to the right if $a > 0$ and to the left if $a < 0$. This can be seen, because $u = u(x(t), t)$ and therefore

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{dx}{dt} \frac{\partial u}{\partial x} \quad (1.6)$$

The curve $x = x(t)$ in the t - x -plane satisfying $dx/dt = a$ is called *characteristic curve* and comparison with (1.3) shows, that u remains constant along this curve. The speed a is called the *characteristic speed*. The characteristic curve passing through the point x_0 is $x(t) = x_0 + at$ and recalling that u remains constant along the characteristic curves finally leads us to (1.5). It is important, that the solution at a point $P(x^*, t^*)$ only depends on the initial data at a point (x_0, t_0) which is connected to P by the characteristic curve.

1.2 Accuracy And Truncation Error Analysis

One can use Taylor's theorem to approximate the partial derivatives in space and time, but the numerical approximation cannot take into account all terms of the series. This leads to an error in the numerical approximation which is called **truncation error** $\tau(\Delta t, \Delta x)$.

A numerical scheme is said to be *p-th-order accurate in time and q-th-order accurate in space*, if for sufficiently smooth exact solutions [13]

$$\tau(\Delta t, \Delta x) = \mathcal{O}(\Delta t^p, \Delta x^q). \quad (1.7)$$

A numerical scheme is said to be *convergent* if

$$\lim_{\Delta t, \Delta x \rightarrow 0} \max_{i,n} |u(x_i, t^n) - u_i^n| = 0 \quad (1.8)$$

where $u(x, t)$ represents the exact solution [13].

Truncation error analysis reveals that, for first-order schemes, the error which arises due to neglecting terms in the approximation is equivalent to solving an equation of the form

$$u_t + au_x = \alpha u_{xx} \quad (1.9)$$

where α is a numerical viscosity coefficient. For second-order schemes the *modified equation* is of the form [3]

$$u_t + au_x = \beta u_{xxx}. \quad (1.10)$$

In order to measure how far the numerical solution drifts from the exact solutions, numerical approximations to the L_p -norms are introduced. Let $u_i^{\text{exact}} = u(x_i, t^n)$ be the exact solution, then two possible norms are defined as

$$L_1(u) = \Delta x \sum_i |u_i - u_i^{\text{exact}}| \quad (1.11)$$

$$L_2(u) = \Delta x \sum_i |u_i - u_i^{\text{exact}}|^2. \quad (1.12)$$

If the norm is plotted against the space resolution Δx in a log-log-plot, a straight line with a slope smaller or equal to the order of the scheme can be expected.

1.3 The Riemann Problem

In the first section the solutions of the linear advection equation have been studied. Now a partial differential equation (PDE) with initial conditions (IC), also called initial value problem (IVP), of the form

$$\left. \begin{array}{l} \text{PDE:} \quad u_t + au_x = 0 \\ \text{IC:} \quad u(x, 0) = u_0(x) = \begin{cases} u_L & \text{if } x < 0 \\ u_R & \text{if } x > 0 \end{cases} \end{array} \right\} \quad (1.13)$$

is considered. The solution is piecewise constant depending on whether the point of interest is to the left or to the right of the discontinuity. As has been explained before, u remains constant along a characteristic curve which is why the discontinuity at $x = 0$ propagates a distance $d = at$ in time t . Since the initial profile does not change its shape, for all points to the left or to the right of the discontinuity the solution remains u_L or u_R respectively. Thus, the solution of the Riemann problem (1.1) is [1]

$$u(x, t) = u_0(x - at) = \begin{cases} u_L & \text{if } x - at < 0 \\ u_R & \text{if } x - at > 0. \end{cases} \quad (1.14)$$

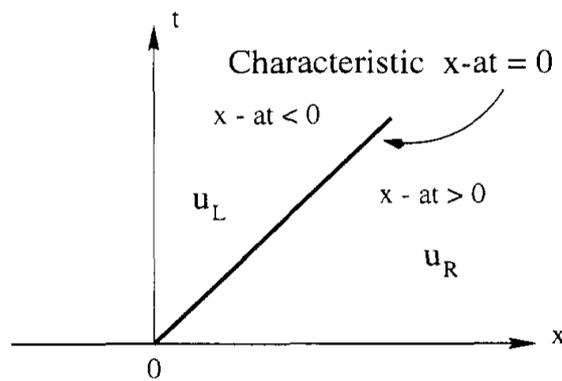


Figure 1.1: Solution of the Riemann problem in the x - t -plane [1].

1.4 Conservative Methods

There are different ways of formulating the equations describing the physics behind a certain phenomenon, but it is desirable to retain the physical and mathematical properties as accurately as possible. Methods based on non-conservative formulations other than (1.1) fail in some aspects when it comes to discontinuities, such as shock-waves [1]. The Burgers' equation is an example of an equation for which the solution can develop discontinuities. It can be written in both, conservative and non-conservative form

$$\begin{aligned} u_t + \left(\frac{1}{2}u^2\right)_x &= 0 \\ u_t + uu_x &= 0, \end{aligned}$$

but only the former is capable of handling discontinuous solutions [3]. For this reason, methods based on conservative formulations are studied. Recall the scalar conservation law

$$u_t + f(u)_x = 0 \quad (1.15)$$

which is written in differential form. In order to include weak solutions, the integral form of (1.15) can be used, which can either be formulated as

$$\oint (u dx - f dt) = 0 \quad (1.16)$$

using Green's theorem or

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t_{n+1}) dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t_n) dx + \int_{t_n}^{t_{n+1}} f(u(x_{i-\frac{1}{2}}, t)) dt - \int_{t_n}^{t_{n+1}} f(u(x_{i+\frac{1}{2}}, t)) dt \quad (1.17)$$

for any rectangular control volume $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [t_n, t_{n+1}]$ [1]. With this definition we no longer have point values u_i^n , but integral averages over a finite volume which directly leads us to the *conservative update formula*

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{\Delta x} (f_{i-\frac{1}{2}} - f_{i+\frac{1}{2}}) \quad (1.18)$$

with $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, $\Delta t = t_{n+1} - t_n$, u_i^n and $f_{i\pm\frac{1}{2}}$ being the following integral averages

$$u_i^n = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t_n) dx \quad (1.19)$$

$$f_{i\pm\frac{1}{2}} = \frac{1}{\Delta t} \int_0^{\Delta t} f(u(x_{i\pm\frac{1}{2}}, t)) dt \quad (1.20)$$

as shown in Fig. 1.2 and 1.3.

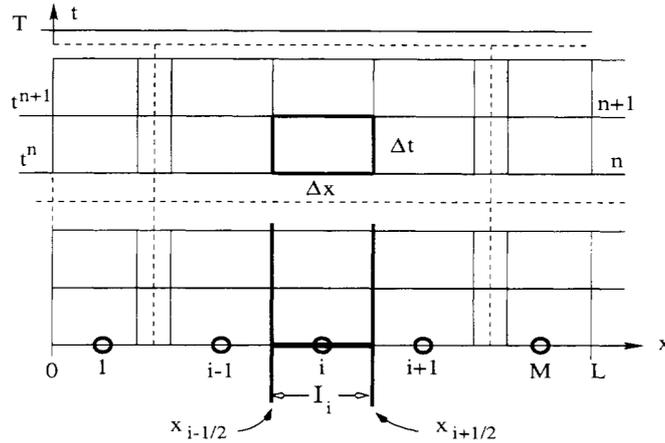


Figure 1.2: Control volume in the x - t -plane [1].

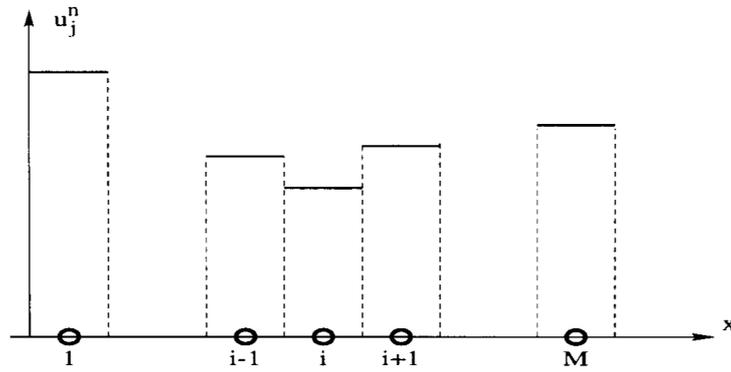


Figure 1.3: Piecewise constant distribution of the data [1].

A **Conservative Method** for the scalar conservation law (1.15) is a numerical method of the form (1.18) where

$$f_{i+\frac{1}{2}} = f_{i+\frac{1}{2}}(u_{i-l_L}^n, \dots, u_{i+l_R}^n) \quad (1.21)$$

with l_L, l_R two non-negative integers; $f_{i+\frac{1}{2}}$ is called the numerical flux, an approximation to the physical flux $f(u)$ in (1.15) [1].

Chapter 2

First- And Second-Order Numerical Methods

Now after having discussed the scalar conservation law and the basic idea of conservative methods, the focus is on various first- and second-order numerical methods. Some of those can be derived directly by approximating (1.15) with finite differences, but the main interest lies in the finite volume representation with the conservative update formula (1.18). At the end of this chapter simulation results are compared for different initial profiles.

2.1 Godunov's First-Order Upwind Scheme

The approximation of the time and space derivatives of (1.3) by first-order one-sided finite differences yields

$$u_i^{n+1} = u_i^n - c(u_i^n - u_{i-1}^n) \quad (2.1)$$

for a positive velocity a , where

$$c = a \frac{\Delta t}{\Delta x} \quad (2.2)$$

is called the *Courant-Friedrichs-Lewy (CFL) number*. This scheme is called **First Order Upwind scheme** or **CIR scheme** and it is first-order accurate in space and time. Furthermore, it remains conditionally stable for $0 \leq c \leq 1$ [1]. The viscosity coefficient for the modified equation is [3]

$$\alpha_{\text{upwind}} = \frac{1}{2} \Delta x a (1 - |c|). \quad (2.3)$$

Godunov was able to extend the CIR scheme to a conservative form that uses the conservative update formula (1.18). The intercell fluxes needed for this formula can be computed by using the solutions of local Riemann problems. Thus, (1.15) must be solved at each intercell boundary for a piecewise constant distribution at the time level n to evolve the

solution in time [1]. As explained before, the exact solution of the local Riemann problem $\text{RP}(u_{i-1}^n, u_i^n)$ is given by

$$u_{i-\frac{1}{2}}(x/t) = \begin{cases} u_{i-1}^n & \text{if } x/t < a, \\ u_i^n & \text{if } x/t > a \end{cases} \quad (2.4)$$

for a velocity $a > 0$, where $(0,0)$ is the local origin of the Riemann problem. With the solutions of the two Riemann problems $\text{RP}(u_{i-1}^n, u_i^n)$ and $\text{RP}(u_i^n, u_{i+1}^n)$ Godunov defines the updated solution u_i^{n+1} as the integral average

$$u_i^{n+1} = \frac{1}{\Delta x} \left(\int_0^{\frac{1}{2}\Delta x} u_{i-\frac{1}{2}}(x/\Delta t) dx + \int_{-\frac{1}{2}\Delta x}^0 u_{i+\frac{1}{2}}(x/\Delta t) dx \right). \quad (2.5)$$

In this notation Δt is the local time in the Riemann problems' own frame of reference. The integrals can be evaluated using (2.4) and the whole expression will reduce to (2.1) [1]. However, Godunov also presented a second formulation with appropriate expressions for the intercell fluxes. The integral average (2.5) can be rewritten as

$$u_i^{n+1} = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \tilde{u}(x, \Delta t) dx \quad (2.6)$$

where $\tilde{u}(x, t)$ is the combined solution of the two Riemann problems [1]. Application of the integral form (1.17) to (2.6) for a control volume $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [0, \Delta t]$ leads to the averages $f_{i\pm\frac{1}{2}}$ of the physical flux $f(u)$ in (1.19). For each cell the exact solutions at the interfaces (along the t-axis) are given by

$$\tilde{u}(x_{i-\frac{1}{2}}, t) = u_{i-\frac{1}{2}}(0), \quad \tilde{u}(x_{i+\frac{1}{2}}, t) = u_{i+\frac{1}{2}}(0) \quad (2.7)$$

which finally yields

$$f_{i-\frac{1}{2}} = f(u_{i-\frac{1}{2}}(0)), \quad f_{i+\frac{1}{2}} = f(u_{i+\frac{1}{2}}(0)) \quad (2.8)$$

for the fluxes [1]. For the choice of $f(u) = au$ with positive a , we get

$$f_{i-\frac{1}{2}} = au_{i-1}^n, \quad f_{i+\frac{1}{2}} = au_{i+1}^n \quad (2.9)$$

for the intercell fluxes and if applied to (1.18) this method reproduces the CIR scheme. Again, it is stable for $0 \leq c \leq 1$ and first-order accurate in both space and time [1].

2.2 Lax-Friedrichs Scheme

The Lax-Friedrichs scheme can easily be obtained by substituting u_i^n in the approximation of the time derivative

$$u_t = \frac{u_i^{n+1} - u_i^n}{\Delta t} \quad (2.10)$$

by the mean value of the two neighbouring cells

$$u_i^n = \frac{1}{2}(u_{i-1}^n + u_{i+1}^n). \quad (2.11)$$

Applied to the linear advection equation this results in

$$u_i^{n+1} = \frac{1+c}{2}u_{i-1}^n + \frac{1-c}{2}u_{i+1}^n. \quad (2.12)$$

Equation (2.12) is called the **Lax-Friedrichs scheme** and it is first-order accurate in time and second-order accurate in space. Furthermore, the numerical viscosity coefficient for this scheme is

$$\alpha_{LF} = \frac{\Delta x a}{2c}(1-c^2) \quad (2.13)$$

and it is conditionally stable for $0 \leq c \leq 1$ [1].

For the intercell flux the choice of

$$f_{i+\frac{1}{2}}^{LF} = \frac{1+c}{2c}f(u_i^n) + \frac{(c-1)}{2c}f(u_{i+1}^n) \quad (2.14)$$

respectively

$$f_{i+\frac{1}{2}}^{LF} = \frac{1}{2}(f_i^n + f_{i+1}^n) + \frac{1}{2} \frac{\Delta x}{\Delta t} (u_i^n - u_{i+1}^n) \quad (2.15)$$

(if $f(u) = au$) together with the conservative update formula will reproduce the Lax-Friedrichs scheme as defined by (2.12) [1].

2.3 First-Order Centered Scheme (FORCE)

In order to derive this scheme, the solution is first advanced to the intermediate states

$$u_{i\pm\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{\Delta x} \int_{-\frac{1}{2}\Delta x}^{+\frac{1}{2}\Delta x} \tilde{u}_{i\pm\frac{1}{2}}(x, \frac{1}{2}\Delta t) dx \quad (2.16)$$

where $\tilde{u}_{i\pm\frac{1}{2}}(x, t)$ is again the combined solution of the two Riemann problems $\text{RP}(u_{i-1}^n, u_i^n)$ and $\text{RP}(u_i^n, u_{i+1}^n)$ in their local frame of reference. Applying (1.18) yields

$$u_{i-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}(u_{i-1}^n + u_i^n) + \frac{\Delta t}{2\Delta x}(f_{i-1}^n - f_i^n) \quad (2.17)$$

$$u_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}(u_i^n + u_{i+1}^n) + \frac{\Delta t}{2\Delta x}(f_i^n - f_{i+1}^n). \quad (2.18)$$

Now the solution can be advanced in time again to the complete time step Δt , but this time for the Riemann problem $\text{RP}(u_{i-\frac{1}{2}}^{n+\frac{1}{2}}, u_{i+\frac{1}{2}}^{n+\frac{1}{2}})$, which results in an integral average

$$u_i^{n+1} = \frac{1}{\Delta x} \int_{-\frac{1}{2}\Delta x}^{+\frac{1}{2}\Delta x} \tilde{u}_i(x, \frac{1}{2}\Delta t) dx \quad (2.19)$$

where $\tilde{u}_i(x, t)$ is the solution of $\text{RP}(u_{i-\frac{1}{2}}^{n+\frac{1}{2}}, u_{i+\frac{1}{2}}^{n+\frac{1}{2}})$. Application of (1.18) for a second time gives

$$u_i^{n+1} = \frac{1}{2} \left(u_{i-\frac{1}{2}}^{n+\frac{1}{2}} + u_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right) + \frac{\Delta t}{2\Delta x} \left(f_{i-\frac{1}{2}}^{n+\frac{1}{2}} - f_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (2.20)$$

where the intercell flux

$$f_{i+\frac{1}{2}}^{n+\frac{1}{2}} = f(u_{i+\frac{1}{2}}^{n+\frac{1}{2}}) \equiv f_{i+\frac{1}{2}}^{RI} \quad (2.21)$$

is called the Richtmyer flux. Finally, the flux for the FORCE scheme is defined as the arithmetic average of the fluxes for the Richtmyer and Lax-Friedrichs schemes [1]

$$f_{i+\frac{1}{2}}^{FORCE} \equiv \frac{1}{2} \left(f_{i+\frac{1}{2}}^{RI} + f_{i+\frac{1}{2}}^{LF} \right). \quad (2.22)$$

The **FORCE scheme** is first-order accurate, conditionally stable for $0 \leq |c| \leq 1$ and its viscosity coefficient for the modified equation [1] is

$$\alpha_{fo} = \frac{1}{4} \Delta x a \frac{1-c^2}{c} = \frac{1}{2} \alpha_{lf}. \quad (2.23)$$

2.4 Lax-Wendroff Scheme

If one substitutes the space derivative u_x by an average of the upwind and downwind approximations, which are

$$u_x = \frac{u_i^n - u_{i-1}^n}{\Delta x} \quad (2.24)$$

$$u_x = \frac{u_{i+1}^n - u_i^n}{\Delta x}, \quad (2.25)$$

one obtains

$$u_x = \beta_1 \frac{u_i^n - u_{i-1}^n}{\Delta x} + \beta_2 \frac{u_{i+1}^n - u_i^n}{\Delta x}. \quad (2.26)$$

The choice of

$$\beta_1 = \frac{1}{2}(1+c), \quad \beta_2 = \frac{1}{2}(1-c) \quad (2.27)$$

finally results in the **Lax-Wendroff scheme**

$$u_i^{n+1} = \frac{1}{2}c(1+c)u_{i-1}^n + (1-c^2)u_i^n - \frac{1}{2}c(1-c)u_{i+1}^n. \quad (2.28)$$

This scheme is second-order accurate in both space and time and remains conditionally stable for $0 \leq c \leq 1$. As applied to the linear advection equation, the Lax-Wendroff scheme can also be written in conservative form with an intercell flux [1]

$$f_{i+\frac{1}{2}}^{LW} = \frac{1+c}{2}(au_i^n) + \frac{(1-c)}{2}(au_{i+1}^n). \quad (2.29)$$

2.5 Warming-Beam Scheme

The Warming-Beam scheme is another second-order accurate scheme in space and time. For positive a it is defined as

$$u_i^{n+1} = \frac{1}{2}c(c-1)u_{i-2}^n + c(2-c)u_{i-1}^n + \frac{1}{2}(c-1)(c-2)u_i^n. \quad (2.30)$$

Contrarily to the Lax-Wendroff scheme for example, this scheme is fully one-sided because all mesh points involved are on the left-hand side of the centre of the stencil. The stability restriction for this scheme is $0 \leq c \leq 2$ and hence less stringent than those of the others. Therefore, higher time steps are possible increasing the efficiency. The intercell flux

$$f_{i+\frac{1}{2}}^{WB} = \frac{1}{2}(c-1)f_{i-1} + \frac{1}{2}(3-c)f_i \quad (2.31)$$

applied to the conservative formula reproduces the **Warming-Beam scheme** [1].

