Journal of Computational Physics 230 (2011) 4955–4974

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Partially implicit peer methods for the compressible Euler equations

Stefan Jebens [a,*], Oswald Knoth [a], Rüdiger Weiner [b]

[a] Institute for Tropospheric Research, Permoserstrasse 15, 04318 Leipzig, Germany
[b] Martin Luther University Halle-Wittenberg, Institute for Mathematics, 06099 Halle, Germany

ARTICLE INFO

Article history:
Received 30 August 2010
Received in revised form 21 February 2011
Accepted 8 March 2011
Available online 12 March 2011

Keywords:
W-methods
Linearly implicit peer methods
Approximate Jacobian
Compressible Euler equations
Cut cells

ABSTRACT

When cut cells are used for the representation of orography in numerical weather prediction models this leads to very small cells. On one hand this results in very harsh time step restrictions for explicit methods due to the CFL criterion. On the other hand cut cells only appear in a small region of the domain. Therefore we consider a partially implicit method: In cut cells the Jacobian incorporates advection, diffusion and acoustics while in the full cells of the free atmosphere only the acoustic part is used, i.e. the method is linearly implicit in the cut cell regions and semi-explicit in the free regions and computes with time step sizes restricted only by the CFL condition in the free atmosphere. Furthermore we use a simplified Jacobian in the cut cell regions in order to save storage and gain computational efficiency. While the method retains the order independently of the Jacobian we present a linear stability theory which takes the effects of the simplifications of the Jacobian on stability into account. The presented method is as stable and accurate as the underlying split-explicit method but furthermore it can compute with cut cells with nearly no additional effort.

1. Introduction

Weather services use a massively parallel environment to forecast the weather. Because they are easy to parallelize explicit methods are very popular in numerical weather prediction models. In compressible models the highest-frequency modes like sound waves are often not the physical modes of interest. In order to avoid time step restrictions for explicit methods due to CFL criterion for those meteorologically irrelevant modes a widely used ansatz is operator splitting. The differential equation is split up into two parts where the slow part is integrated with an accurate explicit method and a time step size restricted by the CFL number of the low-frequency modes while for the integration of the high-frequency modes a simpler method is used that fulfils the CFL criterion for the high-frequency modes. This can be an implicit method or an explicit method which uses smaller time steps. That idea is implemented in several weather models, e.g. in the Weather Research and Forecasting Model WRF of the National Center for Atmospheric Research (NCAR) [18,19] and in the operational model COSMO provided by the German Weather Service DWD.

For the representation of orography terrain-following coordinates are established in operational weather models. Fig. 1 shows a mountain with terrain-following coordinates and with a Cartesian grid using cut cells. As can be seen terrain-following coordinates have the advantage that the cells nearly retain their size even if they are close to the mountain, i.e. an explicit method can be used. On the other hand numerical errors can be induced around steep slopes where the Jacobian of the coordinate transformation is almost singular. Furthermore the stratified atmosphere generates artificial forces due to the terrain-following grid: The horizontal pressure derivative in a vertically stratified atmosphere is zero, i.e. it is small in

* Corresponding author.
E-mail address: jebens@tropos.de (S. Jebens).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.03.015

practical applications. This behaviour is reproduced by a Cartesian grid but the lower layers of a terrain-following grid have a slope so that the numerical pressure derivative becomes artificially large as illustrated by Fig. 1. As proposed in [21] the condition that such numerical forces remain reasonably small for the operational model of the DWD is $\delta h < \delta z$ with δz being the layer thickness and δh the change of orography between one grid point and the next, i.e. the stratification should be weak and the terrain smooth. This condition is violated in most operational models, even for larger-scale operational hydrostatic applications. On fine meshes the model orography tends to be steeper than on coarser meshes. Therefore artificial circulations driven by numerical errors can be substantial [23]. In the course of time the spatial resolutions of numerical weather prediction models increase so that terrain will be resolved better in the future what will result in steeper gradients and the described problems with terrain-following coordinates will become more stringent. They can be reduced by introducing a spatially homogeneous reference profile and computing only with the deviation from this profile as this is also done in this paper. But for operational applications on large areas it will not be possible to choose a horizontally homogeneous atmospheric reference state in the way indicated. Therefore operational applications will always suffer from numerically generated artificial forces near mountains [21]. For these reasons the DWD developed a cut cell model called LM-Z besides its operational model. The Regional Atmospheric Modeling System RAMS [13] originally used terrain-following coordinates and a cut cells option was added later. The ocean–land–atmosphere model OLAM [26] which was developed from RAMS uses cut cells only. The All Scale Atmospheric Model ASAM [7] is another model which uses a Cartesian grid with cut cells instead of terrain-following coordinates.

The main problem when using cut cells is that arbitrary small cells can occur and therefore the maximum time step is restricted by the CFL condition for these small cells if explicit time integration methods are used. A similar problem occurs when computing on the sphere with a latitude-longitude grid because of the pole singularities. To avoid these problems we use peer methods which are a subclass of general linear methods. General linear methods were introduced in [2], good overviews can be found in [5,8]. Peer methods are a very comprehensive class of general linear methods, they include the common Runge–Kutta methods and linear multi-step methods. They can be interpreted as cyclic multi-stage multi-step methods which means that every of the stages of a peer method is a linear multi-step method. They have the same order in every stage so they have no order reduction even for very stiff systems. Furthermore the order of linearly implicit peer methods does not depend on what is used as Jacobian. Because of the generality of this class of methods they showed to be applicable in many different kinds of problems. They were successfully applied to the compressible Euler equations as split-explicit methods [9] and as linearly implicit methods [10]. While split-explicit methods are computationally very efficient when advancing the numerical solution from one time step to another linearly implicit methods can use large time step sizes even when cut cells are present. It turned out that the split-explicit peer method from [9] was more accurate than the linearly implicit peer method from [10]. The presented class of partially implicit peer methods in this paper combines the advantages of both, the computational efficiency and accuracy of split-explicit peer methods and the stability of linearly implicit peer methods.

The idea behind partially implicit methods is that the Jacobian is only used where it is necessary for stability, i.e. they are a mixture of split-explicit and linearly implicit methods. Because we want to compute with time steps that are restricted by the CFL condition in the free atmosphere we only need the full Jacobian in cut cells, i.e. in a very small part of the domain. In the full cells advection and diffusion are treated explicitly, only the acoustic part is integrated implicitly. Therefore partially implicit methods are nearly as efficient as split-explicit methods which use a one-step implicit integrator for the acoustics. In cut cells we will use a Jacobian which uses a lower-order spatial discretization of advection in order to save storage and gain computational efficiency. As mentioned before these two simplifications of the Jacobian, ignoring the advection and diffusion parts of the Jacobian in the free atmosphere and using a lower-order approximation of advection in cut cells, have no influence on the order of the peer method but they require a careful consideration of linear stability theory in order to derive a method which is stable despite of these simplifications.

This paper is organized as follows: In Section 2 we formulate the class of linearly implicit peer methods and derive order conditions.

In Section 3 we consider the linear stability theory and show the amplitude and phase properties of the presented partially implicit peer method, the split-explicit peer method and the Rosenbrock method ROS3Pw for comparison.

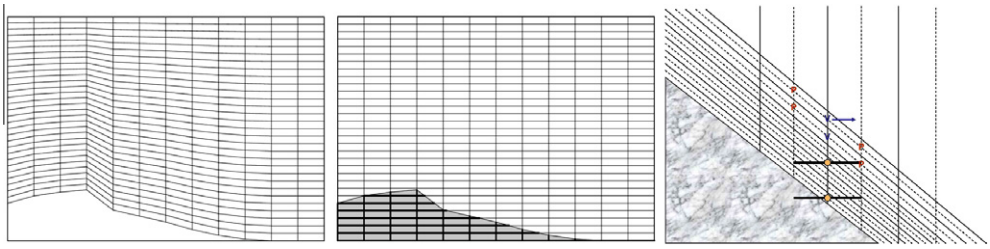

Fig. 1. Orography modelled with terrain-following coordinates (left) and cut cells (middle). Pressure in a model with terrain-following coordinates (right). Images from [25].