2.6 Flux Limiter Centered Scheme (FLIC)

High-order linear (constant coefficient) schemes cause oscillations near discontinuities or large gradients [1], which can be seen at the end of this chapter. On one hand, reduction to a first-order scheme in the vicinity of large gradients would solve this problem. But on the other hand, at least second-order accuracy on smooth solutions is desirable. This contradiction can be solved by introducing an intercell flux of the form [1]

$$f_{i+\frac{1}{2}} = f_{i+\frac{1}{2}}^{LO} + \Phi_{i+\frac{1}{2}} \left[f_{i+\frac{1}{2}}^{HI} - f_{i+\frac{1}{2}}^{LO} \right], \quad (2.32)$$

where $f_{i+\frac{1}{2}}^{LO}$ is a low-order flux and $f_{i+\frac{1}{2}}^{HI}$ a high-order flux. The function $\Phi_{i+\frac{1}{2}}$ is called *flux limiter function* and it can take any value between 0 and 1. For $\Phi_{i+\frac{1}{2}} = 0$ the constructed flux reduces to the low-order flux and for $\Phi_{i+\frac{1}{2}} = 1$ to the high-order flux. There is no unique way of defining the flux limiter function and the proportion of high- and low-order fluxes [4].

One possible choice for $\Phi_{i+\frac{1}{2}}$ is the so called *superbee flux limiter*, which is defined as

$$\Phi_{i+\frac{1}{2}}(r) = \begin{cases} 0 & \text{if } r \leq 0 \\ 2r & \text{if } 0 \leq r \leq \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} \leq r \leq 1 \\ \min(2, \Phi_g + (1 - \Phi_g)r) & \text{if } r > 1 \end{cases} \quad (2.33)$$

with $\Phi_g \equiv (1 - c)/(1 + c)$ for example, c the CFL-coefficient and

$$r = \begin{cases} \frac{u_i^n - u_{i-1}^n}{u_{i+1}^n - u_i^n} & \text{if } a > 0 \\ \frac{u_{i+1}^n - u_i^n}{u_i^n - u_{i-1}^n} & \text{if } a < 0. \end{cases} \quad (2.34)$$

With the above definition of $\Phi_{i+\frac{1}{2}}$ and (2.32) the **FLIC scheme** is finally constructed by setting [1]

$$f_{i+\frac{1}{2}}^{LO} = f_{i+\frac{1}{2}}^{FORCE}, \quad f_{i+\frac{1}{2}}^{HI} = f_{i+\frac{1}{2}}^{RI}. \quad (2.35)$$

2.7 Slope Limiter Centered Scheme (SLIC)

Following [1], the motivation of this approach is to reach higher order of accuracy by reconstruction of the data. In (1.19) u_i^n was defined as the integral average over the control volume. This led to a piecewise constant distribution of the data to be advanced in time. One possible high-order extension is achieved by linearly interpolating the data with a function

$$u_i(x) \equiv u_i^n + \frac{(x - x_i)}{\Delta x} \Delta_i \quad (2.36)$$

where $x \in [0, \Delta x]$ and x_i denotes the centre of the cell. In this case the slope is $\frac{\Delta_i}{\Delta x}$, but here it is continued calling Δ_i the slope of the local linear interpolation. One possible choice is

$$\Delta_i = \frac{1}{2}(1 + \omega)\Delta u_{i-\frac{1}{2}} + \frac{1}{2}(1 - \omega)\Delta u_{i+\frac{1}{2}} \quad (2.37)$$

where

$$\Delta u_{i-\frac{1}{2}} \equiv u_i^n - u_i^{n-1}, \quad \Delta u_{i+\frac{1}{2}} \equiv u_{i+1}^n - u_i^n \quad (2.38)$$

and $\omega \in [-1, 1]$ is a free parameter. In this local coordinates $x = \frac{1}{2}\Delta x$ corresponds to the centre x_i of the cell and thus $u_i(x_i) = u_i^n$. Of importance are also the so called *boundary extrapolated values*

$$u_i^L = u_i(0) = u_i^n - \frac{1}{2}\Delta_i, \quad u_i^R = u_i(\Delta x) = u_i^n + \frac{1}{2}\Delta_i. \quad (2.39)$$

The reconstruction process is illustrated by Fig. 2.1.

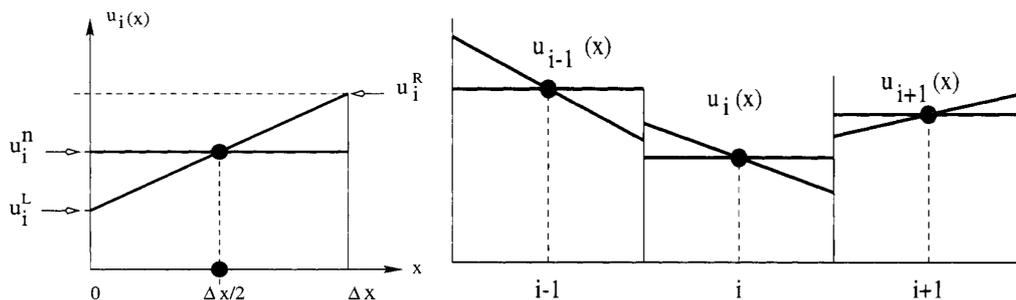


Figure 2.1: Reconstructed locally linearly interpolated data [1].

It is proven and advantageous to further adapt the slope by introducing a slope-limiter function ξ_i .

$$\bar{\Delta}_i = \xi_i \Delta_i. \quad (2.40)$$

Similarly to the flux-limiter function previously introduced, this function takes an argument r defined by (2.34) as a measure of the flow. One possible choice is the *superbee slope-limiter function*

$$\xi_{sb}(r) = \begin{cases} 0 & \text{if } r \leq 0 \\ 2r & \text{if } 0 \leq r \leq \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} \leq r \leq 1 \\ \min(r, \xi_R(r), 2) & \text{if } r > 1 \end{cases} \quad (2.41)$$

where $\xi_R(r) \equiv 4[(1-c)(1-\omega+(1+\omega)r)]^{-1}$ and c the CFL number [1]. The **SLIC scheme** can now be created in three steps. First, u_i^L and u_i^R are calculated with the updated slope.

$$u_i^L = u_i^n - \frac{1}{2}\bar{\Delta}_i, \quad u_i^R = u_i^n + \frac{1}{2}\bar{\Delta}_i \quad (2.42)$$

Second, u_i^L and u_i^R are evolved by a time $\frac{1}{2}\Delta t$ according to

$$\bar{u}_i^L = u_i^L + \frac{1}{2} \frac{\Delta t}{\Delta x} [f(u_i^L) - f(u_i^R)] \quad (2.43)$$

$$\bar{u}_i^R = u_i^R + \frac{1}{2} \frac{\Delta t}{\Delta x} [f(u_i^L) - f(u_i^R)]. \quad (2.44)$$

Third, the conservative update formula (1.18) with the intercell flux

$$f_{i+\frac{1}{2}}^{SLIC} = f_{i+\frac{1}{2}}^{FORCE}(\bar{u}_i^L, \bar{u}_i^R) \quad (2.45)$$

of the FORCE scheme (2.22) is used. The so constructed scheme is second-order accurate in space and time and stable if the CFL number satisfies $|c| \leq 1$ for any $\omega \in [-1, 1]$ [1].

2.8 Weighted Average Flux Scheme (WAF)

The last scheme to be discussed is a second-order Riemann problem based method. A Riemann problem based scheme (Godunov's first order upwind scheme) has already been studied, but it is only first-order accurate. Alternative ways of increasing the accuracy have also been considered, like using slope-limiter and flux-limiter functions for the SLIC and FLIC schemes respectively.

However, the general formula for a WAF-type flux for a cell $[x_1, x_2] \times [t_1, t_2]$ in the x - t -plane is

$$f_{i+\frac{1}{2}}^{WAF} = \frac{1}{t_2 - t_1} \frac{1}{x_2 - x_1} \int_{t_1}^{t_2} \int_{x_1}^{x_2} f(\tilde{u}_{i+\frac{1}{2}}(x, t)) dx dt \quad (2.46)$$

where $\tilde{u}_{i+\frac{1}{2}}(x, t)$ is the solution of a particular IVP. In order to obtain the original WAF-flux, the integration limits $t_1 = 0$, $t_2 = \Delta t$, $x_1 = -\frac{1}{2}\Delta x$ and $x_2 = \frac{1}{2}\Delta x$ are chosen first. Second, $\tilde{u}_{i+\frac{1}{2}}(x, t)$ is defined to be the solution of the Riemann problem $RP(u_i^n, u_i^{n+1})$. Third, approximation of the time integration of (2.46) by the midpoint rule yields

$$f_{i+\frac{1}{2}}^{WAF} = \frac{1}{\Delta x} \int_{x_1}^{x_2} f(\tilde{u}_{i+\frac{1}{2}}(x, \frac{1}{2}\Delta t)) dx \quad (2.47)$$

provided Δt is such that $|c| \leq 1$. Under the above considerations the original formula (2.46) reduces to (2.47) and only the space integration is left to be completed.

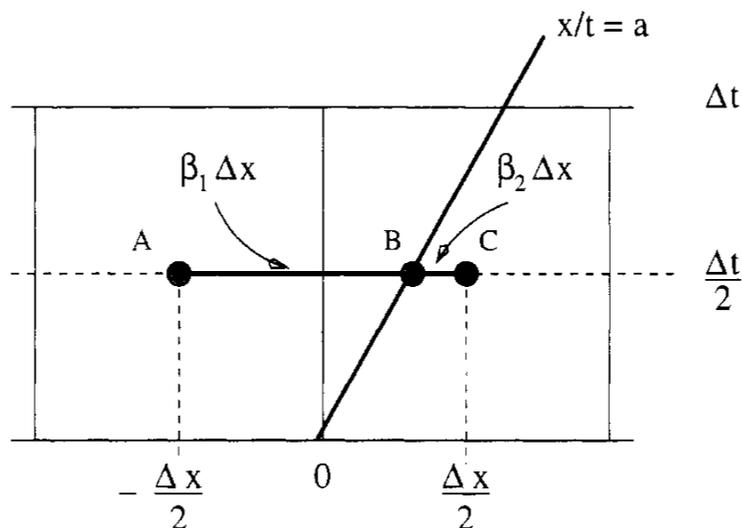


Figure 2.2: Construction of the WAF-flux [1].

Figure 2.2 illustrates the integration path in the x - t -plane consisting of the pieces AB and BC separated by the characteristic curve of the particular Riemann problem. For geometric

reasons, the coefficients are $\beta_1 = (1 + c)/2$ and $\beta_2 = (1 - c)/2$. Thus, we eventually get the flux for the **WAF-scheme** [1]

$$f_{i+\frac{1}{2}}^{WAF} = \frac{1}{\Delta x} \left[\frac{1}{2}(1 + c)\Delta x(au_i^n) \right] + \frac{1}{\Delta x} \left[\frac{1}{2}(1 - c)\Delta x(au_{i+1}^n) \right] \quad (2.48)$$

or

$$f_{i+\frac{1}{2}}^{WAF} = \frac{1}{2}(1 + c)(au_i^n) + \frac{1}{2}(1 - c)(au_{i+1}^n). \quad (2.49)$$

as applied to the linear advection equation.

For this equation the flux is identical to the Lax-Wendroff one. Instead of c weighting the upwind and downwind contributions of (2.49), one can extend the scheme by using a flux-limiter function $\Phi_{i+\frac{1}{2}}$. Then the flux becomes

$$f_{i+\frac{1}{2}}^{WAF} = \frac{1}{2}(1 + \Phi_{i+\frac{1}{2}})(au_i^n) + \frac{1}{2}(1 - \Phi_{i+\frac{1}{2}})(au_{i+1}^n) \quad (2.50)$$

where $\Phi_{i+\frac{1}{2}} = \Phi_{i+\frac{1}{2}}(|c|, r_{i+\frac{1}{2}})$ with r defined by (2.34). A possible choice for the WAF-type limiter is

$$\Phi_{i+\frac{1}{2}} = 1 - (1 - |c|)B_{sb} \quad (2.51)$$

with $B_{sb}(r)$ being the superbee limiter function

$$B_{sb}(r) = \begin{cases} 0 & \text{if } r \leq 0 \\ 2r & \text{if } 0 \leq r \leq \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} \leq r \leq 1 \\ r & \text{if } 1 \leq r \leq 2 \\ 2 & \text{if } r \geq 2. \end{cases} \quad (2.52)$$

Since the WAF-flux also contains the Godunov flux, it may be regarded as an extension to Godunov's first order upwind scheme [1].

2.9 Comparison For Scalar Advection

Having briefly discussed different approaches and the resulting schemes, the numerical solutions for continuous and discontinuous initial profiles are now studied. Three different initial conditions are advected on a periodic domain $[0, 1]$ with velocity $a = 1.0$, a CFL number $c = 0.9$ and a resolution of 100 cells. The first profile to be tested is a sine-wave representing continuous initial profiles. Subsequently, the top-hat profile serves as an example to study the behaviour on discontinuous profiles and finally both together form the third profile which is referred to as combined profile. This profile was tested twice for one and ten cycles.

Sine-wave

$$u(x, 0) = \sin(2\pi x) \quad (2.53)$$

Top-hat

$$u(x, 0) = \begin{cases} 0 & \text{if } x < 1/3, \\ 1 & \text{if } 1/3 \leq x \leq 2/3, \\ 0 & \text{if } x > 2/3 \end{cases} \quad (2.54)$$

Combined

$$u(x, 0) = \begin{cases} 0 & \text{if } x < 1/6, \\ 1 & \text{if } 1/6 \leq x \leq 1/3, \\ 0 & \text{if } 1/3 < x \leq 1/2, \\ \sin(4\pi x) & \text{if } x > 1/2 \end{cases} \quad (2.55)$$

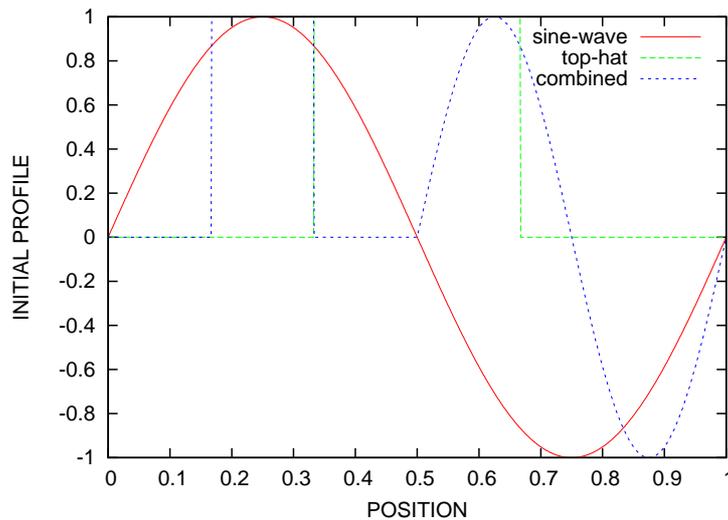


Figure 2.3: Initial profiles given by (2.53)-(2.55).

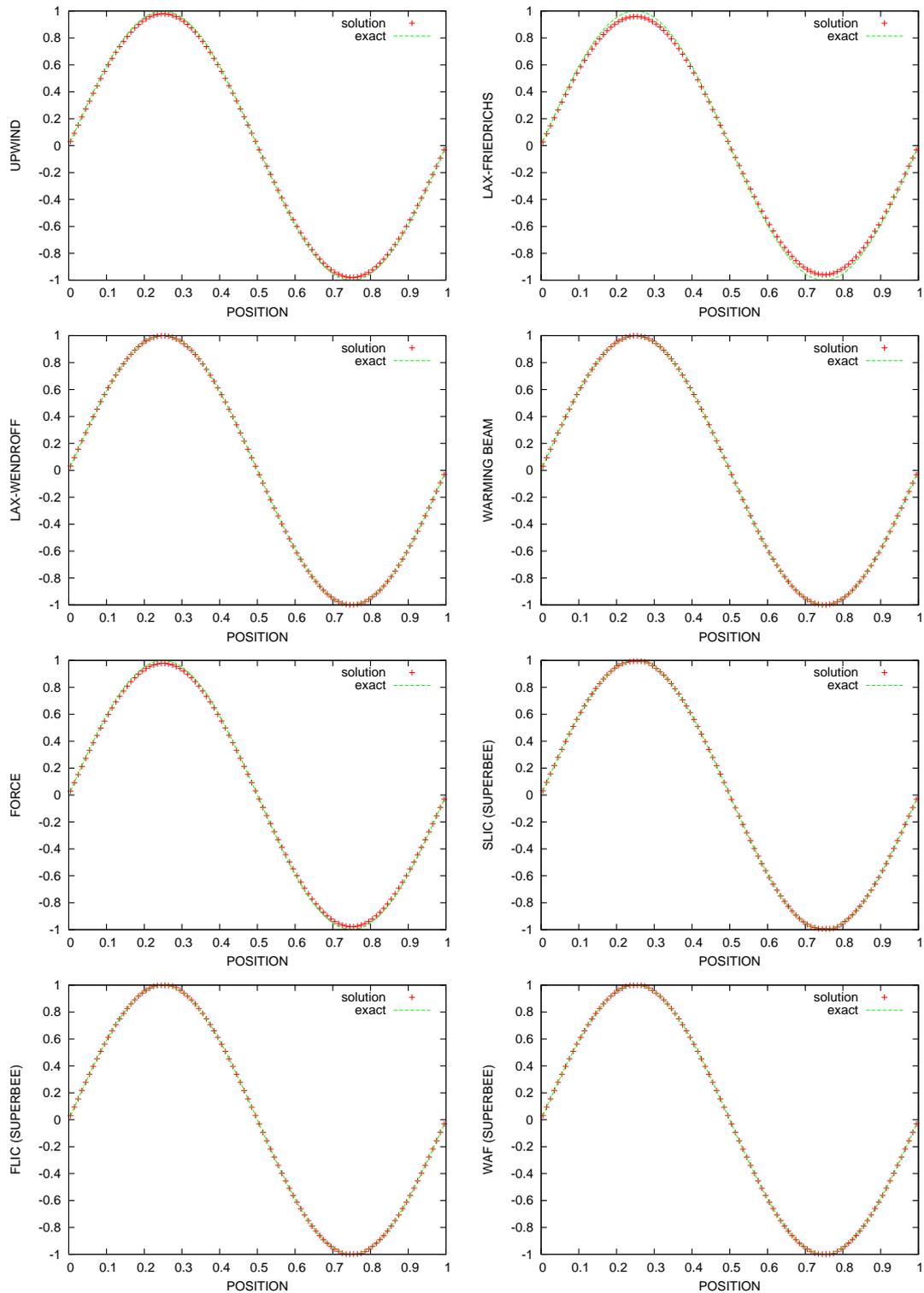


Figure 2.4: Solutions for the sine-wave profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme.