Four numerical tests with the compressible Euler equations are presented in Section 4 in order to illustrate the effects of the simplifications of the Jacobian.

Finally we give some conclusions in Section 5.

2. Formulation of the methods

Explicit two-step peer methods for first-order problems were considered in [27,17,29]. Linearly implicit peer methods were discussed in [28,14,4]. In [9] split-explicit methods for the compressible Euler equations were considered where the advection part was integrated with an explicit peer method. In [10] a linearly implicit peer method with an inexact Jacobian was used for the integration of the compressible Euler equations. We use the same representation of linearly implicit peer methods for the solution of autonomous first-order differential equations

$$\dot{y} = f(y), \quad t \in [t_0, t_e], \quad y(t_0) = y_0 \in \mathbb{R}^n \tag{1}$$

as in [10] with the exception that we also incorporate linear combinations of the numerical solutions from the actual time step, i.e. we consider

$$(I - \Delta t\gamma J)Y_{mi} = \sum_{j=1}^{s} b_{ij}Y_{m-1,j} + \sum_{j=1}^{i-1} s_{ij}Y_{mj} + \Delta t\sum_{j=1}^{s} a_{ij}f(Y_{m-1,j}) + \Delta t\sum_{j=1}^{i-1} r_{ij}f(Y_{mj}) + \Delta t\sum_{j=1}^{s} g_{ij}JY_{m-1,j} + \Delta t\sum_{j=1}^{i-1} h_{ij}JY_{mj},$$

where s is the number of stages, Y_{mi} is the approximation to the analytical solution at time $t_{mi} := t_m + c_i\Delta t$ and J is an approximation to the Jacobian of f at time t_m. Δt is the time step size which is constant in the remainder of this paper as this is the case in many weather codes. In compact notation the considered methods read

$$Y_m(I - \Delta t\gamma J)^T = BY_{m-1} + SY_m + \Delta tAF_{m-1} + \Delta tRF_m + \Delta tGY_{m-1}J^T + \Delta tHY_mJ^T \tag{2}$$

with

$$Y_m := (Y_{mi})_{i=1}^s \in \mathbb{R}^{s\times n},$$
$$F_m := (f(Y_{mi}))_{i=1}^s \in \mathbb{R}^{s\times n}$$

and $B,\ S,\ A,\ R,\ G,\ H \in \mathbb{R}^{s\times s}$ where $S,\ R$ and H are strictly lower triangular matrices. So in each time step we have to solve s linear systems of equations each with the same matrix $I - \Delta t\gamma J$. While we use formulation (2) for the derivation of order conditions and stability theory the methods are implemented in the equivalent formulation

$$(I - \Delta t\gamma J)\widehat{Y}_{mi} = \sum_{j=1}^{s}\left(b_{ij} + \frac{g_{ij}}{\gamma}\right)Y_{m-1,j} + \sum_{j=1}^{i-1}\left(s_{ij} + \frac{h_{ij}}{\gamma}\right)Y_{mj} + \Delta t\sum_{j=1}^{s} a_{ij}f(Y_{m-1,j}) + \Delta t\sum_{j=1}^{i-1} r_{ij}f(Y_{mj})$$
$$Y_{mi} = \widehat{Y}_{mi} - \sum_{j=1}^{s}\frac{g_{ij}}{\gamma}Y_{m-1,j} - \sum_{j=1}^{i-1}\frac{h_{ij}}{\gamma}Y_{mj}$$

in order to avoid the matrix–vector multiplications with the Jacobian in (2).

We now briefly present order conditions for the considered class of methods. Thereby we make no assumptions for J. Consistency of the peer methods is discussed by considering the local residuals $\triangle_{mi}$ obtained by substituting the exact solution y into the method:

$$\triangle_{mi} := (I - \Delta t\gamma J)y(t_{mi}) - \sum_{j=1}^{s} b_{ij}y(t_{m-1,j}) - \sum_{j=1}^{i-1} s_{ij}y(t_{mj}) - \Delta t\sum_{j=1}^{s} a_{ij}\dot{y}(t_{m-1,j}) - \Delta t\sum_{j=1}^{i-1} r_{ij}\dot{y}(t_{mj})$$
$$- \Delta t\sum_{j=1}^{s} g_{ij}Jy(t_{m-1,j}) - \Delta t\sum_{j=1}^{i-1} h_{ij}Jy(t_{mj}).$$

Definition 1. A peer method has order of consistency p, if

$$\max_i ||\triangle_{mi}|| = \mathcal{O}((\Delta t)^{p+1}).$$

Taylor expansion at t_m leads to the following theorem.

Theorem 1. *A peer method has order of consistency p, if*

$$AB(k) = 0, \quad k = 0, 1, \ldots, p,$$

$$\widehat{AB}(k) = 0, \quad k = 1, \ldots, p,$$

where

$$AB_i(k) := c_i^k - \sum_{j=1}^{s} b_{ij}(c_j-1)^k - \sum_{j=1}^{i-1} s_{ij}c_j^k - k\sum_{j=1}^{s} a_{ij}(c_j-1)^{k-1} - k\sum_{j=1}^{i-1} r_{ij}c_j^{k-1},$$

$$\widehat{AB}_i(k+1) := \gamma c_i^k + \sum_{j=1}^{s} g_{ij}(c_j-1)^k + \sum_{j=1}^{i-1} h_{ij}c_j^k \quad \text{for } i=1,\dots,s.$$

The order conditions $AB(k)=0$ are the conditions of the explicit part of the peer method, i.e. only these conditions remain in the case of $J=0$. The order conditions $\widehat{AB}(k)=0$ remain when subtracting $AB(k)$ from the Taylor expansion, i.e. the order conditions decouple into summands with and without J. This is the reason why the peer methods have the order independently of J. Because we consider methods with constant time step sizes convergence follows from consistency and zero stability:

Definition 2. A peer method is called zero stable, if the eigenvalues λ of $(I-S)^{-1}B$ satisfy the condition

$$|\lambda| \leqslant 1 \quad \text{and} \quad \{|\lambda| = 1 \Rightarrow \lambda \text{ is simple}\}.$$

Definition 3. A linearly implicit peer method has order of convergence p, if the global error $\varepsilon_m := Y(t_m) - Y_m$, where $Y(t_m)$ is a matrix containing the analytical solution y at times t_{mi}, has order p, i.e. if there is a constant C with

$$||Y(t_m) - Y_m|| \leqslant C(\Delta t)^p.$$

Theorem 2. *Let the following conditions for the peer method be satisfied.*

- *The peer method has order of consistency p.*
- *The peer method is zero stable.*
- *The starting values have order p, i.e.* $y(t_{0i}) - Y_{0i} = \mathcal{O}((\Delta t)^p)$ *for* $i=1,\dots,s$.

Then this peer method has order of convergence p.

The collected order conditions $AB(k)=0$ and $\widehat{AB}(k)=0$ for order $p=s-1$ can be written in compact matrix form. Therefore we introduce the diagonal matrices $C=\mathrm{diag}(c_1,\dots,c_s)$, $\Gamma=\gamma I$, $D=\mathrm{diag}(1,2,\dots,s)$ and the square matrices

$$V_0 = \begin{pmatrix} 1 & c_1 & \cdots & c_1^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s & \cdots & c_s^{s-1} \end{pmatrix} \quad \text{and} \quad V_1 = \begin{pmatrix} 1 & c_1-1 & \cdots & (c_1-1)^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s-1 & \cdots & (c_s-1)^{s-1} \end{pmatrix}.$$

The first condition $AB(0)=0$ simply is

$$(B+S)\mathbb{1} = \mathbb{1},$$

where $\mathbb{1}=(1,\dots,1)^T$. The other conditions $1 \leqslant k \leqslant s-1$ lead to the matrix equations

$$CV_0 - B(C-I)V_1 - SCV_0 - AV_1D - RV_0D - \beta e_s^T D = 0,$$

$$\Gamma V_0 + GV_1 + HV_0 + \widetilde{\beta} e_s^T = 0,$$

where e_s is the sth unit vector and β and $\widetilde{\beta}$ represent the remaining degrees of freedom. Because V_1 is a Vandermonde matrix it is regular if the nodes c_i distinguish from each other. Therefore a peer method with

$$(B+S)\mathbb{1} = \mathbb{1},$$
$$A = \left(CV_0 - B(C-I)V_1 - SCV_0 - RV_0D - \beta e_s^T D\right)D^{-1}V_1^{-1},$$
$$G = -\left(\Gamma V_0 + HV_0 + \widetilde{\beta} e_s^T\right)V_1^{-1}$$

has order of convergence $p=s-1$ independently of J if $B+S$ satisfies the zero stability condition and if the starting values are sufficiently accurate. In the remainder we will concentrate on second-order methods with $s=3$ stages.

In [9] a second-order three-stage split-explicit peer method was used for the integration of the compressible Euler equations. That method is quite accurate and has good stability properties. Therefore we want to use it for the explicit part of the linearly implicit peer method. Applying the split-explicit peer method from [9] to the split-differential equation

$$\dot{y} = N(y) + Ly,$$

where N represents the nonlinear advection part and L the linearized acoustic part results in the method rule

$$Z_{mi}(0) = \sum_{j=1}^{s} b_{ij}Y_{m-1,j} + \sum_{j=1}^{i-1} s_{ij}Y_{mj},$$

$$\dot{Z}_{mi}(t) = \frac{1}{\alpha_i}\left(\sum_{j=1}^{s} a_{ij} N(Y_{m-1,j}) + \sum_{j=1}^{i-1} r_{ij} N(Y_{mj})\right) + L Z_{mi}(t),$$

$$Y_{mi} = Z_{mi}(\alpha_i \Delta t).$$

Solving the occurring differential equation with the trapezoidal rule leads to the equation

$$\left(I - \frac{\alpha_i \Delta t}{2} L\right) Y_{mi} = \sum_{j=1}^{s} b_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} s_{ij} Y_{mj} + \Delta t \sum_{j=1}^{s} a_{ij} f(Y_{m-1,j}) + \Delta t \sum_{j=1}^{i-1} r_{ij} f(Y_{mj}), \tag{3}$$

$$+ \Delta t \sum_{j=1}^{s} \left(\frac{\alpha_i}{2} b_{ij} - a_{ij}\right) L Y_{m-1,j} + \Delta t \sum_{j=1}^{i-1} \left(\frac{\alpha_i}{2} s_{ij} - r_{ij}\right) L Y_{mj}, \tag{4}$$

where $f(y) = N(y) + Ly$, i.e. the split-explicit peer method with the trapezoidal rule as integrator for the fast differential equation is equivalent to a non-singly linearly implicit peer method with $J = L,\ \gamma_i = \frac{\alpha_i}{2},\ g_{ij} = \frac{\alpha_i}{2} b_{ij} - a_{ij}$ and $h_{ij} = \frac{\alpha_i}{2} s_{ij} - r_{ij}$. As shown in [9] this method is stable up to advection CFL numbers of 1.7 and arbitrary large sound CFL numbers despite of the fact that only the Jacobian which belongs to the acoustics is used.

For the linearly implicit peer method we want to use only the acoustic part of the Jacobian in the free atmosphere. Therefore we want to derive a method which also has the same stability property as the split-explicit method. Furthermore the linearly implicit peer method should be stable for arbitrary large advection CFL numbers if the Jacobian also incorporates the advection part because this property is needed when cut cells occur. The method (3) does not have this property and is not singly implicit. So we use the explicit parts $c_i,\ b_{ij},\ s_{ij},\ a_{ij}$ and r_{ij} of (3) and optimize the remaining implicit parts $\gamma, \widetilde{\beta}_1,\ \widetilde{\beta}_2,\ \widetilde{\beta}_3,\ h_{21},\ h_{31}$ and h_{32} in order to obtain a second-order peer method which retains the first stability property in the full cells and furthermore is stable for arbitrary large advection CFL numbers if the full Jacobian is used in the cut cells.

3. Stability investigations

For the investigation of stability properties we start with the one-dimensional compressible Euler equations in conservative form. Because some of the numerical tests will use diffusion we also include a diffusion term with constant diffusion coefficient ν:

$$\dot{\rho} = -\frac{\partial \rho u}{\partial x},$$

$$\dot{(\rho u)} = -\frac{\partial \rho u u}{\partial x} - \frac{\partial p}{\partial x} + \nu \rho \frac{\partial^2 u}{\partial x^2},$$

$$\dot{(\rho \theta)} = -\frac{\partial \rho u \theta}{\partial x}.$$

Here ρ is the density, u is the wind speed, θ the potential temperature and p the pressure. The prognostic variables are $\rho,\ \rho u$ and $\rho\theta$. The pressure p is given diagnostically by the equation of state

$$p = \left(\frac{R\rho\theta}{p_0^{\kappa}}\right)^{\frac{1}{1-\kappa}},$$

where R is the gas constant for dry air, $\kappa = R/c_p,\ c_p$ the heat capacity of dry air at constant pressure and p_0 is the pressure at ground. At first we replace the pressure from the momentum equation with the chain rule

$$\frac{\partial p}{\partial x} = \frac{\partial p}{\partial \rho\theta} \frac{\partial \rho\theta}{\partial x}.$$

It holds

$$\frac{\partial p}{\partial \rho\theta} = \frac{R}{p_0^{\kappa}(1-\kappa)} \left(\frac{R\rho\theta}{p_0^{\kappa}}\right)^{\frac{\kappa}{1-\kappa}} = \frac{1}{\rho\theta(1-\kappa)} \left(\frac{R\rho\theta}{p_0^{\kappa}}\right)^{\frac{1}{1-\kappa}} = \frac{c_s^2}{\theta},$$

where

$$c_s := \sqrt{\frac{1}{\rho(1-\kappa)} \left(\frac{R\rho\theta}{p_0^{\kappa}}\right)^{\frac{1}{1-\kappa}}}$$

is the speed of sound. For the other nonlinear terms we use the product rule:

$$\frac{\partial \rho u u}{\partial x} = -u^2 \frac{\partial \rho}{\partial x} + 2u \frac{\partial \rho u}{\partial x},$$

$$\frac{\partial \rho u \theta}{\partial x} = -u\theta \frac{\partial \rho}{\partial x} + \theta \frac{\partial \rho u}{\partial x} + u \frac{\partial \rho\theta}{\partial x},$$

$$\rho\frac{\partial^2 u}{\partial x^2}=\frac{\partial^2 \rho u}{\partial x^2}-u\frac{\partial^2 \rho}{\partial x^2}-\frac{2}{\rho}\frac{\partial \rho}{\partial x}\frac{\partial \rho u}{\partial x}+\frac{2u}{\rho}\frac{\partial \rho}{\partial x}\frac{\partial \rho}{\partial x}.$$

We now linearize these equations by subtracting some time- and space-independent background state (denoted by a bar, e.g. $\rho' := \rho - \overline{\rho}$) and dropping all nonlinear terms,i.e. products of two disturbed quantities. The speed of sound is assumed to be constant 340 ms^{-1}. We derive the linearized equations for the disturbed quantities in compact matrix form

$$\begin{pmatrix}\dot{\rho}'\\ (\dot{\rho u})'\\ \frac{1}{\overline{\theta}}(\dot{\rho\theta})'\end{pmatrix}=-\begin{pmatrix}0 & 1 & 0\\ -\overline{u}^2 & 2\overline{u} & c_s^2\\ -\overline{u} & 1 & \overline{u}\end{pmatrix}\begin{pmatrix}\rho'_x\\ (\rho u)'_x\\ \frac{1}{\overline{\theta}}(\rho\theta)'_x\end{pmatrix}+\nu\begin{pmatrix}0 & 0 & 0\\ -\overline{u} & 1 & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\rho'_{xx}\\ (\rho u)'_{xx}\\ \frac{1}{\overline{\theta}}(\rho\theta)'_{xx}\end{pmatrix}, \tag{5}$$

where we divided the potential temperature by the background potential temperature for simplicity of notation.