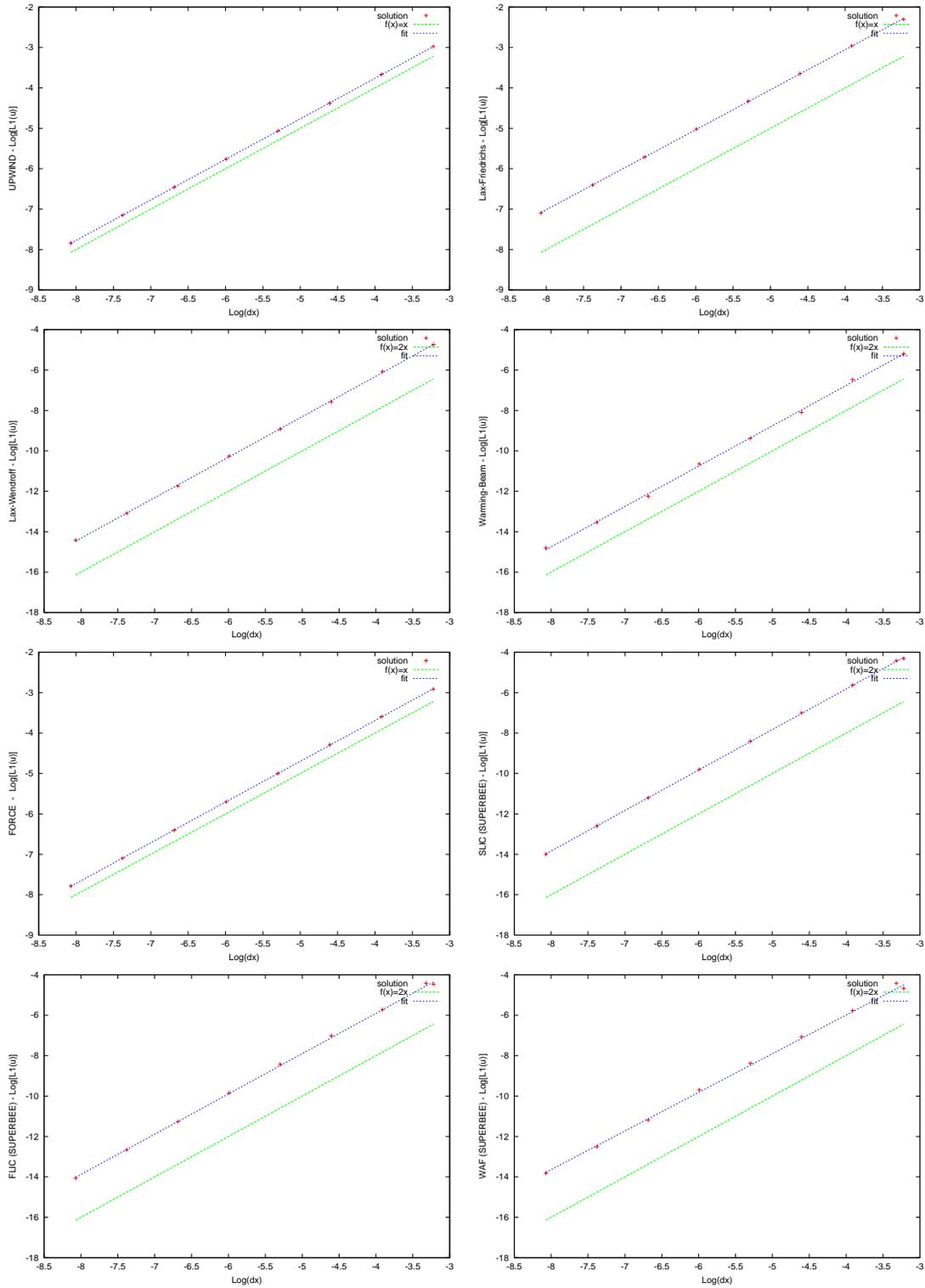


Figure 2.5: Convergence tests for the sine-wave profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and $\text{cells} = 2^i \cdot 25$ ($i = 0, 1, \dots, 8$). Each chart represents a different scheme.

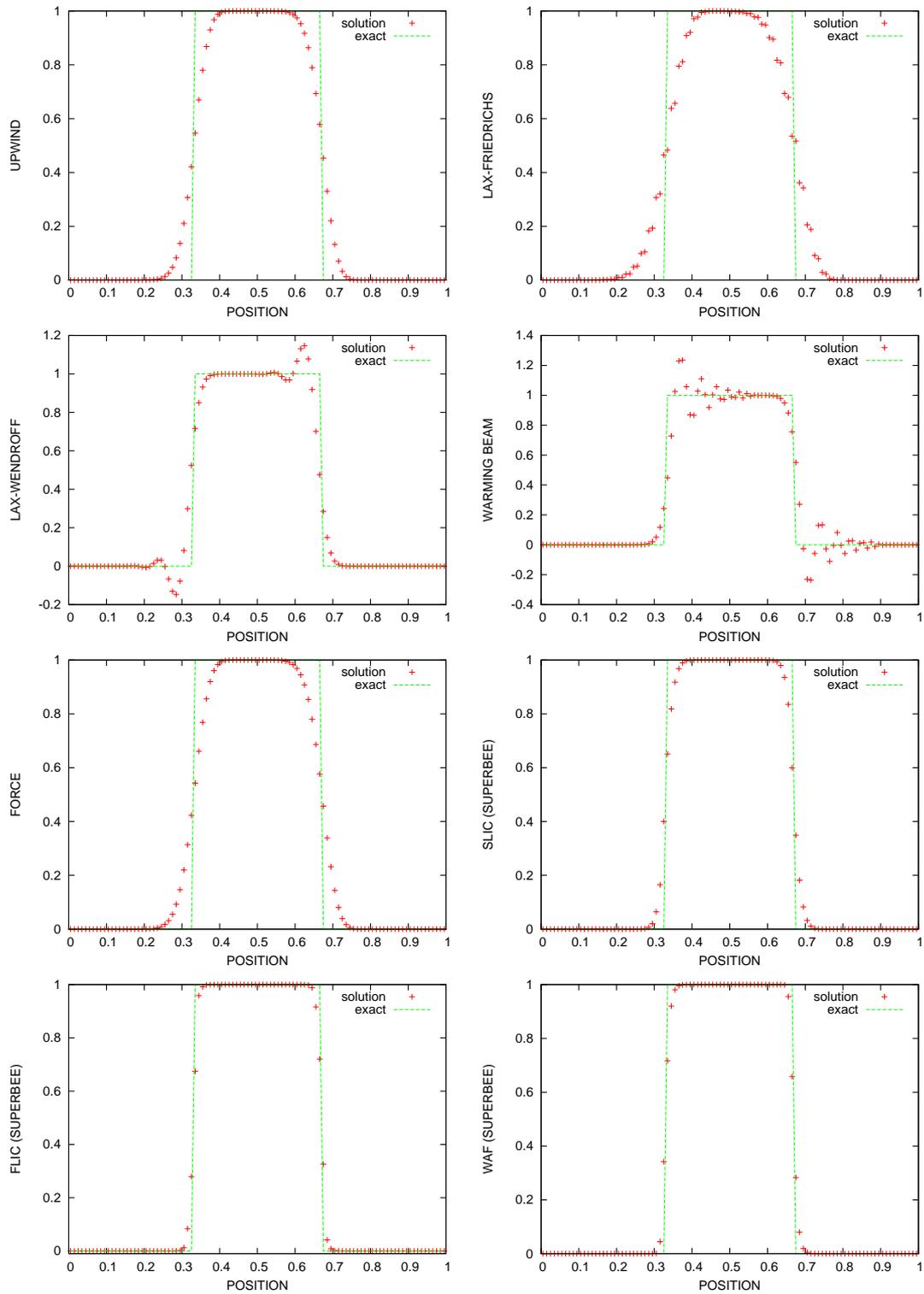


Figure 2.6: Solutions for the top-hat profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme.

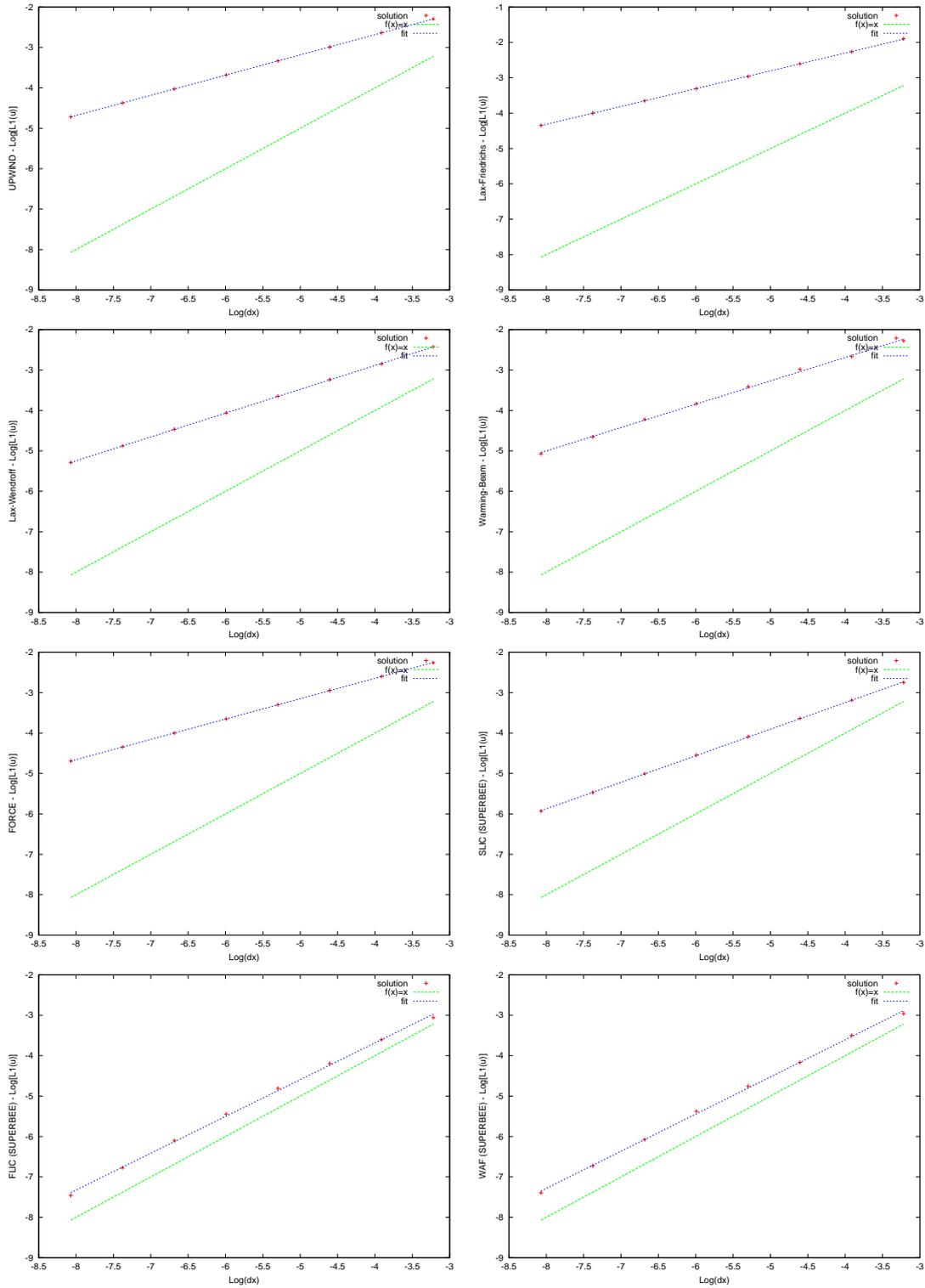


Figure 2.7: Convergence tests for top-hat profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and $\text{cells} = 2^i \cdot 25$ ($i = 0, 1, \dots, 8$). Each chart represents a different scheme.

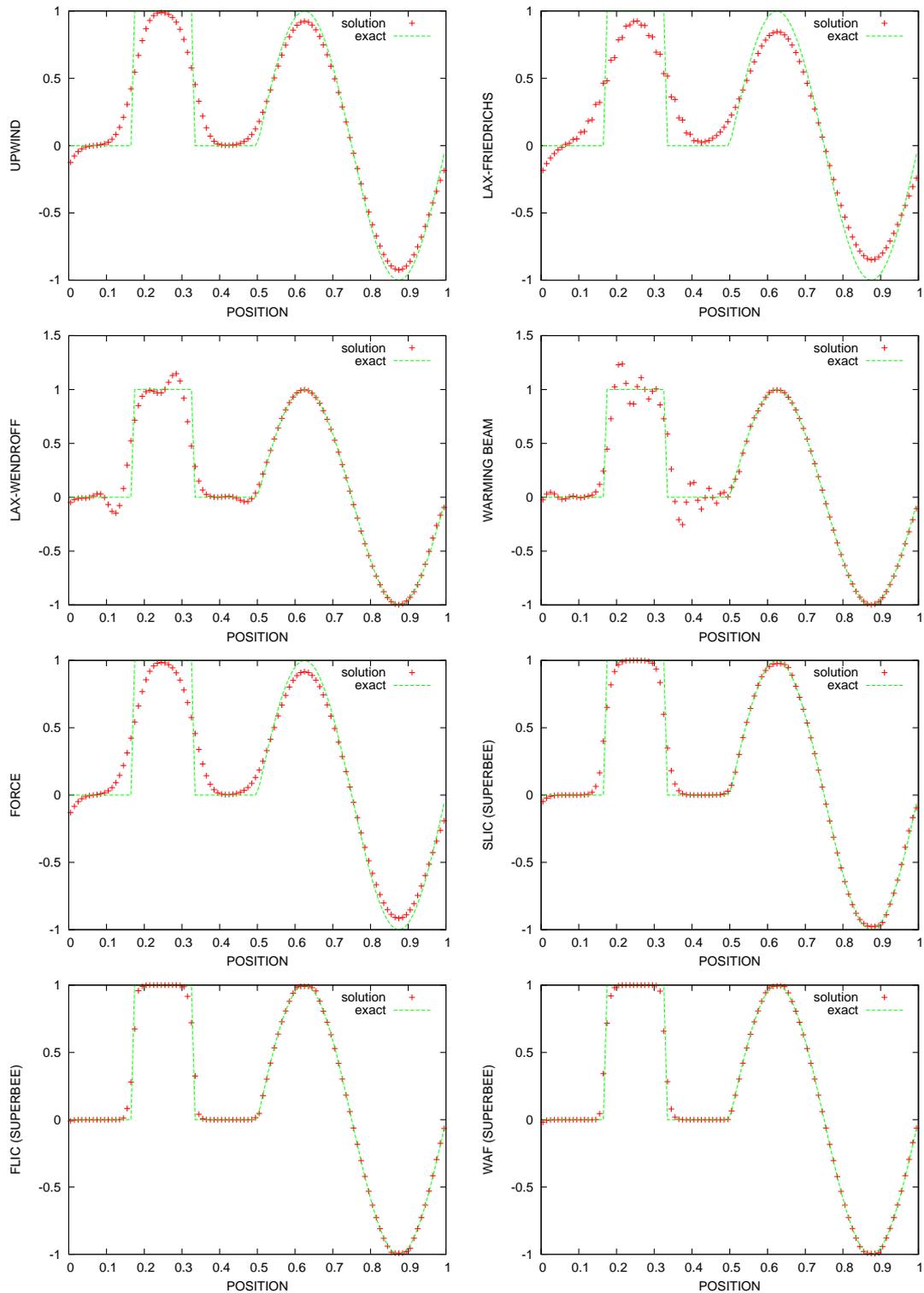


Figure 2.8: Solutions for the combined profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme.

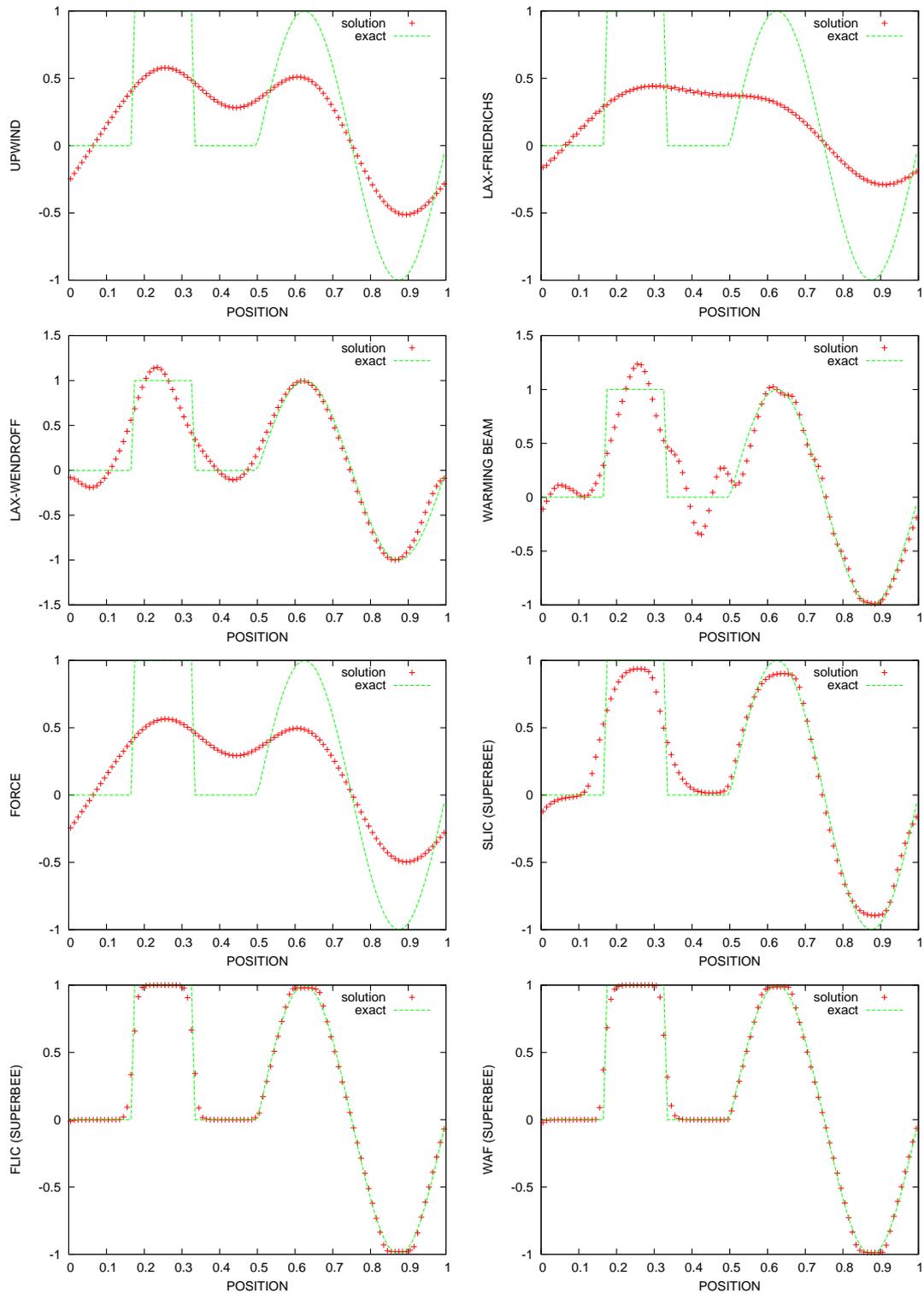


Figure 2.9: Solutions for the combined profile at $t = 10.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme.

2.10 Discussion

Figures 2.4 to 2.9 show the numerical solutions and the convergence tests for the schemes as applied to the three initial profiles. As for the sine-wave profile, the results look similar at first sight. However, one can see that the first-order methods (Lax-Friedrichs, FORCE and UPWIND) show slight *diffusion*, whereas the second-order methods approximate the exact solution quite well. For the top-hat profile, the diffusive behaviour of the first-order methods can be seen clearly. In the vicinity of discontinuities the numerical solutions are smeared over several neighbouring cells and fail to reproduce the large gradients properly. On the contrary, for some of the second-order methods, namely the Lax-Wendroff and the Warming Beam method, another property can be identified. The numerical solutions show oscillations in the vicinity of large gradients for both, which is why they are *dispersive*. The solutions obtained by the other second-order schemes (FLIC, SLIC and WAF) are considerably better and without dispersion.

The effects of diffusion and dispersion are even more evident for the combined profile. The top-hat part and the positive domain of the sine-wave differ only slightly for the first-order schemes. Furthermore, spurious oscillations are present in the numerical solutions of the Lax-Wendroff and Warming Beam schemes. However, the FLIC and WAF scheme again produce the best approximations showing no dispersion. Even after ten cycles the solutions remain quite good, whereas the other solutions drift far off the exact solution.

The results of the convergence tests are illustrated by Fig. 2.5 and 2.7. For this, the numerical solution was calculated for different resolutions starting with 25 cells gradually doubling up to 6400 cells. Subsequently, the results were interpolated linearly and plotted against the space resolution in a log-log-plot. As expected, the Lax-Friedrichs, FORCE and UPWIND schemes give first-order accuracy and the rest second-order accuracy for the sine-wave. However, for the discontinuous top-hat profile the convergence rates are significantly lower. The slopes of the linear fits which indicate the orders of accuracy are given by tab. 2.1.

Scheme	Slope (sine-wave)	Slope (top-hat)
UPWIND	1.0031 ± 0.0013	0.5001 ± 0.0006
Lax-Friedrichs	0.9897 ± 0.0028	0.5033 ± 0.0016
Lax-Wendroff	2.0041 ± 0.0103	0.5891 ± 0.0014
Warming-Beam	1.9965 ± 0.0230	0.5776 ± 0.0082
FORCE	1.0077 ± 0.0019	0.5028 ± 0.0011
SLIC (SUPERBEE)	2.0042 ± 0.0070	0.6571 ± 0.0023
FLIC (SUPERBEE)	1.9929 ± 0.0157	0.9105 ± 0.0132
WAF (SUPERBEE)	1.9125 ± 0.0246	0.9193 ± 0.0109

Table 2.1: Results for the convergence tests with the settings $c = 0.9$ and $a = 1.0$ at $t = 1.0$.