In order to save storage and gain computational efficiency we make three simplifications for the Jacobian J:

- We use the conservative form of the Euler equations, i.e. equations with prognostic variables ρu and $\rho\theta$, as right-hand side because this guarantees conservation properties, i.e. conservation of mass, momentum and entropy. Contrariwise we use the Jacobian which comes from the advection form of the Euler equations, i.e. instead of an equation for ρu there is an equation for u.
- The transport terms in the Euler equations are discretized with the third-order upwind scheme described in [30]. For the Jacobian we use the first-order upwind scheme.
- We will use the full but simplified Jacobian in cut cells only, in the free atmosphere the Jacobian only contains the acoustic part, i.e. advection and diffusion are treated explicitly. In the remainder of this paper we will call it partial Jacobian.

We now discuss the effects of these simplifications for the stability (remember that they have no influence on the order of the peer methods). If we do not use $(\rho u)'$ but $\overline{\rho}u' = (\rho u)' - \overline{u}\rho' - \rho' u'$ as prognostic variable and drop the product of the two disturbed quantities $\rho' u'$, i.e. assume $\overline{\rho}u' = (\rho u)' - \overline{u}\rho'$, we obtain the simpler equations

$$\begin{pmatrix}\dot{\rho}'\\ (\overline{\rho}\dot{u}')\\ \frac{1}{\overline{\theta}}(\rho\dot{\theta})'\end{pmatrix}=-\begin{pmatrix}\overline{u} & 1 & 0\\ 0 & \overline{u} & c_s^2\\ 0 & 1 & \overline{u}\end{pmatrix}\begin{pmatrix}\rho'_x\\ (\overline{\rho}u')_x\\ \frac{1}{\overline{\theta}}(\rho\theta)'_x\end{pmatrix}+\nu\begin{pmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\rho'_{xx}\\ (\overline{\rho}u')_{xx}\\ \frac{1}{\overline{\theta}}(\rho\theta)'_{xx}\end{pmatrix}. \tag{6}$$

This set of equations (6) is simpler than (5) with respect to two aspects: It has less terms than (5) which results in computational efficiency when solving the resulting linear system of equations. Table 1 in the concluding section shows the storage saving also in 2D and 3D. Furthermore it is easier to implement because there is are no ρ derivatives in the momentum and entropy equations and it is not obvious how to implement these derivatives on a staggered grid. Contrariwise the additional ρ derivative in the mass equation of (6) can be treated in the same manner as the temperature derivative in the entropy equation and therefore produces no difficulties. Because of these reasons we will use (5) as right-hand side but the derivative of (6) as Jacobian despite that it comes from the advection form of the Euler equations.

We will now perform a von Neumann stability analysis to investigate the influence of the second simplification of the Jacobian. We use a staggered grid, i.e. the density and the potential temperature are cell-centered while the wind is defined on the faces. Four different spatial discretization schemes are used: The first-order derivatives without the wind speed $\overline{u}$ involved, i.e. the u derivatives in the mass and entropy equations and the θ derivative in the momentum equation, belong to the acoustic part of the compressible equations and are therefore discretized with central differences, e.g.

$$\begin{aligned}&u(t,x_j)=u(t)e^{ikx_j},\\ &\Rightarrow\quad \left.\frac{\partial u}{\partial x}\right|_{(t,x_j)}\approx u(t)\frac{e^{ikx_j}}{\Delta x}\left(e^{\frac{ik\Delta x}{2}}-e^{-\frac{ik\Delta x}{2}}\right),\end{aligned}$$

where Δx is the spatial step size and k is the wave number. The other first-order terms are discretized with the third-order upwind scheme for the computation of the right-hand side or with the first-order upwind scheme for the simplified Jacobian. The second-order derivative which originates from the diffusion term are discretized with second-order central differences. The four operators (for positive wind speeds) are:

$$\mathcal{D}_1=\frac{1}{\Delta x}\left(1-e^{-ik\Delta x}\right),$$

Table 1
Numbers of entries per grid cell in the different Jacobians.

	Exact Jacobian	Simplified Jacobian	Ratio	Partial Jacobian	Ratio
1D	$3\mathcal{D}_2+4\mathcal{D}_3=22$	$3\mathcal{D}_2+3\mathcal{D}_1=12$	55%	$3\mathcal{D}_2=6$	50%
2D	$6\mathcal{D}_2+14\mathcal{D}_3=68$	$6\mathcal{D}_2+8\mathcal{D}_1=28$	41%	$6\mathcal{D}_2=12$	43%
3D	$9\mathcal{D}_2+30\mathcal{D}_3=138$	$9\mathcal{D}_2+15\mathcal{D}_1=48$	35%	$9\mathcal{D}_2=18$	38%

$$\mathcal{D}_2 = \frac{1}{\Delta x}\left(e^{\frac{ik\Delta x}{2}} - e^{-\frac{ik\Delta x}{2}}\right),$$
$$\mathcal{D}_3 = \frac{1}{6\Delta x}\left(2e^{ik\Delta x} + 3 - 6e^{-ik\Delta x} + e^{-2ik\Delta x}\right),$$
$$\mathcal{D}_v = \frac{1}{(\Delta x)^2}\left(e^{ik\Delta x} - 2 + e^{-ik\Delta x}\right).$$

Using these operators as approximations to the spatial derivatives results in the ordinary differential equation

$$\begin{pmatrix} \dot{\rho}' \\ (\dot{\rho u})' \\ \frac{1}{\theta}(\rho\dot{\theta})' \end{pmatrix} = -\begin{pmatrix} 0 & \mathcal{D}_2 & 0 \\ -\overline{u}^2\mathcal{D}_3 + v\overline{u}\mathcal{D}_v & 2\overline{u}\mathcal{D}_3 - v\mathcal{D}_v & c_s^2\mathcal{D}_2 \\ -\overline{u}\mathcal{D}_3 & \mathcal{D}_2 & \overline{u}\mathcal{D}_3 \end{pmatrix}\begin{pmatrix} \rho' \\ (\rho u)' \\ \frac{1}{\theta}(\rho\theta)' \end{pmatrix}. \tag{7}$$

For the Jacobian we instead use the matrix which belongs to the system

$$\begin{pmatrix} \dot{\rho}' \\ (\overline{\rho}\dot{u}') \\ \frac{1}{\theta}(\rho\dot{\theta})' \end{pmatrix} = -\begin{pmatrix} \overline{u}\mathcal{D}_1 & \mathcal{D}_2 & 0 \\ 0 & \overline{u}\mathcal{D}_1 - v\mathcal{D}_v & c_s^2\mathcal{D}_2 \\ 0 & \mathcal{D}_2 & \overline{u}\mathcal{D}_1 \end{pmatrix}\begin{pmatrix} \rho' \\ \overline{\rho}u' \\ \frac{1}{\theta}(\rho\theta)' \end{pmatrix}. \tag{8}$$

The third simplification for the Jacobian is used in the regions without cut cells. There the Jacobian includes the acoustic part only, i.e. no advection and diffusion. The corresponding system is

$$\begin{pmatrix} \dot{\rho}' \\ (\overline{\rho}\dot{u}') \\ \frac{1}{\theta}(\rho\dot{\theta})' \end{pmatrix} = -\begin{pmatrix} 0 & \mathcal{D}_2 & 0 \\ 0 & 0 & c_s^2\mathcal{D}_2 \\ 0 & \mathcal{D}_2 & 0 \end{pmatrix}\begin{pmatrix} \rho' \\ \overline{\rho}u' \\ \frac{1}{\theta}(\rho\theta)' \end{pmatrix}. \tag{9}$$

We now compare the eigenvalues of the different discretizations. We denote M the matrix of system (7) which is used for the right-hand side. For cut cells the matrix $\widetilde{M}$ of the corresponding system (8) is used as Jacobian and in the free atmosphere we use the discretization (9) with its matrix named $\widehat{M}$. While the eigenvalues of the matrices from (5) and (6) are equal if there is no diffusion, namely $-\overline{u}$ and $-\overline{u} \pm c_s$, for $v = 0$ the eigenvalues of M are

$$-\overline{u}\mathcal{D}_3 \quad \text{and} \quad -\overline{u}\mathcal{D}_3 \pm \sqrt{c_s^2\mathcal{D}_2^2 + \overline{u}^2\mathcal{D}_3^2 - \overline{u}^2\mathcal{D}_2\mathcal{D}_3},$$

whereas the eigenvalues of $\widetilde{M}$ are

$$-\overline{u}\mathcal{D}_1 \quad \text{and} \quad -\overline{u}\mathcal{D}_1 \pm c_s\mathcal{D}_2$$

and $\widehat{M}$ has the eigenvalues

$$0 \quad \text{and} \quad \pm c_s\mathcal{D}_2.$$

Fig. 2 shows the eigenvalues of M and $\widetilde{M}$. The simplifications of the Jacobian have no significant influence on the imaginary part because it is dominated by the acoustics which is treated equally in (7)–(9). In contrast the simplified Jacobian has eigenvalues with real parts which are up to 2.5 times larger than the real parts of the eigenvalues of the exact Jacobian. Because the acoustic part is discretized with central differences the eigenvalues of $\widehat{M}$ are purely imaginary. These effects of the simplifications have to be taken in account when searching for stable methods.

In order to obtain a good second-order three-stage peer method we use an algorithm which optimizes the remaining degrees of freedom with respect to amplitude and phase errors. As side condition we claim that the method has to be stable for wind speeds $\overline{u} < c_s/6 \approx 56\ \mathrm{m\ s^{-1}} \approx 200\ \mathrm{km\ h^{-1}}$ in the following situations:

- In cut cells the matrix $\widetilde{M}$ is used as Jacobian, the peer method should be stable for arbitrary large advection CFL numbers $C_{adv} := \overline{u}\frac{\Delta t}{\Delta x}$ and acoustic CFL numbers $C_{sound} := c_s\frac{\Delta t}{\Delta x}$. Because stability is determined numerically this condition is verified for CFL numbers up to 10000.
- In the free atmosphere only the acoustic part $\widehat{M}$ is used as Jacobian. The peer method should be as stable as the underlying split-explicit peer method from [9], i.e. it should be stable for $C_{adv} < 1.7$ and arbitrary large sound CFL numbers, in practice up to 10000.

To verify these aims we apply the peer method (2) to the linear test Eq. (7). Depending on whether we use $\widetilde{M}$ respectively $\widehat{M}$ as Jacobian we obtain the equations

$$Y_m(I - \Delta t\gamma\widetilde{M})^T = BY_{m-1} + SY_m + \Delta tAY_{m-1}M^T + \Delta tRY_mM^T + \Delta tGY_{m-1}\widetilde{M}^T + \Delta tHY_m\widetilde{M}^T,$$

respectively

$$Y_m(I - \Delta t\gamma\widehat{M})^T = BY_{m-1} + SY_m + \Delta tAY_{m-1}M^T + \Delta tRY_mM^T + \Delta tGY_{m-1}\widehat{M}^T + \Delta tHY_m\widehat{M}^T.$$

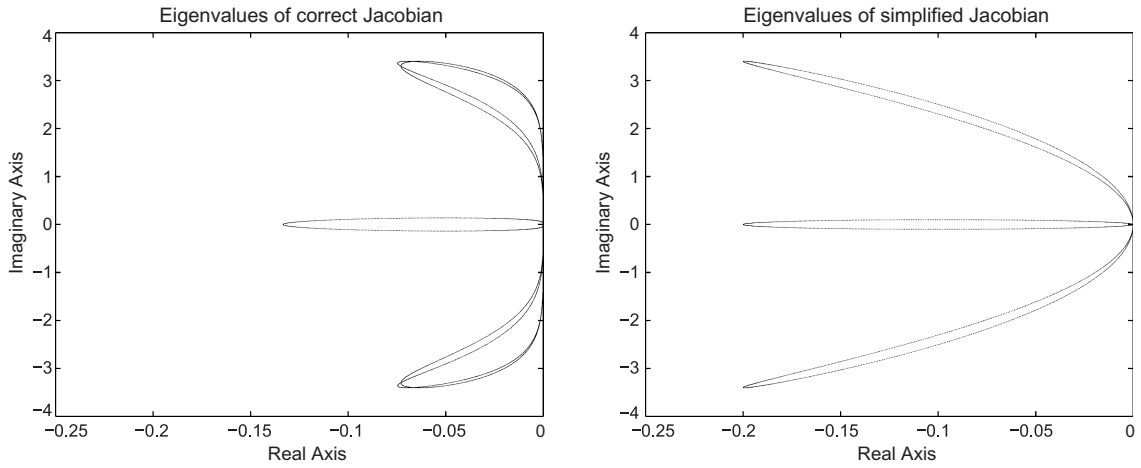

Fig. 2. Eigenvalues of the exact Jacobian M (left) and the simplified Jacobian $\widetilde{M}$ (right).

If we solve these equations for Y_m we obtain the systems $Y_m = \widetilde{A}Y_{m-1}$ respectively $Y_m = \widehat{A}Y_{m-1}$. $\widetilde{A}$ and $\widehat{A}$ are called the amplification matrices and are 9×9 matrices because (7) is a 3×3 system and we consider peer methods with $s = 3$ stages. They depend on the CFL numbers C_{adv} and C_{sound}, the wave number k and the diffusion coefficient ν. Fig. 3 shows the stability diagrams for the peer method presented in this paper. On the horizontal axis the sound CFL number C_{sound} is plotted, the vertical coordinate is the advection CFL number C_{adv}. The diagrams show the maxima of the moduli of the eigenvalues of $\widetilde{A}$ respectively $\widehat{A}$ for $\nu = 0$ and wavelengths $2\pi/k$ between $2\Delta x$ and $20\Delta x$. As shown the presented peer method satisfies the desired stability criteria. Furthermore the method is stable for $C_{adv} = C_{sound} = 0$, i.e. zero stability is guaranteed because for $C_{adv} = C_{sound} = 0$ it holds $\widetilde{A} = \widehat{A} = (I - S)^{-1}B$.

To find that method we used a simple genetic algorithm: A set of 1000 peer methods is generated from the degrees of freedom $\gamma,\ \widetilde{\beta}_1,\ \widetilde{\beta}_2,\ \widetilde{\beta}_3,\ h_{21},\ h_{31}$ and h_{32}. These methods are evaluated with the optimization function and from the coefficients of the best method 999 new methods are randomly generated in a neighbourhood within a radius which decreases in each iteration step. If the best method is not changing during ten steps the algorithm aborts. The optimization function consists of two parts: The first aim is stability as explained above. If the stability criteria are fulfilled the inverse of the maximum of the amplitude and phase errors is the optimization function value. The stability criteria are verified with the functions

$$\widetilde{\varphi}(C_{sound}) := \begin{pmatrix} ||\mathrm{eig}(\widetilde{A}(0, C_{sound}))||_\infty \\ ||\mathrm{eig}(\widetilde{A}(C_{sound}/6, C_{sound}))||_\infty \\ ||\mathrm{eig}(\widehat{A}(0, C_{sound}))||_\infty \\ ||\mathrm{eig}(\widehat{A}(\min(C_{sound}/6, 1.7), C_{sound}))||_\infty \end{pmatrix}$$

and

$$\widehat{\varphi} := \max\{0 \leqslant C_{sound} \leqslant 10000 : ||\widetilde{\varphi}(C_{sound})||_\infty \leqslant 1\} - 10000,$$

where eig denotes the eigenvalue function and the CFL numbers C_{adv} and C_{sound} are the arguments of the amplification matrices $\widetilde{A}$ and $\widehat{A}$. Then our optimization function φ is given by

$$\varphi := \begin{cases} \widehat{\varphi} & \text{if } \widehat{\varphi} < 0, \\ 1/\max\{error_{amp}, error_{phase}\} & \text{if } \widehat{\varphi} = 0 \end{cases}$$

and we are optimizing with respect to the global maximum of φ. Amplitude and phase are defined as follows: The analytical solution of

$$\dot{y} = My \quad \text{is } y(t_m) = \exp(\Delta t M) y(t_{m-1}),$$

while the peer method satisfies

$$Y_m = \widetilde{A}Y_{m-1} \quad \text{respectively } Y_m = \widehat{A}Y_{m-1}.$$