Chapter 3

Multidimensional Advection In Cartesian Geometry

In the previous chapters the scalar advection equation and different numerical approaches as well as their properties have been studied. Now a technique which allows to solve the multidimensional advection problem will be studied. Afterwards results for two- and three-dimensional discontinuous initial profiles are presented and discussed.

3.1 Dimensional Splitting Method

For modelling advection in three dimensions one has to update (1.3) to

$$u_t + \vec{a} \cdot \vec{\nabla} u = 0 \quad (3.1)$$

or

$$u_t + a_1 u_x + a_2 u_y + a_3 u_z = 0 \quad (3.2)$$

where a_1, a_2 and a_3 are the velocities in the x -, y - and z -direction. The exact solution of this equation is a wave propagating in the three-dimensional space, which is for the initial condition

$$u(\vec{r}, 0) = u_0(\vec{r}) \quad (3.3)$$

given by

$$u(\vec{r}, t) = u_0(\vec{r} - \vec{a}t) \quad (3.4)$$

where $\vec{r} = (x, y, z)^T$ is the position vector and \vec{a} the velocity.

One approach to obtain the numerical solution of (3.2) is a technique called **dimensional splitting** or **method of fractional steps**. This technique replaces the three-dimensional equation by a triple of one dimensional problems. Following [1], the exact solution of

the three-dimensional IVP (3.2) can then be obtained by solving three one-dimensional problems, each for a time Δt , that is

$$u^{n+1} = \mathcal{Z}^{\Delta t} \mathcal{Y}^{\Delta t} \mathcal{X}^{\Delta t} u^n \quad (3.5)$$

where $\mathcal{X}^{\Delta t}$, $\mathcal{Y}^{\Delta t}$ and $\mathcal{Z}^{\Delta t}$ denote the solution operators that advance the solution exactly by a time step Δt in time for the respective direction. The sequence of one-dimensional problems can be written as

$$\begin{aligned} \text{PDE: } & u_t + a_1 u_x = 0 \\ \text{IC: } & u^n \end{aligned}$$

$$\begin{aligned} \text{PDE: } & u_t + a_2 u_y = 0 \\ \text{IC: } & u^{n+\frac{1}{3}} \end{aligned}$$

$$\begin{aligned} \text{PDE: } & u_t + a_3 u_z = 0 \\ \text{IC: } & u^{n+\frac{2}{3}} \end{aligned}$$

where $u^{n+\frac{1}{3}}$, called **x sweep**, is the solution of the first equation and $u^{n+\frac{2}{3}}$, called **y sweep**, is the solution of the second equation. Both are separately advanced in time by the same time step Δt but with different initial conditions. Applying the **z sweep** finally yields the solution u^{n+1} of the full problem. However, the one dimensional problems must be solved for all strips in the x -, y - and z -direction.

In contrast to the one-dimensional case, the time step must be chosen more carefully since the CFL condition must hold for all directions. In order to guarantee this [1]

$$\Delta t = c \times \min_{i,j,k} \left[\frac{\Delta x_{i,j,k}}{(a_1)_{max}}, \frac{\Delta y_{i,j,k}}{(a_2)_{max}}, \frac{\Delta z_{i,j,k}}{(a_3)_{max}} \right] \quad (3.6)$$

is defined, where c is the CFL-coefficient and $\Delta x_{i,j,k}$, $\Delta y_{i,j,k}$ and $\Delta z_{i,j,k}$ are the edge lengths of the cell i,j,k of the three-dimensional grid. Although only constant velocities and cubed cells of equal size are considered at this stage, this formulation will be used later for the variable coefficient extension.

3.2 Two-dimensional and three-dimensional advection

The dimensional splitting approach for three dimensions given by (3.5) does not change for two dimensions, except for the z sweep. The calculation of the maximum stable time

step is analogous to (3.6). The simulations were run on a 100×100 and a $30 \times 30 \times 30$ cells grid respectively with the following initial profiles

top-hat 2D

$$u(x, y, 0) = \begin{cases} 1 & \text{if } 1/3 \leq x \leq 2/3 \text{ and } 1/3 \leq y \leq 2/3, \\ 0 & \text{else} \end{cases} \quad (3.7)$$

cube profile

$$u(x, y, z, 0) = \begin{cases} 1 & \text{if } 1/3 \leq x \leq 2/3 \text{ and } 1/3 \leq y \leq 2/3 \text{ and } 1/3 \leq z \leq 2/3, \\ 0 & \text{else.} \end{cases} \quad (3.8)$$

In order to perform the convergence tests of the above examples as well, the definition of $L_1(u)$ has to be modified. Instead of multiplying the deviation of the numerical solution from the exact solution by the space resolution Δx , it is now multiplied by $\Delta x \Delta y$ respectively by $\Delta v = \Delta x \Delta y \Delta z$ which yields

$$L_1(u) = \Delta x \Delta y \sum_{i,j} |u_{i,j} - u_{i,j}^{\text{exact}}| \quad (3.9)$$

and

$$L_1(u) = \Delta v \sum_{i,j,k} |u_{i,j,k} - u_{i,j,k}^{\text{exact}}|. \quad (3.10)$$

Again periodic boundaries were used for the domains $[0, 1] \times [0, 1]$ and $[0, 1] \times [0, 1] \times [0, 1]$. The time step was calculated according to (3.6) and the velocities are $\vec{a} = (1, 1)^T$ and $\vec{a} = (1, 1, 1)^T$ respectively. Although any of the schemes could have been used for the calculations, the second order WAF-scheme was applied since it reproduces discontinuities very well.

3.2.1 Discussion

The first chart in Fig. 3.1 shows the initial profile with points only at zero and one. The exact solution is expected to move towards the rear right corner, to disappear splitting up into four parts and to reappear again at the front left corner. After one cycle, at $t = 1.0$, the exact solution should match the initial profile exactly. As for the numerical solution, one can observe slight deviations already at $t = 0.2$. However, even after ten cycles it yields quite good results as illustrated by the last chart.

With regard to the three-dimensional example, the situation is similar. When the parcel arrives at the corner, it splits up into eight parts which then merge again and move to the centre of the simulation box. The deviation is now illustrated by the yellow points surrounding the parcel. In order to facilitate the identification of the inner points, the resolution was decreased.

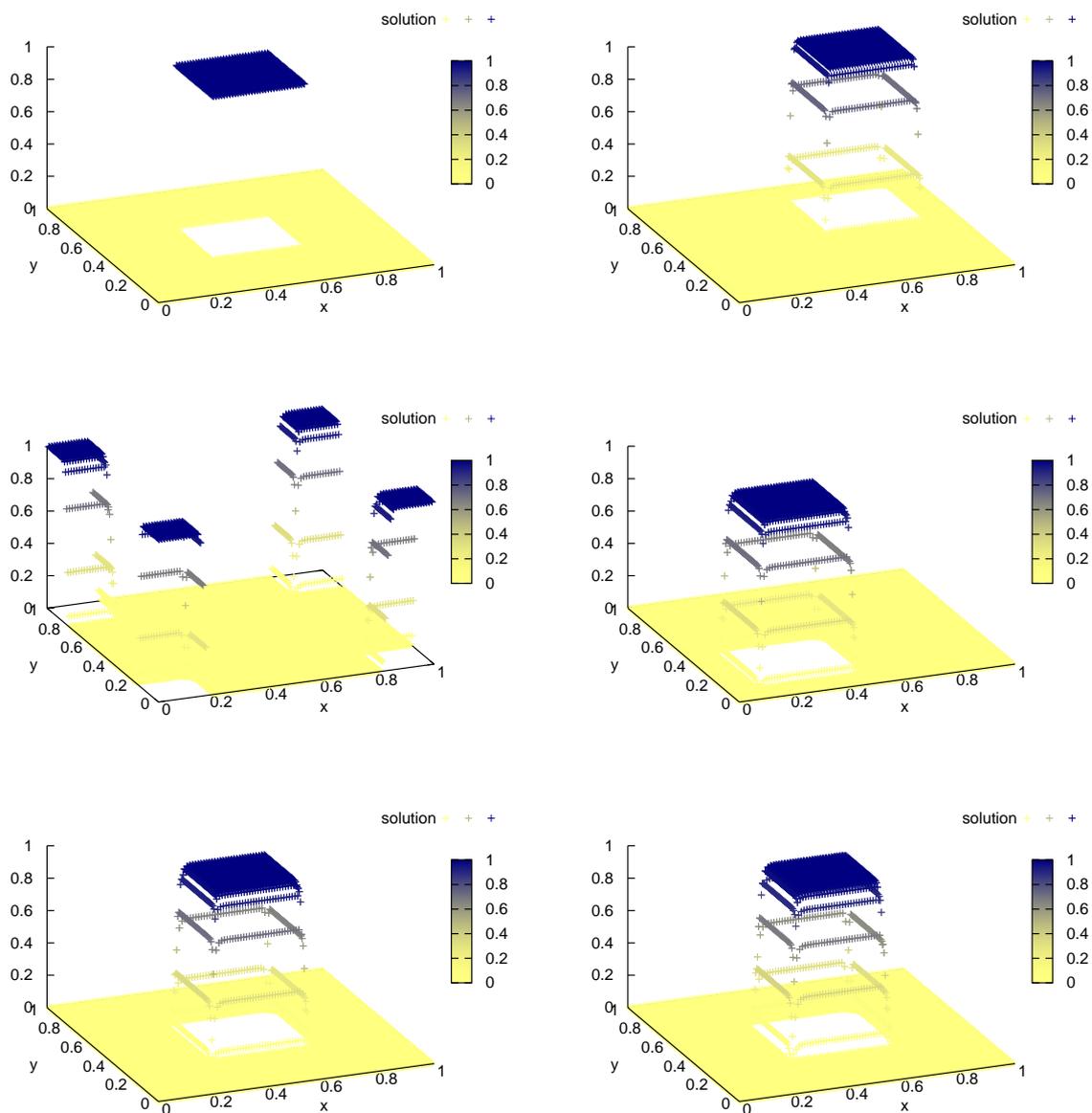


Figure 3.1: Solutions for the top-hat 2D profile with the settings 100×100 cells, $c = 0.9$, $\vec{a} = (1, 1)^T$ for 0, 0.2, 0.5, 0.8, 1 and 10 cycles.

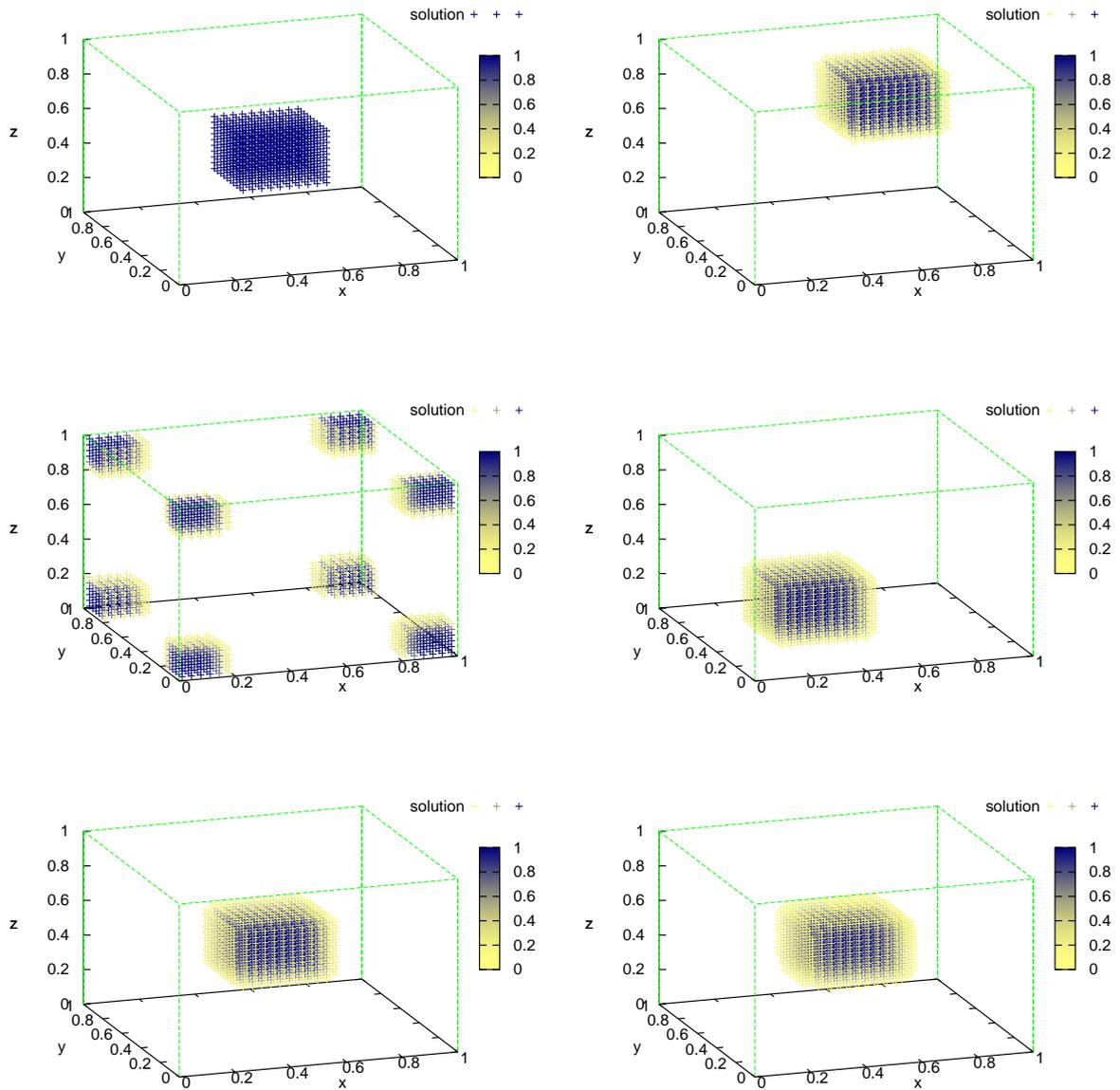
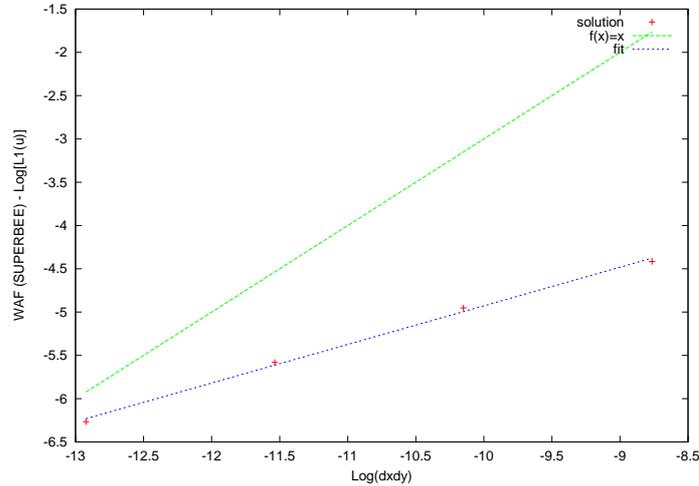
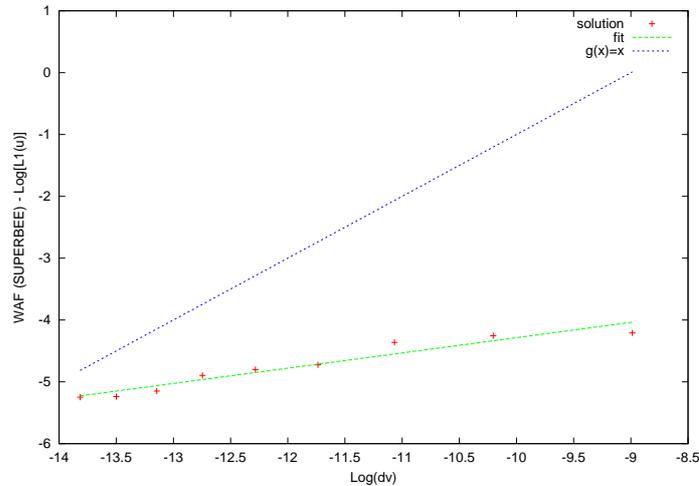


Figure 3.2: Solutions for the cube profile with the settings $30 \times 30 \times 30$ cells, $c = 0.9$, $\vec{a} = (1, 1, 1)^T$ for 0.0, 0.2, 0.5, 0.8, 1.0 and 10.0 cycles.

The charts in Fig. 3.3 indicate that the rate of convergence is reduced by each additional dimension as compared to the one-dimensional convergence test for the WAF-scheme shown in Fig. 2.7. The slopes of the linear fits are 0.446 ± 0.018 for the two-dimensional and 0.2468 ± 0.026 for the three-dimensional test problem.



(a) top-hat 2D



(b) cube profile

Figure 3.3: Convergence tests for the (a) top-hat 2D profile with the settings $c = 0.9$, $\vec{a} = (1, 1, 1)^T$, $t = 1.0$ and 80, 160, 320 and 640 cells per direction (b) cube profile with the settings $c = 0.9$, $\vec{a} = (1, 1, 1)^T$, $t = 1.0$ and 20, 30, ..., 100 cells per direction.

3.3 Variable Coefficient Extension

This section deals with advection velocity fields which vary in space and time. Following [8], we again consider the scalar advection equation

$$\Psi_t + \mathbf{u} \cdot \nabla \Psi = 0 \quad (3.11)$$

where $\Psi(\mathbf{x}, t)$ now represents the advected scalar quantity of interest and $\mathbf{u}(\mathbf{x}, t)$ the velocity field. If the velocity field is divergence free, (3.11) and the conservation law are equivalent. Otherwise it may be rewritten

$$\Psi_t + \mathbf{u} \cdot \nabla \Psi = \Psi \nabla \cdot \mathbf{u} \quad (3.12)$$

where the left-hand side is in conservation form and the right-hand side acts as a forcing term. For any nonzero divergence of the velocity field, (3.12) is inhomogeneous and the forcing term serves as a correction to the conservation law. The integral form of (3.12) is discretized by a finite volume method, because it preserves the conservation property. The entire computational domain is perfectly covered by the control volumes. Integration and application of the Gaussian theorem yield

$$\int_{\Omega} \Psi_t d\Omega + \oint_{\partial\Omega} \Psi \mathbf{u} \cdot d\mathbf{n} = \int_{\Omega} \Psi \nabla \cdot \mathbf{u} d\Omega \quad (3.13)$$

where Ω represents an arbitrary control volume with \mathbf{n} being an outward-pointing normal to its boundary $\partial\Omega$. The control volumes are defined such that they match the underlying grid cells and the solution values are stored at the cell centres. Subsequently, the definition of $\bar{\Psi}$, which is the integral average of Ψ over Ω , enables us to rewrite (3.14)

$$\left(\int_{\Omega} d\Omega \right) \bar{\Psi}_t + \oint_{\partial\Omega} \Psi \mathbf{u} \cdot d\mathbf{n} = \int_{\Omega} \Psi \nabla \cdot \mathbf{u} d\Omega \quad (3.14)$$

which can after additional application of the Gaussian theorem to the right-hand side finally be approximated by

$$\left(\int_{\Omega} d\Omega \right) \bar{\Psi}_t + \oint_{\partial\Omega} \Psi \mathbf{u} \cdot d\mathbf{n} \approx \bar{\Psi} \oint_{\partial\Omega} \mathbf{u} \cdot d\mathbf{n}. \quad (3.15)$$

Now that the equation has been discretized, the next step is to approximate the integrals. The control volume is assumed to be polyhedral and therefore (3.15) may be approximated by

$$\bar{\Psi}_t + \frac{1}{V_{\Omega}} \sum_{m=1}^{N_f} A_m (\mathbf{f}_m^* \cdot \hat{\mathbf{n}}_m)^n = \frac{1}{V_{\Omega}} \bar{\Psi}^n \sum_{m=1}^{N_f} A_m (\mathbf{u}_m^* \cdot \hat{\mathbf{n}}_m)^n \quad (3.16)$$

where N_f is the number of faces of the polyhedron, V_{Ω} is its volume, A_m are the face areas, $\hat{\mathbf{n}}_m$ their unit outward normals, and the superscripts \cdot^n and \cdot^* represent the current

time level and the fluxes. Up to this point, the equations are independent of the chosen coordinate system.

From now on the faces are assumed to be perpendicular to the coordinate system and a two-dimensional structured grid is used. The time derivative of the integral average in (3.16) is discretized by using a forward Euler approximation. The **forward Euler method** approximates the solution of the IVP

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = f_0 \end{cases} \quad (3.17)$$

within the domain $I = [t_0, T]$ with $T < \infty$. The domain is divided into N_h equidistant pieces of length $h = (T - t_0)/N_h$. The exact solution of this problem can be approximated by using simple forward differences for the time derivative

$$y^{n+1} = y^n + hf^n \quad (3.18)$$

where $y'(t^n) = f^n$. It is only first order accurate in time [13].

With this, one can finally replace $\bar{\Psi}$ by Ψ again, which represents from now on a cell-centre value of the solution, and apply the forward Euler method (3.18) to (3.16).

$$\begin{aligned} \Psi_{ij}^{n+1} = & \Psi_{ij}^n - \frac{\Delta t}{V_{ij}} \times ([A(\Psi u)^*]_{i+1/2j}^n - [A(\Psi u)^*]_{i-1/2j}^n \\ & + [A(\Psi v)^*]_{ij+1/2}^n - [A(\Psi v)^*]_{ij-1/2}^n \\ & + \Psi_{ij}^n [Au^*]_{i+1/2j}^n - \Psi_{ij}^n [Au^*]_{i-1/2j}^n \\ & + \Psi_{ij}^n [Av^*]_{ij+1/2}^n - \Psi_{ij}^n [Av^*]_{ij-1/2}^n) \end{aligned} \quad (3.19)$$

The cells of the two-dimensional grid are indexed by (i, j) , half indices denote faces between cells and $(u, v)^T$ is the velocity. Equation (3.19) can now be solved by using the dimensional splitting approach. Furthermore, the intercell flux between the cells $(i - 1, j)$ and (i, j) is denoted by $f_{i-1/2j}$. The flux calculated by using the intermediate state Ψ_{ij}^* is referred to as $f_{i-1/2j}^*$.

$$\Psi_{ij}^* = \Psi_{ij}^n + \frac{\Delta t}{\Delta x} \times (f_{i-1/2j} - f_{i+1/2j} + \Psi_{ij}^n [Au^*]_{i-1/2j}^n - \Psi_{ij}^n [Au^*]_{i+1/2j}^n) \quad (3.20)$$

$$\Psi_{ij}^{**} = \Psi_{ij}^* + \frac{\Delta t}{\Delta y} \times (f_{ij-1/2}^* - f_{ij+1/2}^* + \Psi_{ij}^* [Av^*]_{ij-1/2}^n - \Psi_{ij}^* [Av^*]_{ij+1/2}^n) \quad (3.21)$$

This scheme is second-order accurate in space and first-order accurate in time. It can now be combined with any of the previously introduced schemes. Due to its convincing performance in the previous test cases, the WAF-scheme is used for all further tests.

Reconsidering the derivation of the WAF-scheme, one notices that the velocity was constant at the cell interfaces. Now the situation has changed as the velocity is allowed to vary. Nevertheless, a more complex solution of the Riemann problem may be circumvented by using the mean velocity $\bar{a} = (a_i + a_{i+1})/2$ and the mean CFL number $\bar{c} = (c_i + c_{i+1})/2$. With this the WAF-flux can be rewritten

$$f_{i+1/2}^{WAF} = \frac{1}{2}(1 + \Phi_{i+1/2})\bar{a}\Psi_i + \frac{1}{2}(1 - \Phi_{i+1/2})\bar{a}\Psi_{i+1} \quad (3.22)$$

where $\Phi_{i+1/2} = \Phi_{i+1/2}(|\bar{c}|, r)$ is the flux-limiter function (2.52), but now with the mean CFL number as argument. The formula for the maximum stable time step is given by the two-dimensional version of (3.23)

$$\Delta t = c \times \min_{i,j} \left[\frac{\Delta x_{i,j}}{u_{max}}, \frac{\Delta y_{i,j}}{v_{max}} \right] \quad (3.23)$$

where c is the CFL-coefficient.

3.3.1 Kinematic Cyclogenesis

A frequently used test problem for scalar advection tests in Cartesian geometry is the idealized cyclogenesis problem of Doswell (1984). In contrast to the previous test problems, this one is deformational and more challenging. A more complex extension to spherical geometry is introduced in the following chapter. This one produces a vortex with (x_c, y_c) as the centre of rotation. The non-divergent velocity field is defined by zero normal velocity and

$$\begin{aligned} U_T(r) &= U_{max} \operatorname{sech}^2(r) \tanh(r) \\ r &= x^2 + y^2 \end{aligned} \quad (3.24)$$

where U_T is the tangential velocity in polar coordinates. U_{max} can be chosen to regulate the maximum magnitude of U_T . The exact solution as well as the initial condition for $t = 0$ are given by

$$\Psi(x, y, t) = -\tanh \left[\frac{y - y_c}{\delta} \cos(\omega t) - \frac{x - x_c}{\delta} \sin(\omega t) \right] \quad (3.25)$$

where δ defines the characteristic width of the front zone [5]. Decreasing δ makes the initial condition discontinuous on the grid.

The simulation was run on a two-dimensional grid with 300×300 cells up to 15 time units with a CFL number of 0.9. Figure 3.4 shows contour plots of the time evolution of the system. The vortex centre is placed at the centre of the domain. The horizontal green line in the first plot represents the frontal zone of the initial condition. Although it is chosen to be very small ($\delta = 0.01$), it is still continuous on the grid.

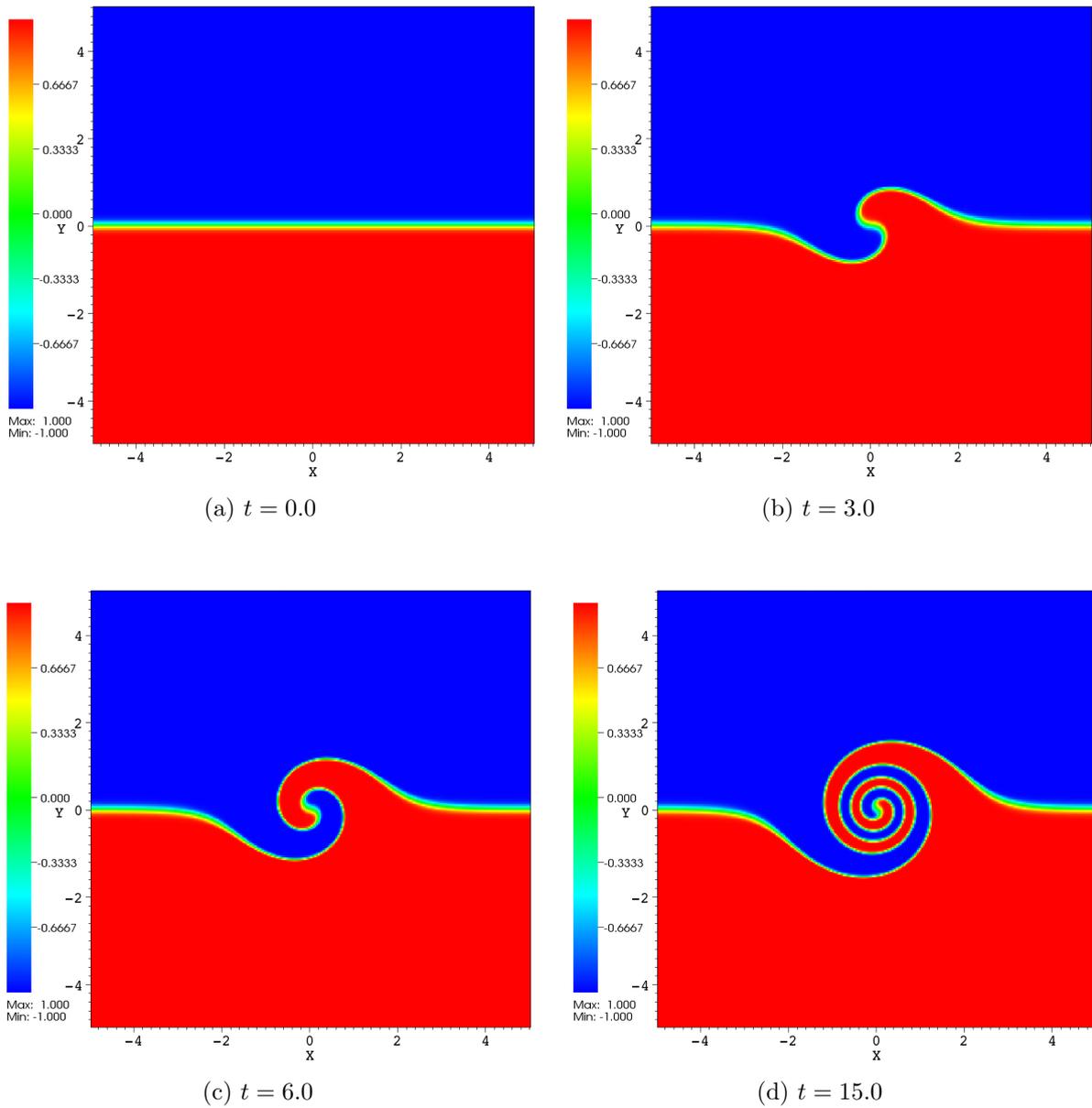


Figure 3.4: Solutions for the kinematic cyclogenesis test problem for a $[-5, 5] \times [-5, 5]$ domain with a resolution of 300×300 cells and a CFL number of 0.9 at times $t = 0, 3, 6$ and 15.

3.3.2 Discussion

The exact solution is very well approximated by the presented approach. To illustrate this, Fig. 3.5 shows a three-dimensional plot of Fig. 3.4d for both, the exact and the numerical solution at the end time. Additionally a slice was cut through the centre of both, to show the behaviour in the centre of the vortex. The blue line in 3.5c represents the exact solution and the green symbols the numerical solution.

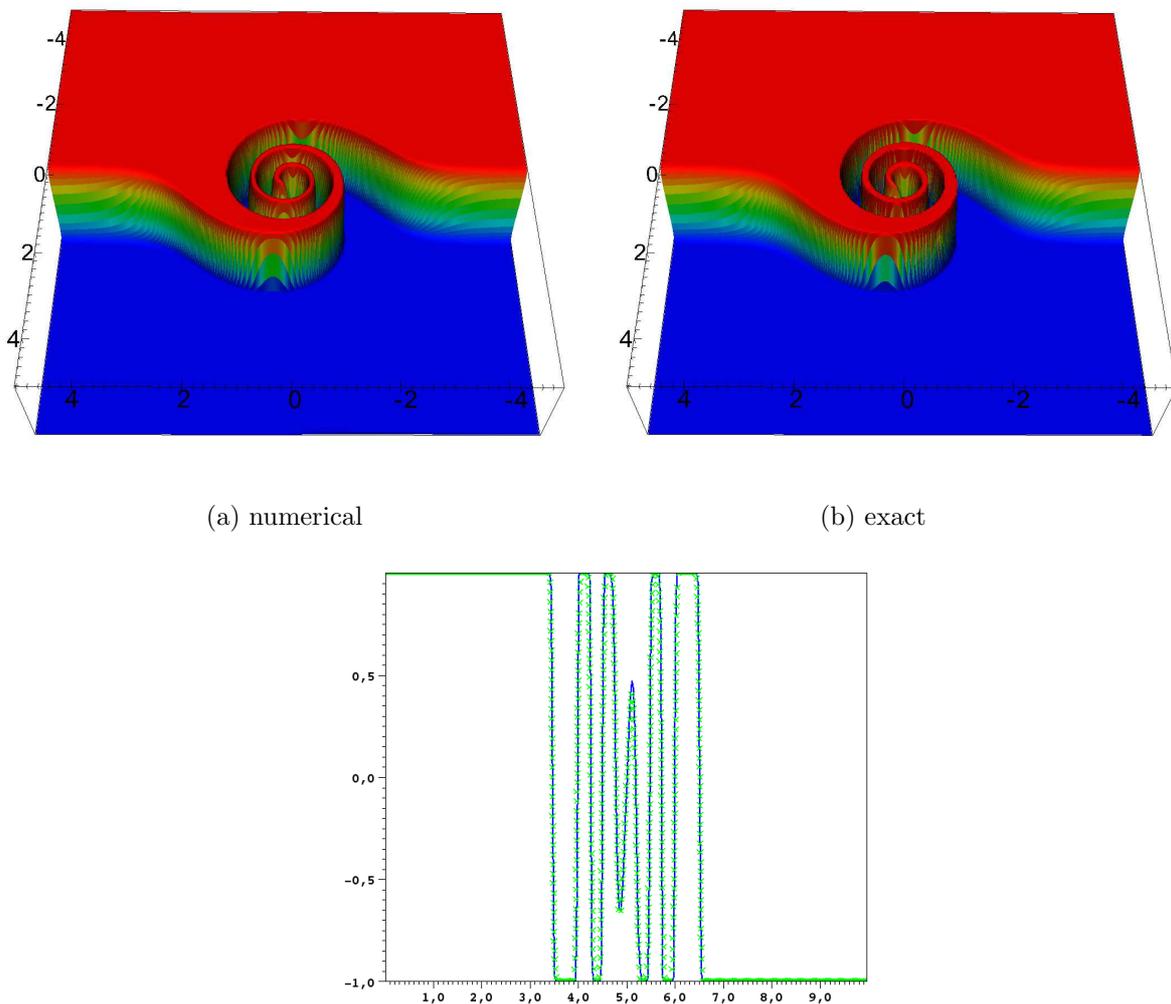


Figure 3.5: (a-b) Results for the numerical and the exact solution for the kinematic cyclogenesis test problem. (c) A slice cut through the centre of both (blue line representing the exact solution); $[-5, 5] \times [-5, 5]$ domain, 300×300 cells, $c = 0.9$ and $t = 15.0$.

Chapter 4

Atmospheric Model With Adaptive Mesh Refinement

4.1 Overview

The objective of this chapter is to simulate processes of global atmospheric transport. The chemical transport model (CTM) describes chemical reactions and chemical processes within the atmosphere. Meteorological analysis provides the CTM with input such as wind velocities. The mathematical representation is based on the continuity equations of mass conservation of the chemicals. By solving these equations accurately one can obtain information about future atmospheric composition [14, 6]. Thus, a high degree of accuracy is desirable. As high resolution simulations are computationally expensive, computational power is a bottle neck. For this reason, alternative approaches are investigated to increase the efficiency. After the presentation of the governing equations, the focus lies on adaptive mesh refinement (AMR), which is a technique to increase the efficiency of the model. Subsequently, two physical test problems, for which analytical solutions are available, are assessed in two different spherical geometries. Finally, the results for the regular latitude-longitude grid and the cubed sphere representation are discussed.

4.2 Governing Equations In Spherical Geometry

The major foundations for the extension to spherical geometry have already been introduced for the variable coefficient approach in Chapter 3. For a regular latitude(θ)-

longitude(λ)-height(r) structured grid an additional dimension is added to (3.19).

$$\begin{aligned}
 \Psi_{ijk}^{n+1} = & \Psi_{ijk}^n - \frac{\Delta t}{V_{ijk}} \times ([A(\Psi u)^*]_{i+1/2jk}^n - [A(\Psi u)^*]_{i-1/2jk}^n \\
 & + [A(\Psi v)^*]_{ij+1/2k}^n - [A(\Psi v)^*]_{ij-1/2k}^n \\
 & + [A(\Psi v)^*]_{ijk+1/2}^n - [A(\Psi v)^*]_{ijk-1/2}^n \\
 & + \Psi_{ijk}^n [Au^*]_{i+1/2jk}^n - \Psi_{ijk}^n [Au^*]_{i-1/2jk}^n \\
 & + \Psi_{ijk}^n [Av^*]_{ij+1/2k}^n - \Psi_{ijk}^n [Av^*]_{ij-1/2k}^n \\
 & + \Psi_{ijk}^n [Av^*]_{ijk+1/2}^n - \Psi_{ijk}^n [Av^*]_{ijk-1/2}^n)
 \end{aligned} \tag{4.1}$$

The volumes V_{ijk} as well as the face areas are given by

$$V_{ijk} = \frac{1}{3} \Delta \lambda [\sin \theta_{j+1/2} - \sin \theta_{j-1/2}] \left[3r_k^2 \Delta r + \frac{(\Delta r)^3}{4} \right] \tag{4.2}$$

$$A_{i+1/2jk} = r_k \Delta \theta \Delta r \tag{4.3}$$

$$A_{ij+1/2k} = r_k \cos \theta_{j+1/2} r_k \Delta \lambda \Delta r \tag{4.4}$$

$$A_{ijk+1/2} = r_{k+1/2}^2 \Delta \lambda [\sin \theta_{j+1/2} - \sin \theta_{j-1/2}] \tag{4.5}$$

where Δr , $\Delta \lambda$ and $\Delta \theta$ are constant over the whole domain. Another modification has to be made for the local CFL number. Whereas we used (3.23) to calculate the CFL number in Cartesian geometry, we now make a modification using the above formulas which yields

$$c_{i+1/2} = u_{i+1/2} \Delta t \frac{2A_{i+1/2}}{V_i + V_{i+1}} \tag{4.6}$$

where $c_{i+1/2}$ stands for the CFL number, $u_{i+1/2}$ for the velocity at the interface, $A_{i+1/2}$ for the face area at the interface and V_i and V_{i+1} for the volumes of the adjacent cells. All modifications which are necessary for the extension to spherical coordinates have now been made [8].

4.3 Adaptive Mesh Refinement (AMR)

Adaptive Mesh Refinement is a technique to locally increase the resolution near the region of interest. This enables one to capture features, such as shock waves, more accurately without the need of uniform grid refinement. The algorithm should also be able to follow sensitive regions dynamically. That is why monitor functions are defined to identify areas which need refinement. First, the requirements on the grid and the nesting of finer patches are defined. Subsequently, the integration of fine patches and the different levels of refinement are described. Finally, a method for the dynamic regridding process is introduced. The AMR algorithm described here bases mostly on the work of Berger and Collela (1989) [10] and Hubbard and Nikiforakis (2003) [8].

4.3.1 Nesting

The whole computational domain is covered by a set of hierarchically nested, logically rectangular patches of cells, called meshes. For simplicity the sides of these meshes are required to lie in the coordinate directions. A gridlevel G_l can be defined as a finite union of grids $G_{l,k}$

$$G_l = \bigcup_k G_{l,k} \quad (4.7)$$

where $l = 0, \dots, l_{max}$ is the sequence of levels. With this terminology G_0 corresponds to the computational domain D . Although meshes at the same gridlevel are in general allowed to overlap, $G_{l,j} \cap G_{l,k} = \emptyset \forall j, k$ is required. Moreover, there are some restrictions on the nesting of grids on the same and on different levels. The corners of the finer grid must coincide with the ones of the next coarser grid. In addition, a level $l-2$ grid must always be separated from a level l grid by at least one grid cell of a level $l-1$ grid.

Each gridlevel G_l may have its individual time t_l and resolution x_l , but all grids are refined by the same mesh refinement factor $r = \frac{\Delta x_{l-1}}{\Delta x_l}$ and therefore

$$\frac{\Delta t_l}{\Delta x_l} = \frac{\Delta t_{l-1}}{\Delta x_{l-1}} = \dots = \frac{\Delta t_1}{\Delta x_1}. \quad (4.8)$$

In this model multiple grids may contain one point $p \in D$, but only the finest grid containing the point will be taken as solution. In Fig. 4.1a an example of an adapted grid is shown.