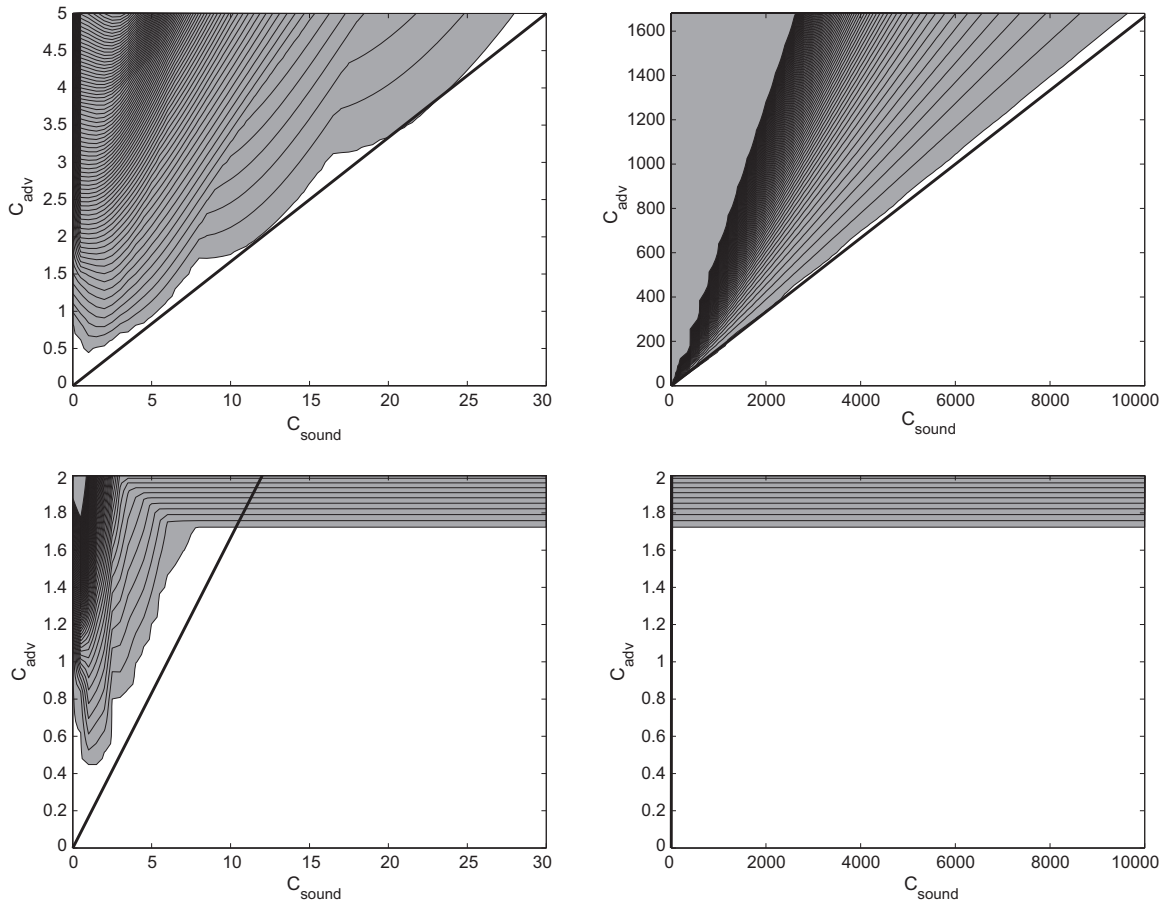

Fig. 3. Stability diagrams for the presented peer method with Jacobian $\widetilde{M}$ (top) and partial Jacobian $\widehat{M}$ (bottom). Unstable regions in grey with contour interval 0.1.

M has one eigenvalue which results from the advection part and two eigenvalues incorporating advection and the sound wave with opposing directions. Let λ be the eigenvalue of $\exp(\Delta t M)$ which originates from the pure advection part and let $\widetilde{\lambda}$ and $\widehat{\lambda}$ be the eigenvalues from the amplification matrices $\widetilde{A}$ and $\widehat{A}$ which correspond to λ. Then $|\lambda|$ is the amplification factor and $\frac{\arctan\frac{\mathrm{Im}\lambda}{\mathrm{Re}\lambda}}{\mathrm{Im}\lambda}$ is the relative phase speed. These values are compared with the corresponding values from the amplification matrices of the peer method and the differences are the amplification and phase errors which are part of the optimization function. Because we use the split-explicit peer method as underlying method for all generated implicit peer methods and consider only the amplitude and phase properties of the advection part the implicit peer methods all have the same amplitude and phase errors in the case where the Jacobian incorporates the acoustics only, i.e. for $\widehat{A}$ the implicit peer methods adopt the good amplitude and phase properties of the split-explicit peer method and we only have to optimize the errors of $\widetilde{A}$.

Fig. 4 shows the amplitude and relative phase for the $4\Delta x$ wave, i.e. $k = 2\pi/(4\Delta x)$, of the peer method presented in this paper. Its coefficients are given in the appendix. We compare it with the underlying split-explicit peer method and ROS3Pw which is presented in [15]. Popular Rosenbrock methods like RODAS [6], RODASP [20] or ROS3P [12] are ROW-methods, i.e. the order decreases to 1 and they are not stable if the partial Jacobian is used. Contrariwise ROS3Pw is a W-method and is second-order if an inexact Jacobian is used. Furthermore it is stable even if only the acoustic part is incorporated in the Jacobian but the CFL condition is $C_{adv} < 0.9$ while the peer method is stable up to advection CFL numbers of 1.7. The peer methods have much better amplitude and phase properties than ROS3Pw. Fig. 4 also shows that ROS3Pw becomes unstable for smaller CFL numbers than the peer methods if $\widehat{M}$ is used as Jacobian. The differences between the properties of the methods decrease if waves with larger wavelengths are considered because for them the eigenvalues are closer to 0.

4. Numerical tests

So far we presented a second-order linearly implicit three-stage peer method with its properties. Now we want to test it in a more practical environment. We consider the two-dimensional dry compressible Euler equations in conservative form with diffusion:

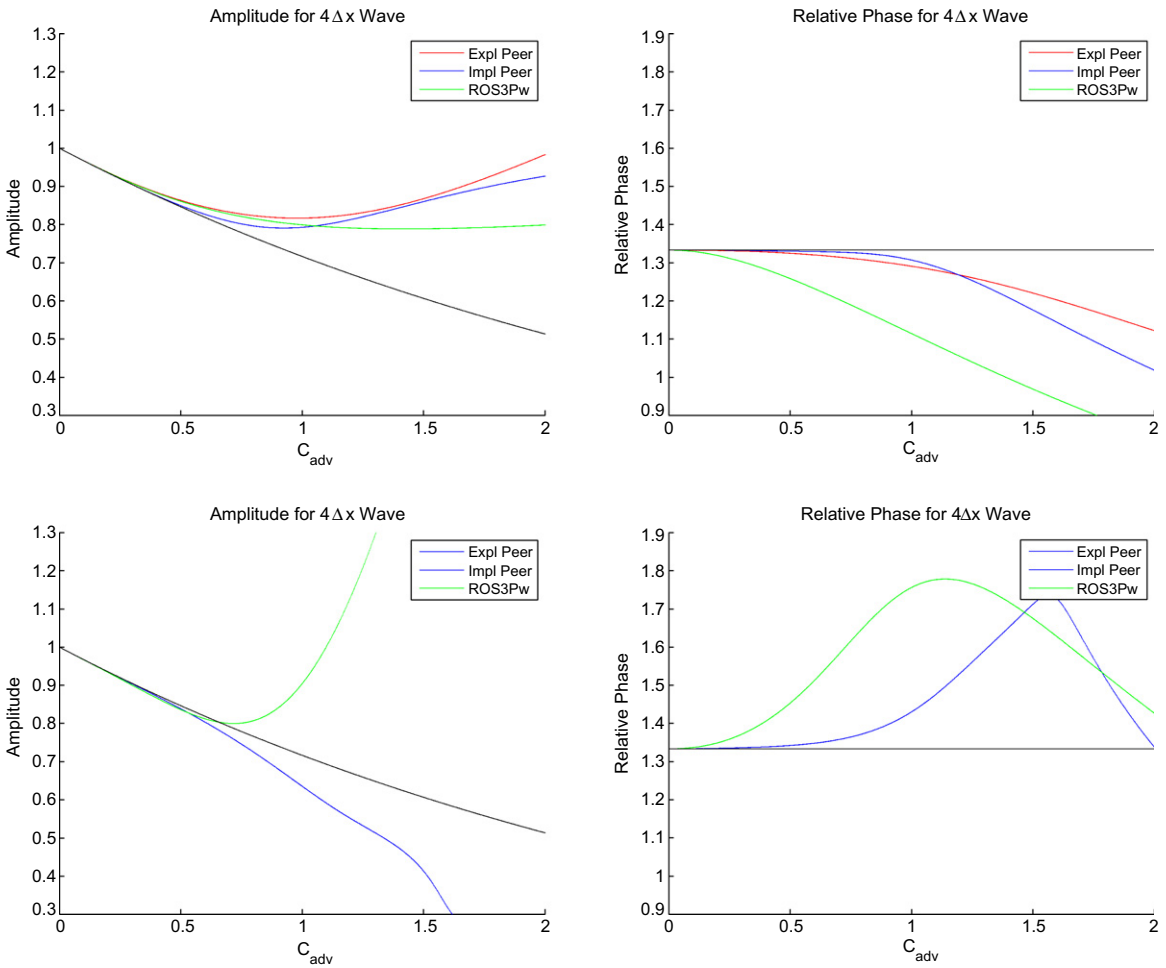

Fig. 4. Amplitude (left) and relative phase (right) for the $4\Delta x$ wave with Jacobian $\widetilde{M}$ (top) and partial Jacobian $\widehat{M}$ (bottom). The analytic amplitude and relative phase in black.

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial z},$$
$$\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho uu}{\partial x} - \frac{\partial \rho wu}{\partial z} - \frac{R}{1-\kappa}\pi\frac{\partial \rho\theta}{\partial x} + \nu\rho\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2}\right),$$
$$\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho uw}{\partial x} - \frac{\partial \rho ww}{\partial z} - \frac{R}{1-\kappa}\pi\frac{\partial \rho\theta}{\partial z} - \rho g + \nu\rho\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial z^2}\right),$$
$$\frac{\partial \rho\theta}{\partial t} = -\frac{\partial \rho u\theta}{\partial x} - \frac{\partial \rho w\theta}{\partial z}.$$

Here u and w are the horizontal and vertical winds, ρ is the density, θ the potential temperature, g the acceleration of gravity, ν the diffusion coefficient and π the Exner pressure

$$\pi = \left(\frac{R\rho\theta}{p_0}\right)^{\frac{\kappa}{1-\kappa}}.$$

We use a finite volume spatial discretization on an Arakawa C grid, so the winds are defined on the cell edges while all scalar variables are defined in the cell centers as illustrated by Fig. 5. The computation of the red first-order derivatives is done with the third-order upwind scheme, respectively with first-order upwinding for the Jacobian, while the other terms are computed with central differences. In cut cells we use the matrix which belongs to the system

$$\begin{pmatrix} \dot{\rho} \\ (\dot{\rho u}) \\ (\dot{\rho w}) \\ (\dot{\rho\theta}) \end{pmatrix} = -\begin{pmatrix} u & 1 & 0 & 0 \\ 0 & u & 0 & \frac{R}{1-\kappa}\pi \\ 0 & 0 & u & 0 \\ 0 & \theta & 0 & u \end{pmatrix} \begin{pmatrix} \rho_x \\ (\rho u)_x \\ (\rho w)_x \\ (\rho\theta)_x \end{pmatrix} - \begin{pmatrix} w & 0 & 1 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & w & \frac{R}{1-\kappa}\pi \\ 0 & 0 & \theta & w \end{pmatrix} \begin{pmatrix} \rho_z \\ (\rho u)_z \\ (\rho w)_z \\ (\rho\theta)_z \end{pmatrix} - \begin{pmatrix} 0 \\ -\nu((\rho u)_{xx} + (\rho u)_{zz}) \\ \rho g - \nu((\rho w)_{xx} + (\rho w)_{zz}) \\ 0 \end{pmatrix}$$

as Jacobian while in the free atmosphere the Jacobian originates from

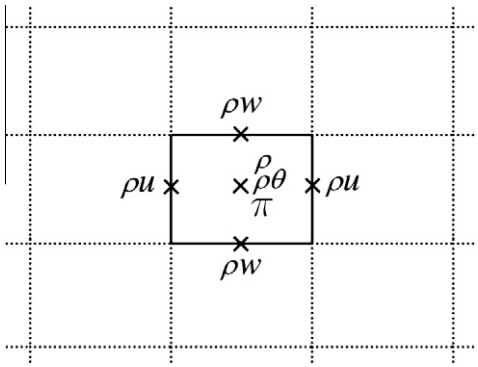

Fig. 5. The positions of the variables on the Arakawa *C* grid.

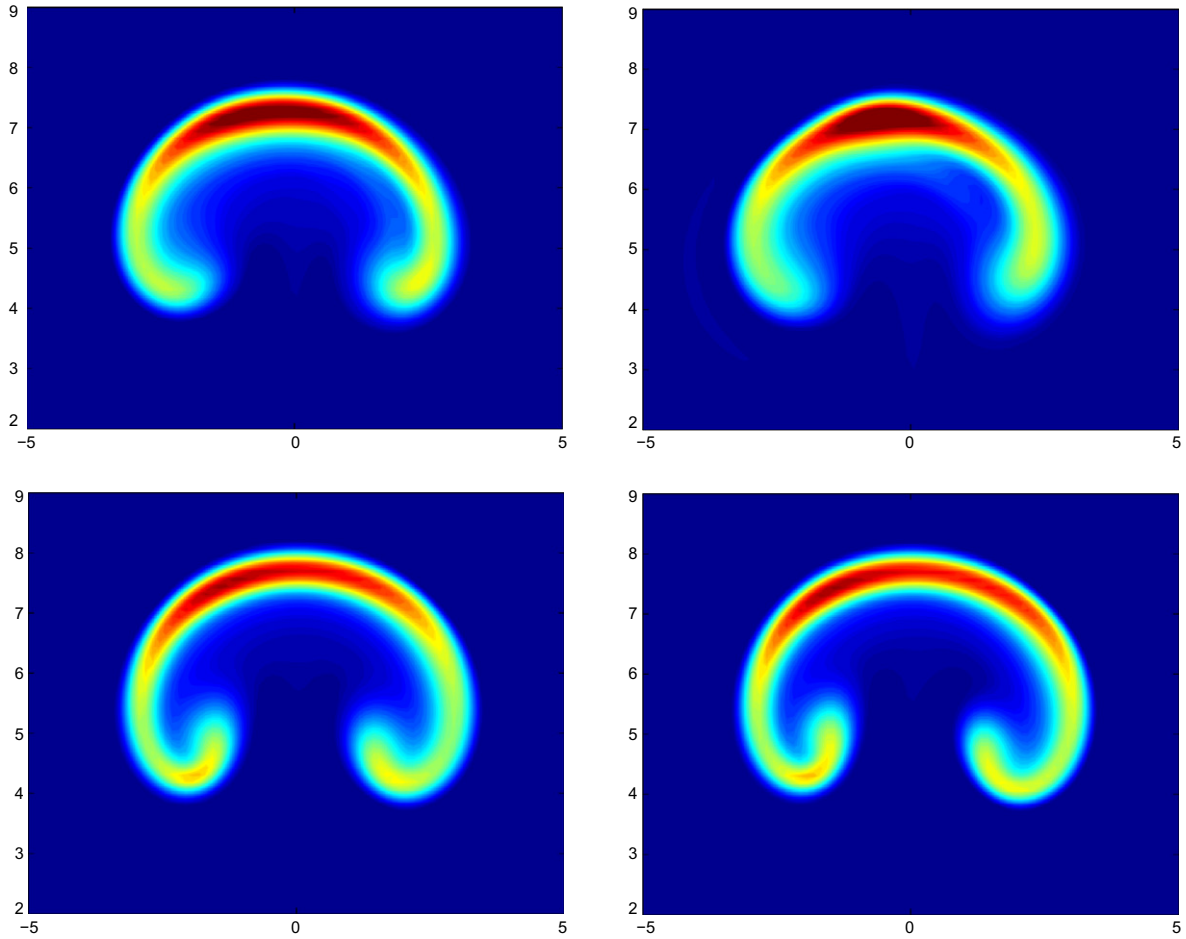

Fig. 6. Potential temperature after 1000 s at the rising bubble test computed with the implicit peer method (left) and ROS3Pw (right) with full Jacobian (top) and partial Jacobian (bottom). Time step size is 3.5 s for ROS3Pw with partial Jacobian, for the other images $\Delta t = 7$ s.