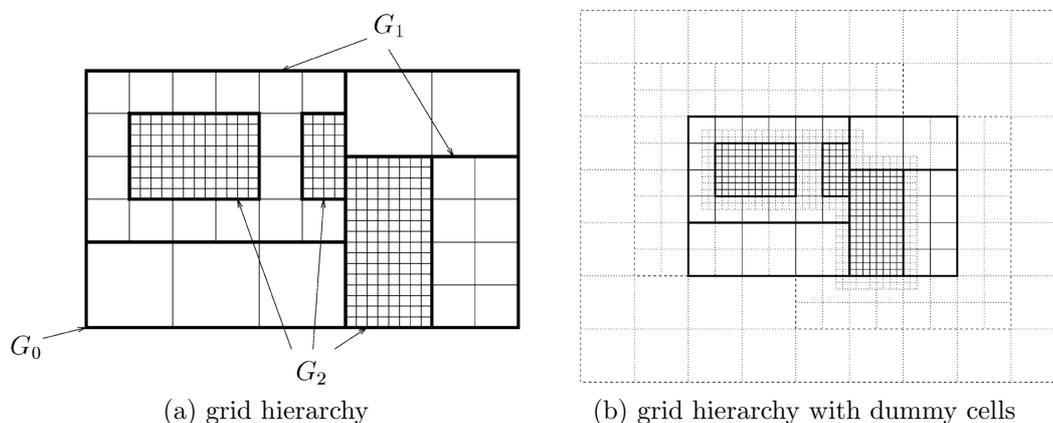


Figure 4.1: Proper nesting of mesh patches a) without b) with superimposed dummy cells [8].

4.3.2 Integration

Advancing the coarsest gridlevel G_0 to the time $t + \Delta t$ needs to be done in a certain manner. As in the previous chapters, a conservative scheme which is easy to integrate in the AMR

algorithm is used. Since a general approach would be beyond the scope of this thesis, the integration procedure is explained using the example of [8] illustrated by Fig. 4.1a. For the given setting of three gridlevels G_0 , G_1 and G_2 with the refinement factors 2 and 4, the steps summarized by tab. 4.1 need to be assessed in order to do the integration step.

Integrate	G_0 with time step Δt
Integrate	G_1 with time step $\Delta t/2$
Integrate	G_2 four times with time step $\Delta t/8$
Project	G_2 on to G_1
Adapt	G_2
Integrate	G_1 with time step $\Delta t/2$
Integrate	G_2 four times with time step $\Delta t/8$
Project	G_2 on to G_1
Project	G_1 on to G_0
Adapt	G_2
Adapt	G_1

Table 4.1: Series of operations necessary to advance the coarsest gridlevel G_0 to $t + \Delta t$ in time [8].

Assuming that all mesh patches are already initialized, each gridlevel is integrated separately with a suitable time step. At this stage, G_2 and G_1 are advanced to the same time and the next step is the projection of the data contained by G_2 on to G_1 . This can be achieved by replacing the coarse grid value of the G_1 grid cell by the conservative average of the finer G_2 grid cells covering the coarse cell. For the two-dimensional example in Fig. 4.3a and a refinement factor r , this conservative average could be [10]

$$\Psi_{i,j}^{coarse} = \frac{1}{r^2} \sum_{p=0}^{r-1} \sum_{q=0}^{r-1} \Psi_{k+p,m+q}^{fine} \quad (4.9)$$

although a volume-weighted average might be preferable. The clustering of G_2 can now be recalculated. Afterwards the fine gridlevels are advanced to $t + \Delta t$ and the G_2 data is projected on to G_1 and subsequently G_1 on to G_0 . Finally, the fine gridlevels may be adapted again. In order to update cells on the borders of a grid, dummy layers surrounding the grids, which depend on the order of the underlying scheme, are required (Fig. 4.3.2b). The values for the dummy cells are directly taken from the finest underlying mesh or interpolated, eg. bilinearly or trilinearly in space [8].

One problem arises at coarse/fine cell boundaries, because the sum of the fine grid intercell fluxes might deviate from the coarse intercell flux. This needs a correction, since the scheme is intended to be conservative. Again, this is explained by using a two-dimensional

example of [10] illustrated by 4.3b. Similarly to (1.18), the corrected update formula is

$$\begin{aligned} \Psi_{i,j}(t + \Delta t_{coarse}) = & \Psi_{i,j}(t) - \frac{\Delta t_{coarse}}{\Delta x} \left[f_{i+\frac{1}{2},j}(t) - \frac{1}{r^2} \sum_{p=0}^{r-1} \sum_{q=0}^{r-1} f_{k+\frac{1}{2},m+p}(t + q\Delta t_{fine}) \right] \\ & - \frac{\Delta t_{coarse}}{\Delta y} \left[g_{i,j+\frac{1}{2}}(t) - g_{i,j-\frac{1}{2}}(t) \right] \end{aligned} \quad (4.10)$$

where f and g denote the fluxes in both space dimensions.

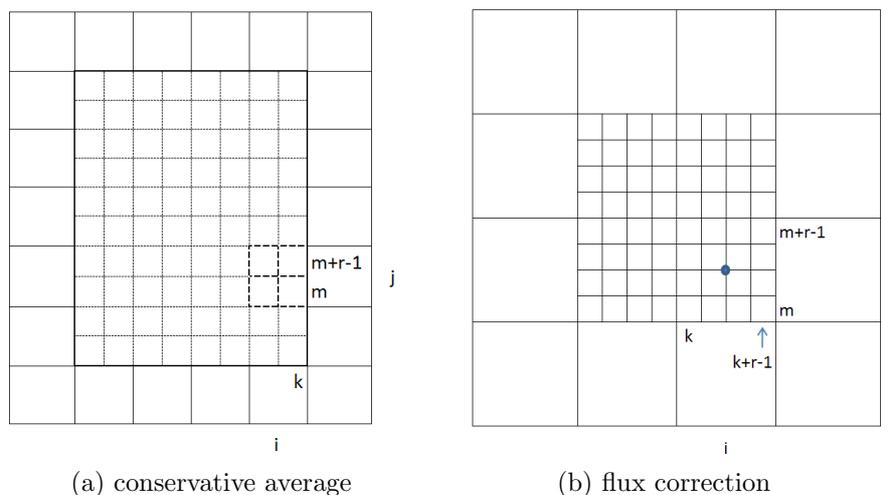


Figure 4.2: a) Replacement of the coarse value of cell (i, j) by the conservative average of the finer gridlevel cells b) Flux correction at a coarse/fine cell boundary [10] (modified).

4.3.3 Regridding

The algorithm should keep track of rapidly changing areas that need refinement. This can be achieved by defining a monitor function that is able to recognize cell-cell differences above a certain level. These cells are flagged and subsequently split in a way such that the smallest possible rectangle is fit around each cluster of flagged cells.

Following [11], this is done by a bijection algorithm that uses the signatures Σ_x and Σ_y which are just the sums of flagged cells in each row and column. Another criteria is the second derivative of the signature, denoted by Δ . Given the rectangular cluster in Fig. 4.3, one wants to decide how it can be efficiently split into smaller rectangular clusters. The flagged cells are marked and the signatures as wells as the second derivatives are calculated for both directions.

Per definition, a zero in the signature means that there is either a whole row or a column without flagged cells. Thus, the cluster can be split into two separate clusters. If the

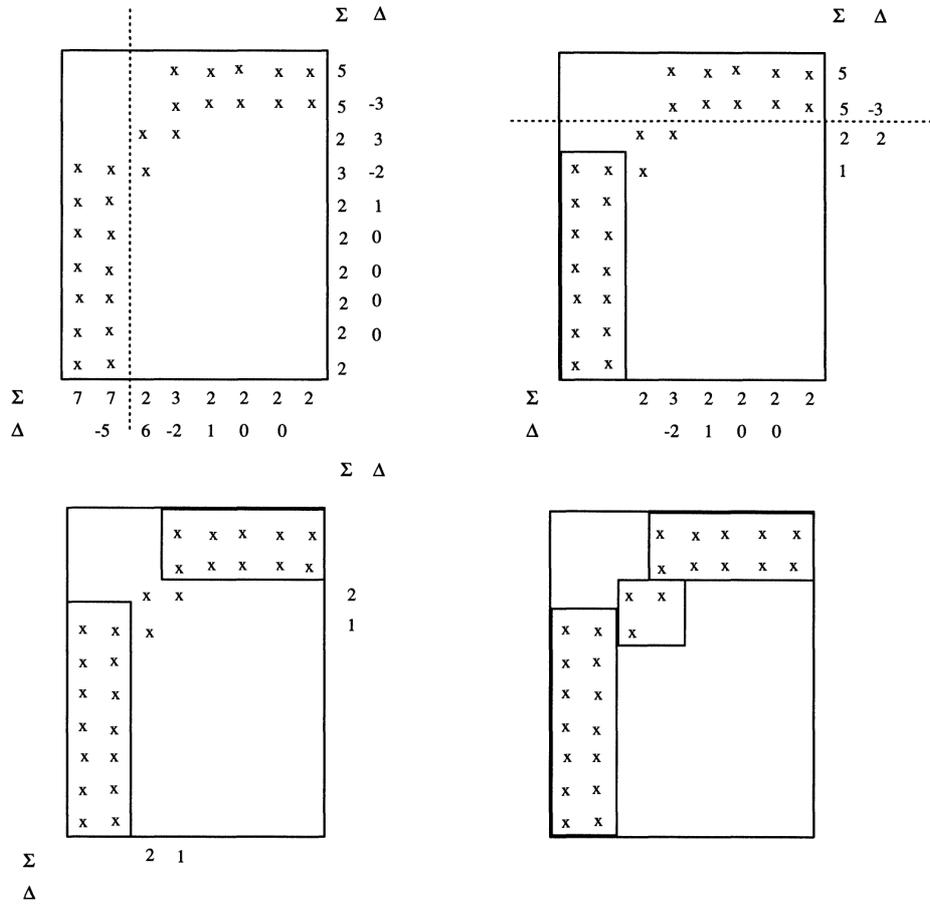


Figure 4.3: Partitioning of the clusters on the basis of the signature arrays Σ_x, Σ_y and the second derivatives Δ_x, Δ_y [11].

signature is nonzero, an edge of a new cluster is indicated by a zero crossing in the second derivative. In our example the procedure stops after having found efficient subclusters in three steps. Of course, additional calculations need to be done in order to determine, if the efficiency of a certain set of clusters is sufficient.

4.4 Cubed Sphere Method

There are some restrictions on solving PDEs on a regular latitude-longitude grid, one of which is known as the pole convergence. The meridians converge in the vicinity of the poles which is why the use of very small time steps is necessary. In order to circumvent this problem, new mapping techniques and numerical methods are investigated. This section deals with a representation called the "cubed sphere", which was first proposed by Ronchi, Iacono and Paolucci (1996) [9].

Following [9], the sphere can be decomposed into six identical regions with local angular variables $(\xi, \eta) \in [-\pi/4, \pi/4] \times [-\pi/4, \pi/4]$ as shown in Fig. 4.4b. Each region is constructed by the intersection of two sets of angularly equidistant circles, which cover the whole sphere when put together.

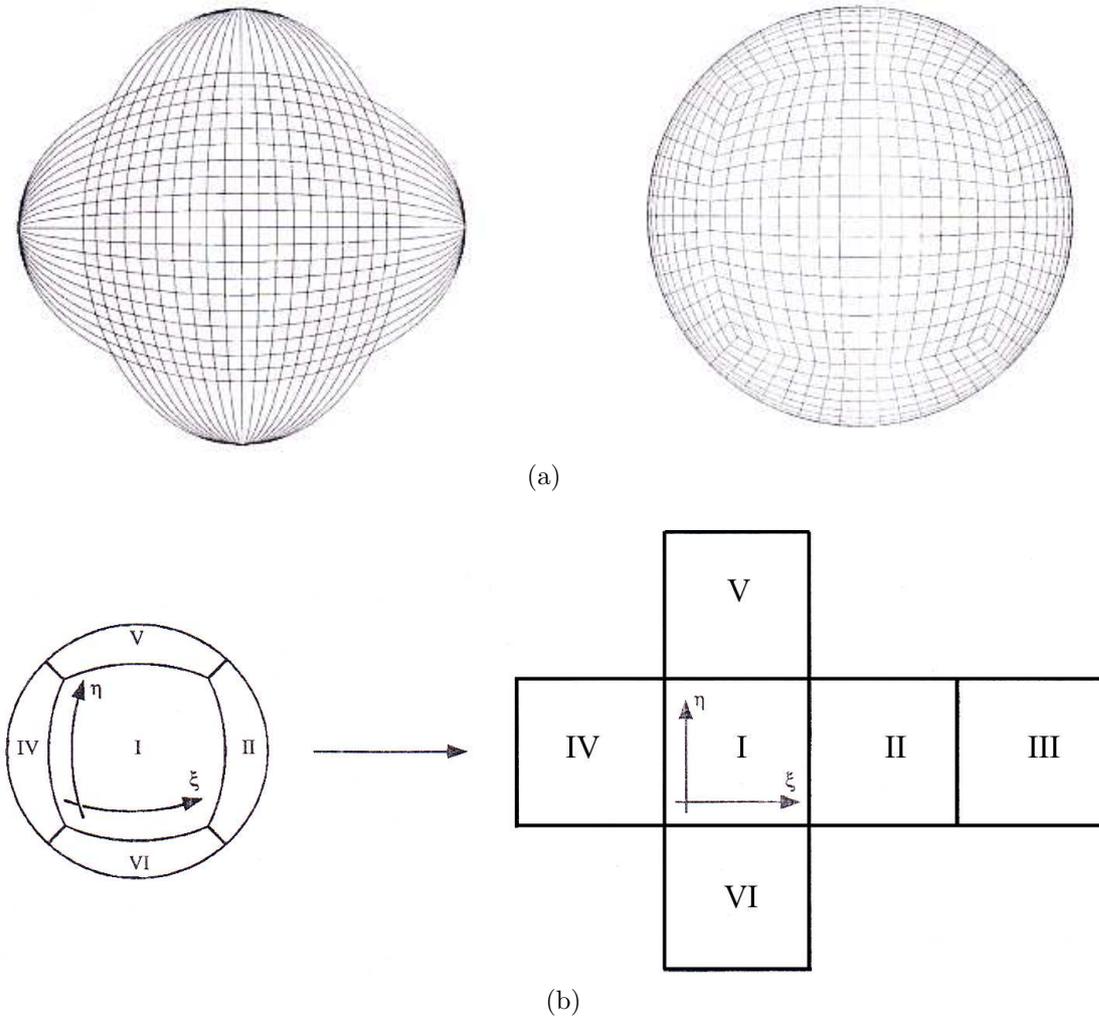


Figure 4.4: (a) Construction of one of the six faces (b) Spherical surface consisting of six components [9].

With the definitions of

$$\begin{aligned}
 X &\equiv \tan(\xi) \\
 Y &\equiv \tan(\eta) \\
 \delta &\equiv 1 + X^2 + Y^2 \\
 C &\equiv (1 + X^2)^{\frac{1}{2}} \\
 D &\equiv (1 + Y^2)^{\frac{1}{2}}
 \end{aligned} \tag{4.11}$$

the transformations from cartesian (x, y, z) and regular spherical (Θ, Φ, r) coordinates to the (ξ, η, r) coordinates of the cubed sphere are given analytically by (4.12).

$$\begin{aligned}
 \text{I (Equator)} & \quad \begin{cases} X = \frac{y}{x} = \tan(\Phi) \\ Y = \frac{z}{x} = \frac{1}{\tan(\Theta) \cos(\Phi)} \end{cases} \\
 \text{II (Equator)} & \quad \begin{cases} X = -\frac{x}{y} = -\frac{1}{\tan(\Phi)} \\ Y = \frac{z}{y} = \frac{1}{\tan(\Theta) \sin(\Phi)} \end{cases} \\
 \text{III (Equator)} & \quad \begin{cases} X = \frac{y}{x} = -\frac{1}{\tan(\Phi)} \\ Y = -\frac{z}{y} = -\frac{1}{\tan(\Theta) \cos(\Phi)} \end{cases} \quad (4.12) \\
 \text{V (North Pole)} & \quad \begin{cases} X = \frac{y}{z} = \tan(\Theta) \sin(\Phi) \\ Y = -\frac{x}{z} = -\tan(\Theta) \cos(\Phi) \end{cases} \\
 \text{V (South Pole)} & \quad \begin{cases} X = -\frac{y}{z} = -\tan(\Theta) \sin(\Phi) \\ Y = -\frac{x}{z} = -\tan(\Theta) \cos(\Phi) \end{cases}
 \end{aligned}$$

The radial component $r = (x^2 + y^2 + z^2)^{\frac{1}{2}}$ is the same for both spherical coordinate systems. For each region exists a set of unit base vectors $\{\vec{e}_\xi, \vec{e}_\eta, \vec{e}_r\}$ such that the metric tensor remains the same on all regions

$$g = \begin{pmatrix} 1 & -XY/CD & 0 \\ -XY/CD & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.13)$$

Furthermore, the transformation laws for vectors from (Θ, Φ, r) to (ξ, η, r) coordinates are given by

$$\begin{aligned}
 \text{I-IV (Equator)} & \quad \begin{pmatrix} A_\xi \\ A_\eta \end{pmatrix} = \begin{pmatrix} 0 & CD/\delta^{\frac{1}{2}} \\ -1 & XY/\delta^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} A_\Theta \\ A_\Phi \end{pmatrix} \\
 \text{V (North Pole)} & \quad \begin{pmatrix} A_\xi \\ A_\eta \end{pmatrix} = \frac{1}{(\delta-1)^{\frac{1}{2}}} \begin{pmatrix} DX & -DY/\delta^{\frac{1}{2}} \\ CY & CX/\delta^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} A_\Theta \\ A_\Phi \end{pmatrix} \quad (4.14) \\
 \text{VI (South Pole)} & \quad \begin{pmatrix} A_\xi \\ A_\eta \end{pmatrix} = \frac{1}{(\delta-1)^{\frac{1}{2}}} \begin{pmatrix} -DX & DY/\delta^{\frac{1}{2}} \\ -CY & -CX/\delta^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} A_\Theta \\ A_\Phi \end{pmatrix}.
 \end{aligned}$$

4.5 Test problems for the latitude-longitude and cubed sphere representation

The CAMR framework is a general and extendible implementation of the AMR technique, which allows to run all discussed test problems in any coordinate system. The implementation of the algorithm for the cubed sphere, for example, is more complex than for regular spherical coordinates with regards to dummy cells, boundary conditions and interpolation. However, with some modifications, the scheme (4.1) can also be used for the cubed sphere.

4.5.1 Solid Body Rotation

The first test problem, which is widely used, is the advection of a cosine bell, which could represent the concentration of a certain quantity. Fortunately, one only needs to know the analytical solution for the regular spherical coordinates since one can easily change to the cubed sphere with (4.12) and (4.14). Following [12], the initial profile is given by

$$\Psi(\Theta, \Phi, r) = \begin{cases} \frac{1}{2} [1 + \cos(\frac{\pi\lambda}{R})] & \text{if } \lambda < R, \\ 0 & \text{if } \lambda \geq R \end{cases} \quad (4.15)$$

where

$$\lambda = \cos^{-1} [\sin \Theta_c \sin \Theta + \cos \Theta_c \cos \Theta \cos(\lambda - \lambda_c)] \quad (4.16)$$

is the great circle distance between the bell centre (Θ_c, Φ_c, r) and the position (Θ, Φ, r) and $R = \frac{7\pi}{64}$ is the bell radius. The tangential velocity u and the normal velocity v are

$$u = u_0(\cos \alpha \cos \Theta + \sin \alpha \cos \Phi \sin \Theta) \quad (4.17)$$

$$v = -u_0 \sin \alpha \sin \Phi \quad (4.18)$$

where u_0 is an arbitrary parameter and $\alpha \in [0, \pi]$ defines the advection angle.

4.5.2 Kinematic Cyclogenesis

The kinematic cyclogenesis test problem was already introduced for cartesian coordinates in conjunction with the variable coefficient extension. However, the formulas change for regular spherical coordinates. Following [12], the coordinate system (Θ, Φ, r) is rotated first such that the centre of the vortex coincides with the north pole of the new coordinate system (Θ^*, Φ^*, r) , where (Θ_0, Φ_0, r) is the position of the vortex with respect to the original coordinates. The only nonzero velocity component is the tangential one defined as

$$u^*(\Theta^*) \equiv \cos \Theta^* \frac{d\lambda^*}{dt} = \frac{3\sqrt{3}}{2} \operatorname{sech}^2(\gamma\rho) \tanh(\gamma\rho) \quad (4.19)$$

where

$$\rho(\Theta^*) = \frac{2 \cos \Theta^*}{1 + \sin \Theta^*} \quad (4.20)$$

and γ is a constant parameter. The simulations were run for a fixed radius and the radial component is only cited to complete the picture. The exact solution in the new coordinates is given by

$$\Psi(\Theta^*, \Phi^*, r, t) = -\tanh \left[\frac{\rho}{\delta} \sin \left(\Phi^* - \frac{u^*(\Theta^*)t}{\cos \Theta^*} \right) \right] \quad (4.21)$$

and finally, after changing back to the original coordinates, one gets $\Psi(\Theta, \Phi, r, t)$.

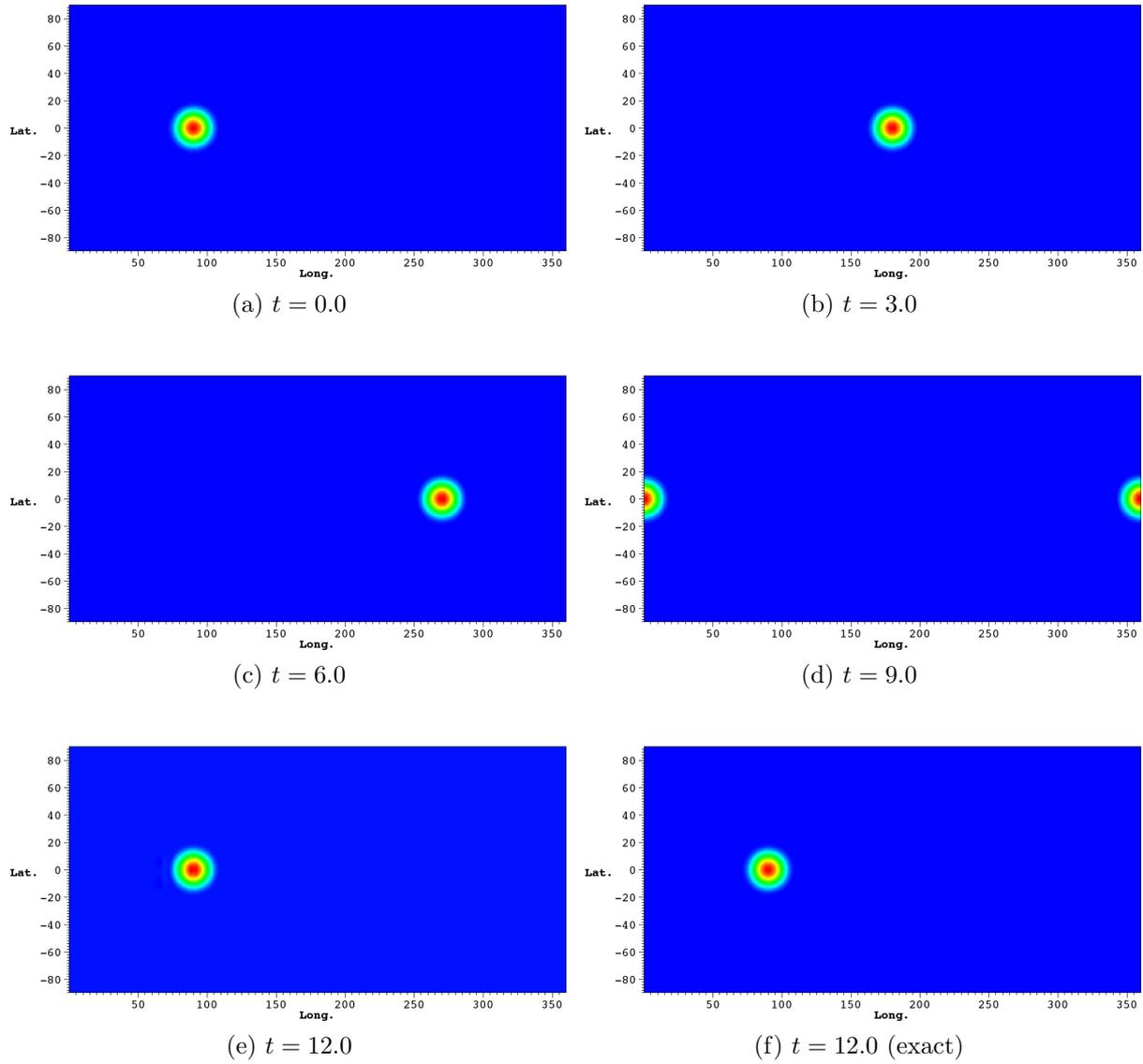


Figure 4.5: Solid body advection of a cosine bell on the regular sphere with a resolution of 128×64 cells, AMR refinement 2×2 (two levels), $\alpha = 0^\circ$, $(\Theta_c, \Phi_c) = (0, 90)$; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0 (f) shows the exact solution at $t=12.0$.

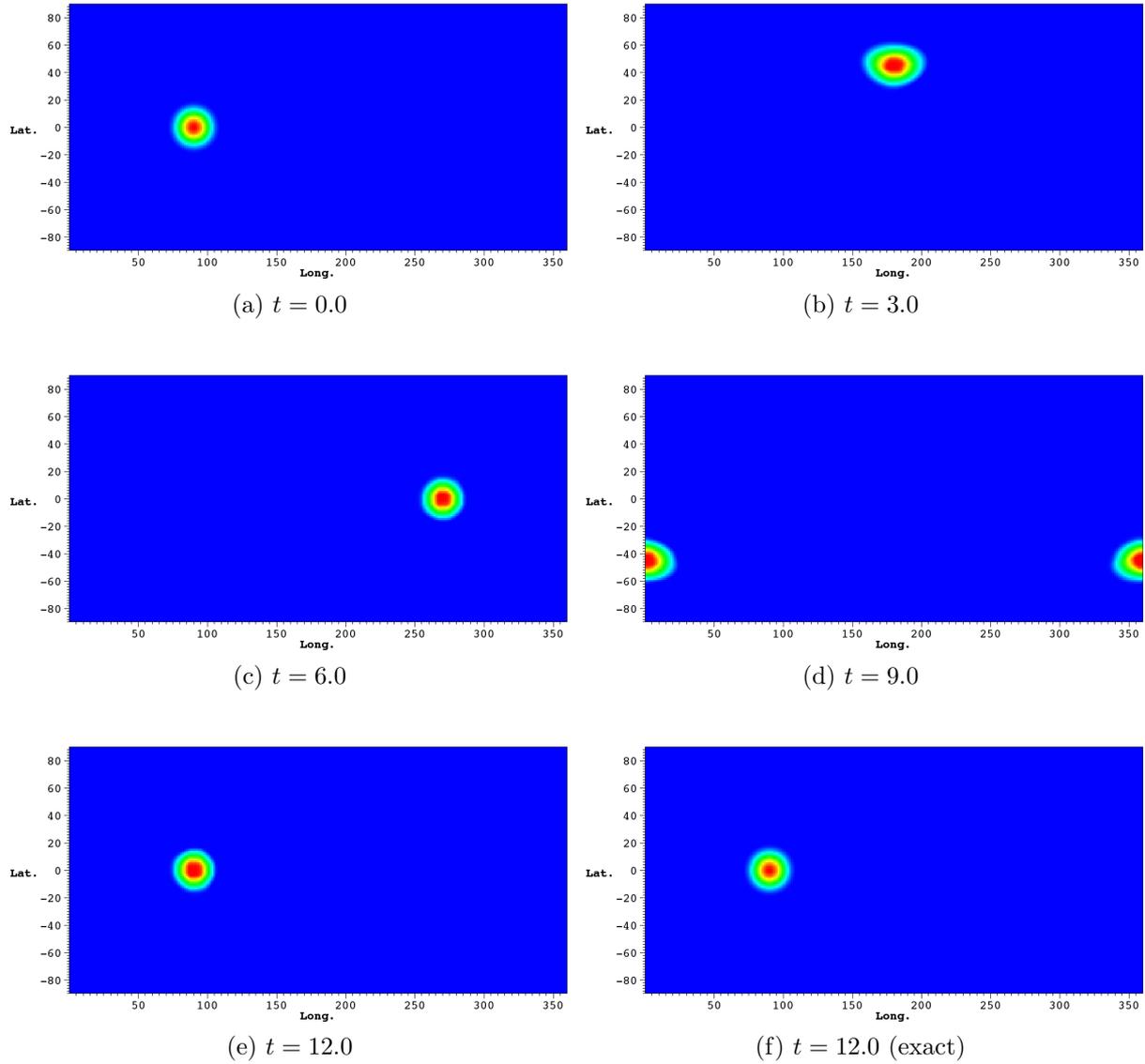


Figure 4.6: Solid body advection of a cosine bell on the regular sphere with a resolution of 256×128 cells, no AMR refinement, $\alpha = 45^\circ$, $(\Theta_c, \Phi_c) = (0, 90)$; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0 (f) shows the exact solution at $t=12.0$.

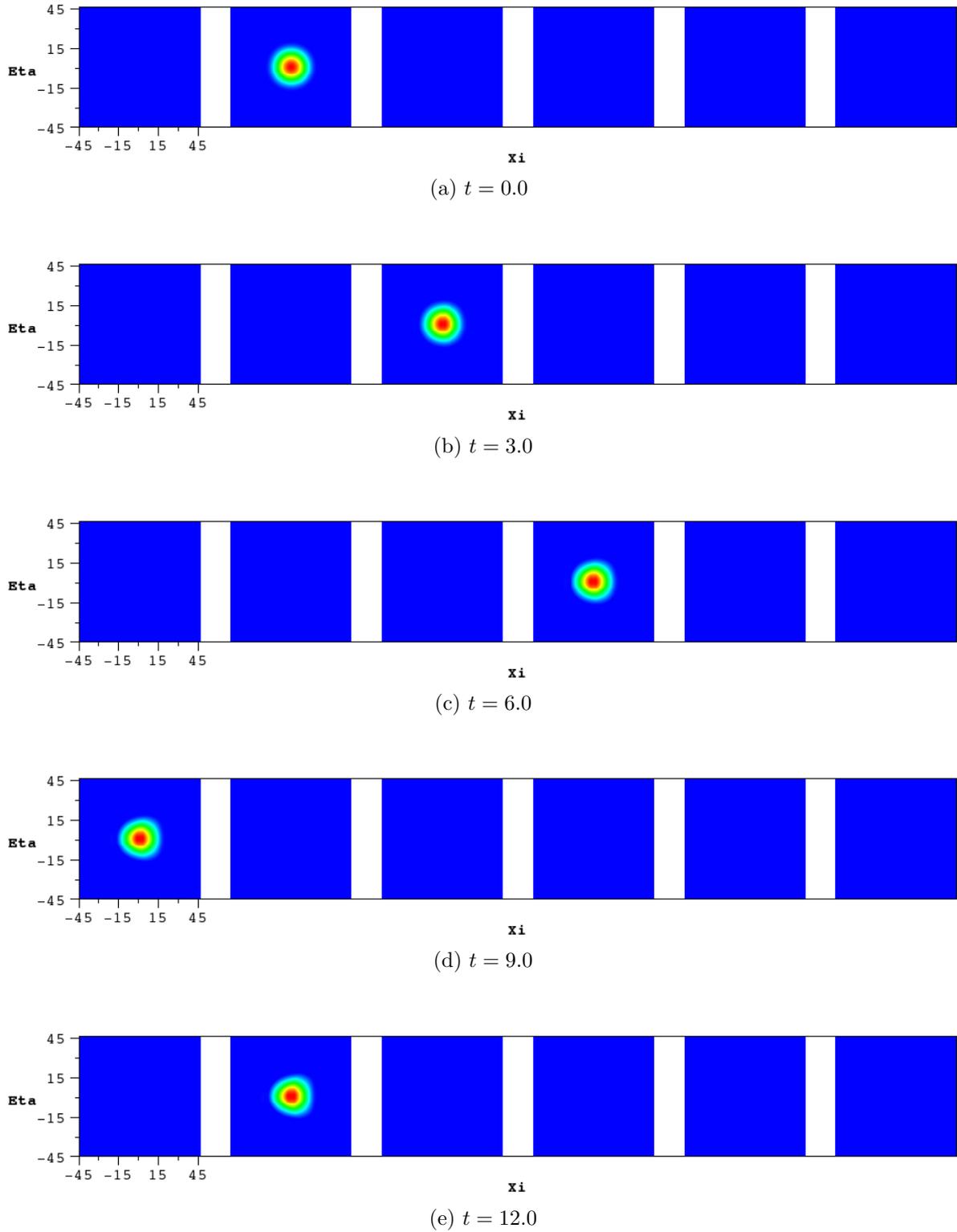


Figure 4.7: Solid body advection of a cosine bell on the cubed sphere with a face resolution of 32×32 cells, AMR refinement 2×2 (two levels), $(\Theta_c, \Phi_c) = (0, 90)$, $\alpha = 0^\circ$; the fifth face in each row represents the north pole and the last one the south pole; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0 .

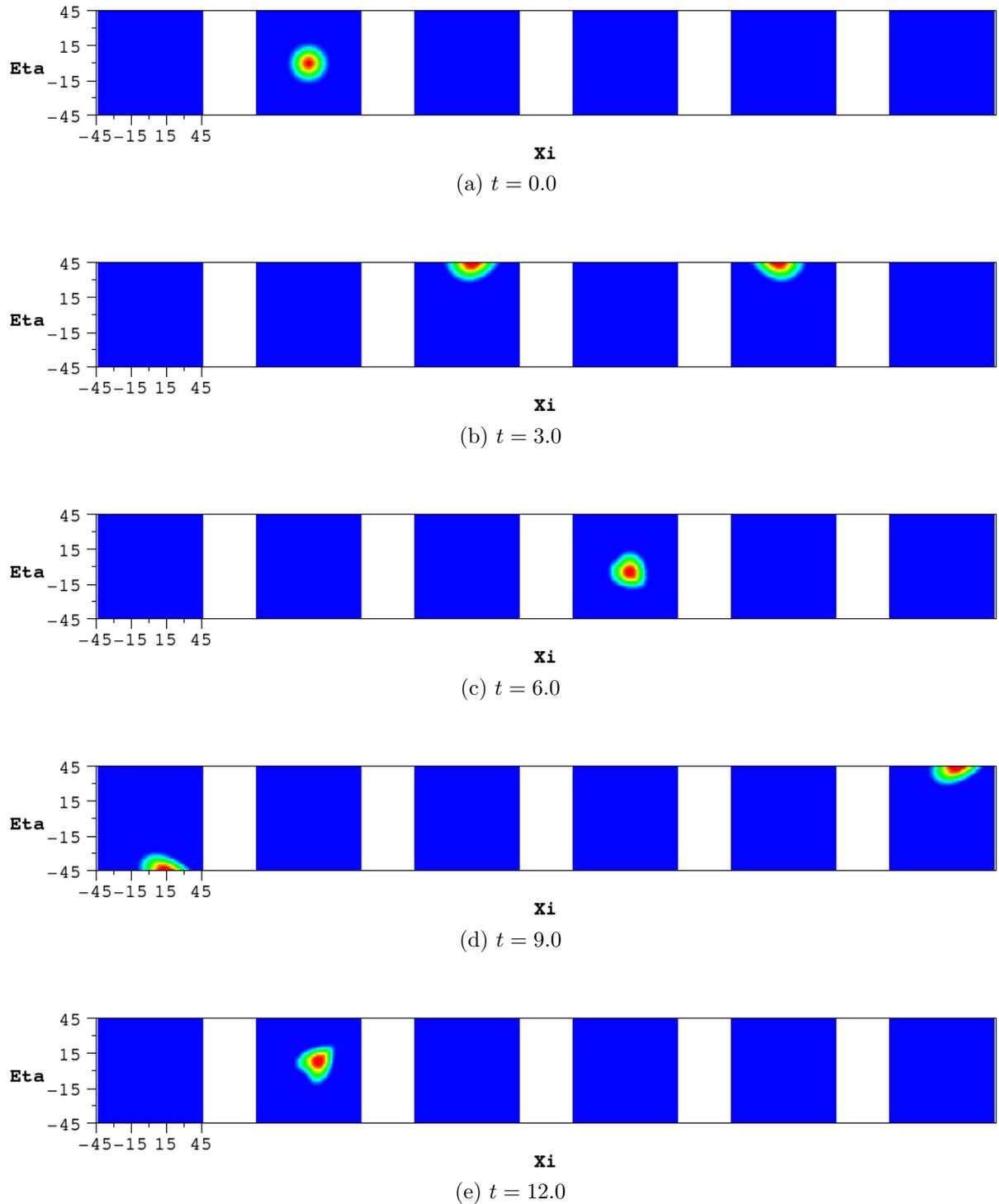


Figure 4.8: Solid body advection of a cosine bell on the cubed sphere with a face resolution of 64×64 cells, no AMR refinement, $(\Theta_c, \Phi_c) = (0, 90)$, $\alpha = 45^\circ$; the fifth face in each row represents the north pole and the last one the south pole; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0 .

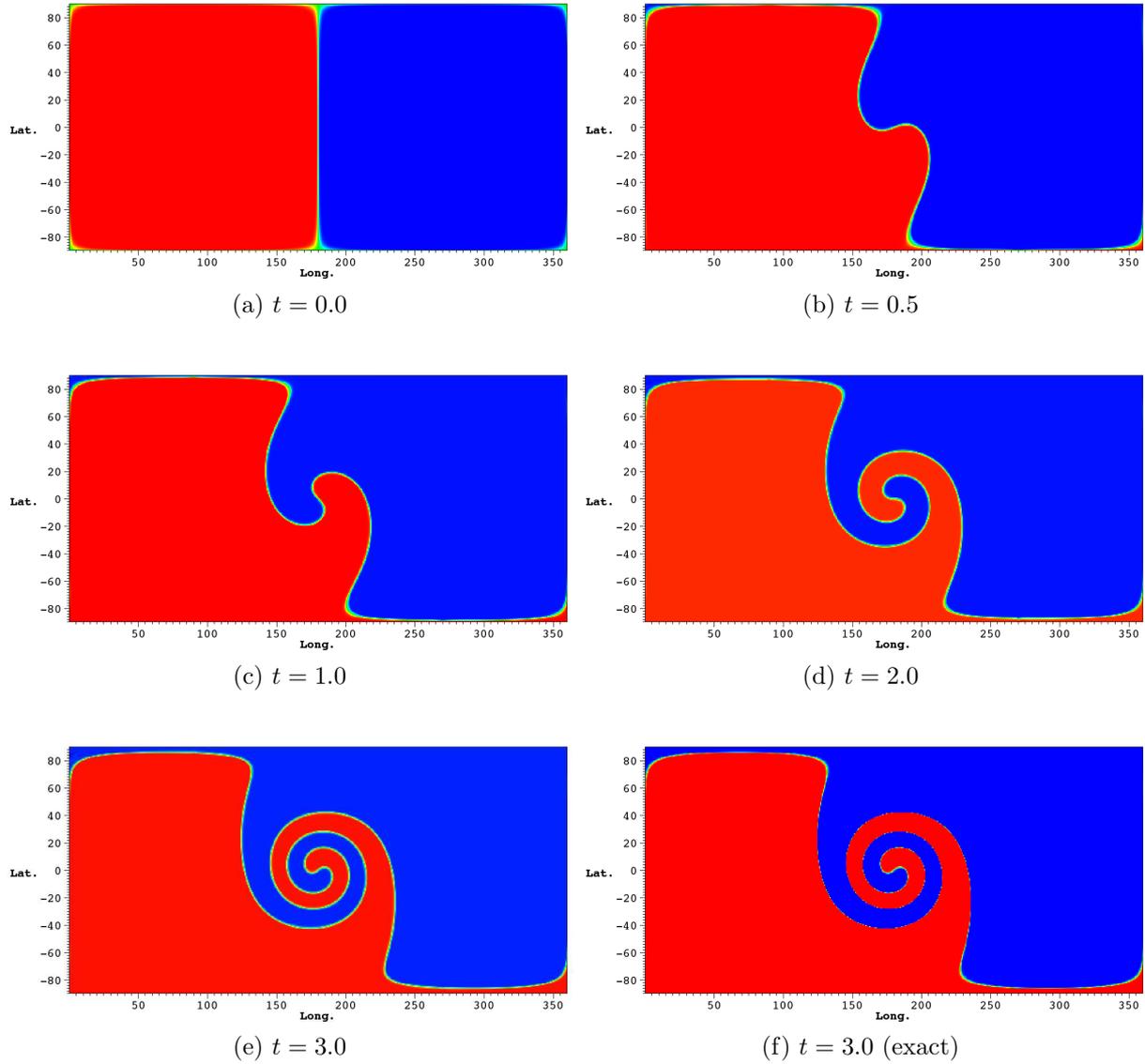
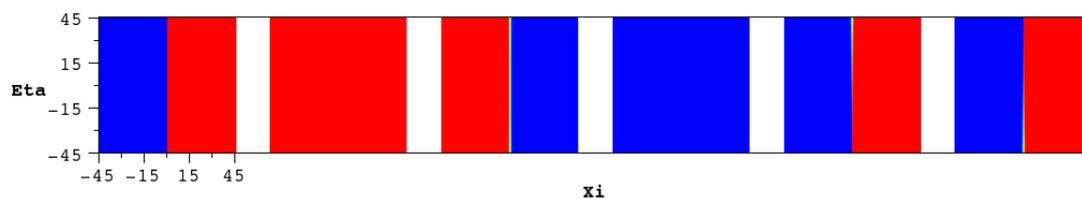
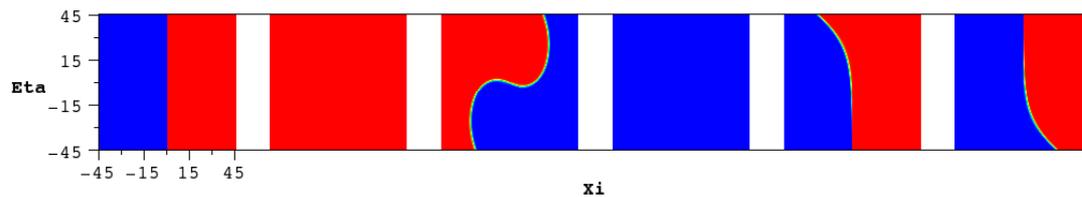


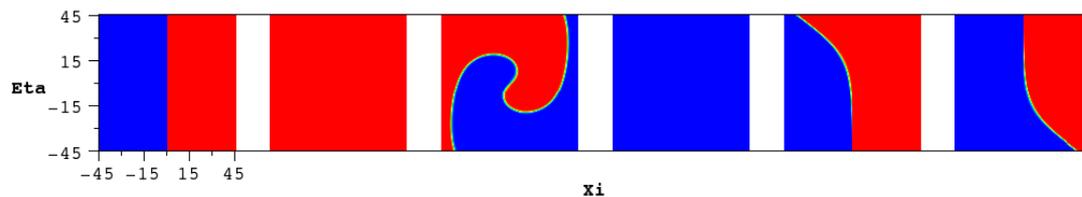
Figure 4.9: Kinematic cyclogenesis on the regular sphere with a resolution of 256×128 cells, AMR refinement 2, $\delta = 0.01$, $(\Theta_c, \Phi_c) = (0, 180)$; (a-e) show the numerical solution at times $t=0.0, 0.5, 1.0, 2.0, 3.0$ (f) shows the exact solution at $t=3.0$.



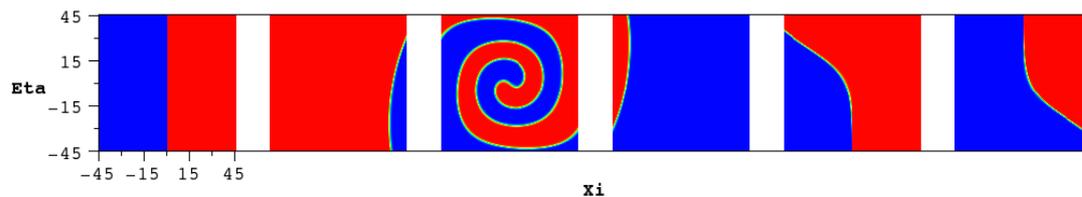
(a) $t = 0.0$



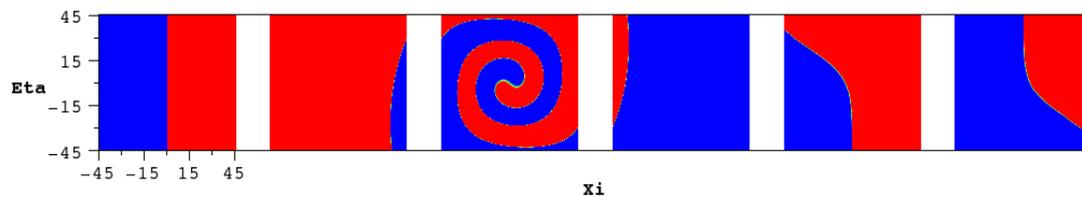
(b) $t = 1.0$



(c) $t = 2.0$



(d) $t = 3.0$



(e) $t = 3.0$ (exact)

Figure 4.10: Kinematic cyclogenesis on the cubed sphere with a face resolution of 64×64 cells, AMR refinement 2, $\delta = 0.01$, $(\Theta_c, \Phi_c) = (0, 180)$; the fifth face in each row represents the north pole and the last one the south pole; (a-d) show the numerical solution at times $t=0.0, 0.5, 1.0, 3.0$ (e) shows the exact solution at $t=3.0$.

4.6 Discussion

The solid body rotation test problem was run twice for each spherical geometry. Figures 4.5 and 4.7 present the results for equatorial advection with two levels of refinement and an effective resolution of 512×256 cells. As the chart shows, the test for the regular sphere was successful and the numerical solution differs only slightly from the exact one after one full cycle. Unfortunately, the results for the cubed sphere are not satisfying yet. After one cycle the bell is already distorted, as shown in 4.7e.

As for an advection angle of 45° , the situation becomes even worse. For both representations the simulations were run without AMR, but with a higher base resolution of 256×128 cells. As shown in Fig. 4.8, a strong distortion is obtained after one cycle. The reason for this is not fully understood yet. Most likely, problems arise in conjunction with the interpolation algorithm for the boundary cells. What is more, the distortion does not vanish as the resolution increases. However, the results for the regular sphere look reasonable again. Figure 4.11 illustrates the dynamic refinement during the simulations.

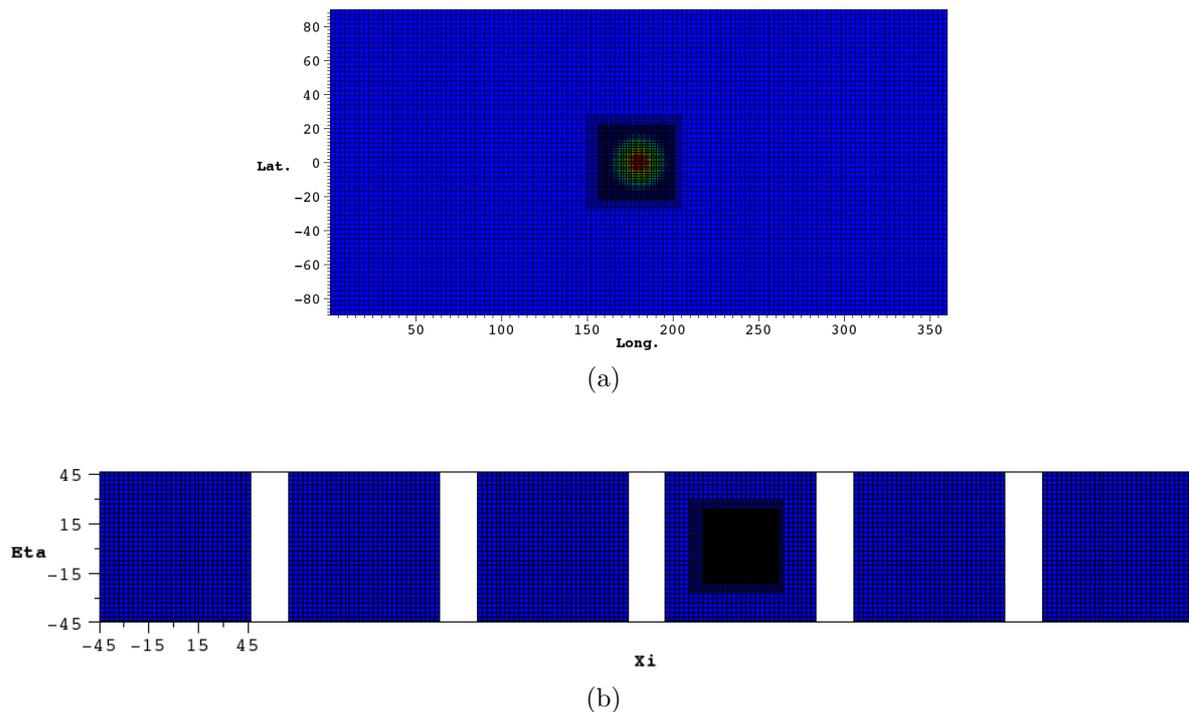


Figure 4.11: Superimposed mesh for the solid body rotation with a base resolution of 128×64 cells and two levels of refinement for (a) the regular sphere at $t = 3.0$ (b) the cubed sphere at $t = 6.0$.

Again the results for the kinematic cyclogenesis test problem look quite good for the regular

sphere (4.9). With regard to the cubed sphere, the deviation of the numerical solution from the exact one does not look as serious as in the first case. Although error norms would tell us more, they are not cited on purpose since a bug in the algorithm is very likely. Finishing the discussion, Fig. 4.12 illustrates the refinement with a superimposed mesh for both geometries.

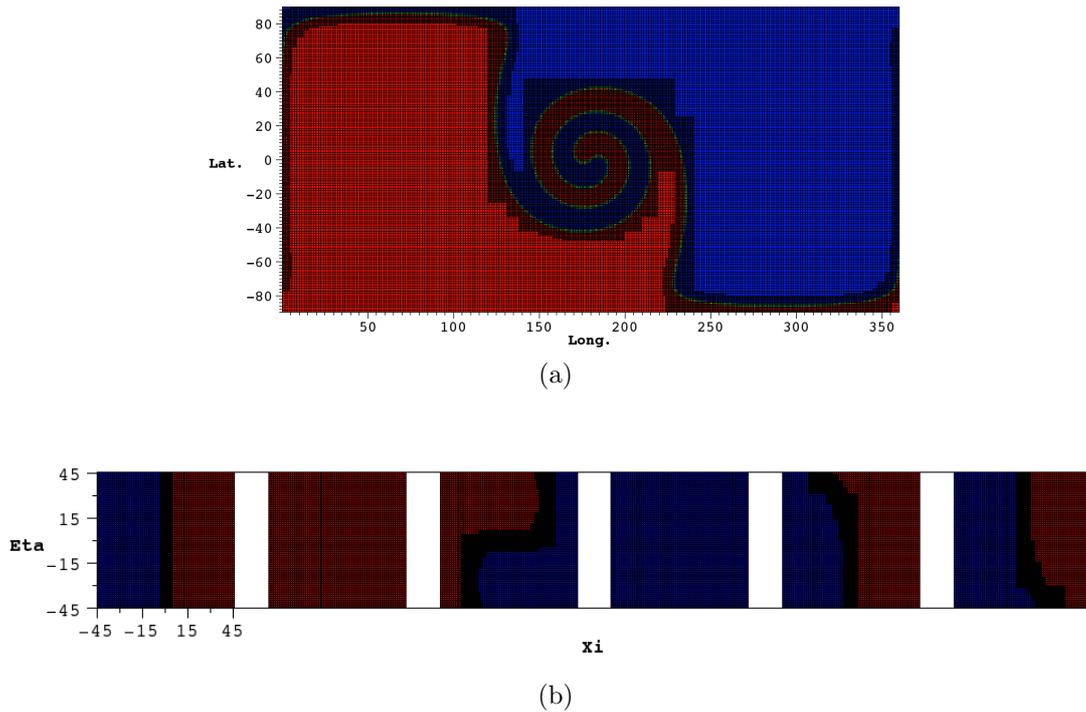


Figure 4.12: Superimposed mesh for the kinematic cyclogenesis with a base resolution of 256×128 cells and one level of refinement for (a) the regular sphere at $t = 3.0$ (b) the cubed sphere at $t = 0.5$.

List of Figures

1.1	Solution of the Riemann problem in the $x-t$ -plane [1].	7
1.2	Control volume in the $x-t$ -plane [1].	8
1.3	Piecewise constant distribution of the data [1].	9
2.1	Reconstructed locally linearly interpolated data [1].	16
2.2	Construction of the WAF-flux [1].	17
2.3	Initial profiles given by (2.53)-(2.55).	19
2.4	Solutions for the sine-wave profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme. . . .	20
2.5	Convergence tests for the sine-wave profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and $\text{cells}=2^i \cdot 25$ ($i = 0,1,\dots,8$). Each chart represents a different scheme.	21
2.6	Solutions for the top-hat profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme. . . .	22
2.7	Convergence tests for top-hat profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and $\text{cells}=2^i \cdot 25$ ($i = 0,1,\dots,8$). Each chart represents a different scheme.	23
2.8	Solutions for the combined profile at $t = 1.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme. . . .	24
2.9	Solutions for the combined profile at $t = 10.0$ and the settings $c = 0.9$, $a = 1.0$ and a resolution of 100 cells. Each chart represents a different scheme.	25
3.1	Solutions for the top-hat 2D profile with the settings 100×100 cells, $c = 0.9$, $\vec{a} = (1, 1)^T$ for 0, 0.2, 0.5, 0.8, 1 and 10 cycles.	30
3.2	Solutions for the cube profile with the settings $30 \times 30 \times 30$ cells, $c = 0.9$, $\vec{a} = (1, 1, 1)^T$ for 0.0, 0.2, 0.5, 0.8, 1.0 and 10.0 cycles.	31
3.3	Convergence tests for the (a) top-hat 2D profile with the settings $c = 0.9$, $\vec{a} = (1, 1, 1)^T$, $t = 1.0$ and 80, 160, 320 and 640 cells per direction (b) cube profile with the settings $c = 0.9$, $\vec{a} = (1, 1, 1)^T$, $t = 1.0$ and 20, 30, ..., 100 cells per direction.	32
3.4	Solutions for the kinematic cyclogenesis test problem for a $[-5, 5] \times [-5, 5]$ domain with a resolution of 300×300 cells and a CFL number of 0.9 at times $t = 0, 3, 6$ and 15.	36

3.5	(a-b) Results for the numerical and the exact solution for the kinematic cyclogenesis test problem. (c) A slice cut through the centre of both (blue line representing the exact solution); $[-5, 5] \times [-5, 5]$ domain, 300×300 cells, $c = 0.9$ and $t = 15.0$	37
4.1	Proper nesting of mesh patches a) without b) with superimposed dummy cells [8].	40
4.2	a) Replacement of the coarse value of cell (i, j) by the conservative average of the finer gridlevel cells b) Flux correction at a coarse/fine cell boundary [10] (modified).	42
4.3	Partitioning of the clusters on the basis of the signature arrays Σ_x, Σ_y and the second derivatives Δ_x, Δ_y [11].	43
4.4	(a) Construction of one of the six faces (b) Spherical surface consisting of six components [9].	44
4.5	Solid body advection of a cosine bell on the regular sphere with a resolution of 128×64 cells, AMR refinement 2×2 (two levels) , $\alpha = 0^\circ$, $(\Theta_c, \Phi_c) = (0, 90)$; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0 (f) shows the exact solution at $t=12.0$	48
4.6	Solid body advection of a cosine bell on the regular sphere with a resolution of 256×128 cells, no AMR refinement, $\alpha = 45^\circ$, $(\Theta_c, \Phi_c) = (0, 90)$; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0 (f) shows the exact solution at $t=12.0$	49
4.7	Solid body advection of a cosine bell on the cubed sphere with a face resolution of 32×32 cells, AMR refinement 2×2 (two levels), $(\Theta_c, \Phi_c) = (0, 90)$, $\alpha = 0^\circ$; the fifth face in each row represents the north pole and the last one the south pole; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0	50
4.8	Solid body advection of a cosine bell on the cubed sphere with a face resolution of 64×64 cells, no AMR refinement, $(\Theta_c, \Phi_c) = (0, 90)$, $\alpha = 45^\circ$; the fifth face in each row represents the north pole and the last one the south pole; (a-e) show the numerical solution at times $t=0.0, 3.0, 6.0, 9.0$ and 12.0	51
4.9	Kinematic cyclogenesis on the regular sphere with a resolution of 256×128 cells, AMR refinement 2, $\delta = 0.01$, $(\Theta_c, \Phi_c) = (0, 180)$; (a-e) show the numerical solution at times $t=0.0, 0.5, 1.0, 2.0, 3.0$ (f) shows the exact solution at $t=3.0$	52
4.10	Kinematic cyclogenesis on the cubed sphere with a face resolution of 64×64 cells, AMR refinement 2, $\delta = 0.01$, $(\Theta_c, \Phi_c) = (0, 180)$; the fifth face in each row represents the north pole and the last one the south pole; (a-d) show the numerical solution at times $t=0.0, 0.5, 1.0, 3.0$ (e) shows the exact solution at $t=3.0$	53
4.11	Superimposed mesh for the solid body rotation with a base resolution of 128×64 cells and two levels of refinement for (a) the regular sphere at $t = 3.0$ (b) the cubed sphere at $t = 6.0$	54

4.12 Superimposed mesh for the kinematic cyclogenesis with a base resolution of 256×128 cells and one level of refinement for (a) the regular sphere at $t = 3.0$ (b) the cubed sphere at $t = 0.5$ 55

List of Tables

2.1	Results for the convergence tests with the settings $c = 0.9$ and $a = 1.0$ at $t = 1.0$	26
4.1	Series of operations necessary to advance the coarsest gridlevel G_0 to $t + \Delta t$ in time [8].	41

Bibliography

- [1] Riemann Solvers and Numerical Methods for Fluid Dynamics, A Practical Introduction, E.F.Toro, 2nd Edition, Springer-Verlag: Berlin, Heidelberg (1999).
- [2] Computational Gasdynamics, Culbert B. Laney, Cambridge University Press, Cambridge (1998).
- [3] Lecture notes (part I) - Numerical Solution of Partial Differential Equations, Dr. N. Nikiforakis, Laboratory for Scientific Computing, Department of Physics, Cavendish Laboratory, University of Cambridge (2002).
- [4] Lecture notes (part II) - Numerical Solution of Partial Differential Equations, Dr. N. Nikiforakis, Laboratory for Scientific Computing, Department of Physics, Cavendish Laboratory, University of Cambridge (2002).
- [5] Lecture notes (part III) - Introduction to Computational Fluid Dynamics , Dr. N. Nikiforakis, Department of Applied Mathematics and Theoretical Physics, University of Cambridge (2002).
- [6] Nikiforakis, Toro (2000): High-resolution global atmospheric transport modelling. Part 1: Suitability of Riemann-problem-based methods., Q.J.R. Meteorol. Soc., pp. 1-20.
- [7] Nikiforakis, Toro (2001): High-resolution global atmospheric transport modelling. Part 2: adaptive mesh refinement., Q.J.R. Meteorol. Soc.
- [8] Hubbard, Nikiforakis (2003): A Three-Dimensional, Adaptive, Godunov-Type Model for Global Atmospheric Flows, Monthly Weather Review, Vol. 131., pp. 1848-1864.
- [9] Ronchi, Iacono, Paolucci (1996): The "Cubed Sphere": A New Method for the Solution of partial Differential Equations in Spherical Geometry, , Journal of Computational Physics 124, Article No. 0047, pp. 93-114.
- [10] Berger, Colella (1989): Local Adaptive Mesh Refinement for Shock Hydrodynamics, Journal of Computational Physics 82, pp. 64-84.

- [11] Bell et al (1994): Three-Dimensional Adaptive Mesh Refinement For Hyperbolic Conservation Laws, SIAM J. SCI. Comput., Vol. 15, No. 1, pp. 127-138.
- [12] Nair, Cote, Staniforth (1999): Cascade interpolation for semi-Lagrangian advection over the sphere, Q.J.R. Meteorol. Soc., Vol. 125, pp. 1445-1468.
- [13] Scientific Computing - Skriptum, Sommersemester 2009, Michael Pürerer, Revisions: Georg Kresse.
- [14] Daniel J. Jacobs, Lecture on Models of Atmospheric Transport and Chemistry (2007), Havard University, <http://acmg.seas.harvard.edu/education/index.html> (access: 2011/04/01).