$$\begin{pmatrix} \dot{\rho} \\ (\dot{\rho u}) \\ (\dot{\rho w}) \\ (\dot{\rho\theta}) \end{pmatrix} = -\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{R}{1-\kappa}\pi \\ 0 & 0 & 0 & 0 \\ 0 & \theta & 0 & 0 \end{pmatrix} \begin{pmatrix} \rho_x \\ (\rho u)_x \\ (\rho w)_x \\ (\rho\theta)_x \end{pmatrix} - \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{R}{1-\kappa}\pi \\ 0 & 0 & \theta & 0 \end{pmatrix} \begin{pmatrix} \rho_z \\ (\rho u)_z \\ (\rho w)_z \\ (\rho\theta)_z \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \rho g \\ 0 \end{pmatrix}.$$

The Euler equations are implemented in MATLAB. For the presented test cases the size of the Jacobian is small enough to solve the occurring linear systems of equations analytically with the built-in MATLAB solver which uses LU decomposition. Because peer methods are multi-step methods we need s initial values. These are computed with the split-explicit Runge–Kutta method RK3 from [31] and sufficiently small time steps.

We present the results for four test cases: The first one is the rising bubble which is described in detail in [1]. As initial condition we have an adiabatic atmosphere with the exception of an initial thermal of 2 K with radius and height of 2 km, i.e. the potential temperature is perturbed by

$$\theta' = 2\cos\left(\frac{\pi C}{2}\right),$$

where

$$C = \sqrt{\left(\frac{x - x_c}{x_r}\right)^2 + \left(\frac{z - z_c}{z_r}\right)^2}, \tag{10}$$

$x_c = 0$ km and $z_c = x_r = z_r = 2$ km. We use a 20 km $\times$ 10 km domain with a spatial resolution of 125 m. A uniform horizontal flow of 20 ms^{-1} from the left as in [30] and periodic boundary conditions cause a lateral transport of the bubble which makes the test more stringent. After the integration period of 1000 s the bubble should be located in the center of the domain and remain symmetric. Fig. 6 shows the potential temperature after 1000 s. The integration was performed with a time step size of 7 s with the exception of ROS3Pw with partial Jacobian because this method is unstable with $\Delta t = 7$ s, instead it computes with 3.5 s. The maximum wind speed that occurred was approximately 29 ms^{-1} so that these time step sizes correspond to CFL numbers of 1.6 and 0.8 for advection,respectively 27 and 13.5 with respect to sound waves. When computing with the full Jacobian the results are worse than the solutions computed with the partial Jacobian. Both solutions show asymmetries but the peer method is a little bit more accurate than ROS3Pw as expected when considering the amplitude and phase properties shown in Fig. 4. The solutions computed with the partial Jacobian are quite good despite that the time step sizes are close to the CFL condition from linear stability theory. Here ROS3Pw produces a little bit more symmetric solution but both solutions are rather similar although the peer method uses the double time step size as ROS3Pw. The solutions appear to be much more accurate than the results in [30] which used a second-order split-explicit Runge–Kutta method with a five times smaller CFL number, namely 0.32.

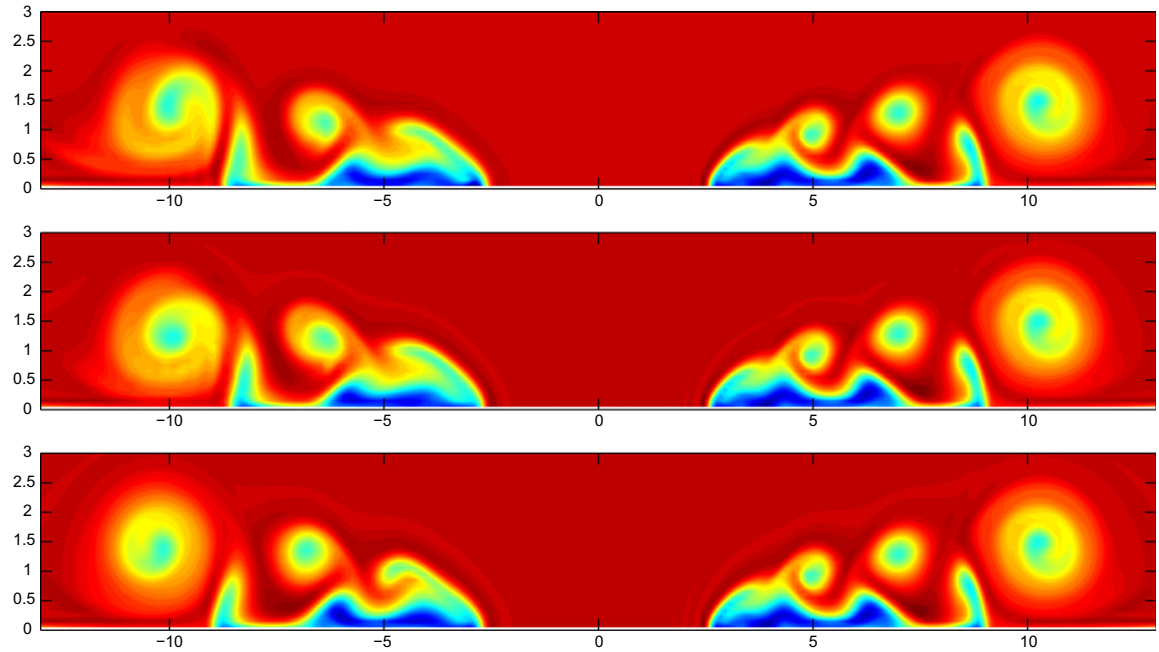

Fig. 7. Potential temperature after 900 s at the density current test computed with the implicit peer method with full Jacobian (top), full Jacobian only in regions with high wind speeds (middle) and partial Jacobian everywhere (bottom). Time step sizes are 3.5 s (top and middle) and 2.5 s (bottom).

The second test problem is related to the first one. Again there is a balanced initial atmosphere but for the density current test the bubble is colder than the surrounding air. The temperature is perturbed by

$$T' = -15 \cos\left(\frac{\pi C}{2}\right)$$

with C from (10) but with $z_c = 3$ km and $x_r = 4$ km. In this test there is diffusion with $\nu = 75\ \mathrm{m^2\ s^{-1}}$ as described in [22]. Because the bubble is colder it sinks and after crashing the bottom (with free slip vertical boundary conditions) several eddies form. The domain has a width of 36 km and the uniform horizontal flow of $20\ \mathrm{ms^{-1}}$ from the left should provide a symmetric solution after 900 s, i.e. half orbit. The spatial resolution is 100 m in both directions. For this test we present the solutions computed with the peer method and the full Jacobian with $\Delta t = 3.5$ s and with the partial Jacobian with $\Delta t = 2.5$ s. Furthermore we compute with $\Delta t = 3.5$ s in one setting where we dynamically adapt the Jacobian so that it incorporates advection, diffusion and acoustics in cells with wind speeds higher than $40\ \mathrm{ms^{-1}}$ while in the remaining cells the Jacobian only contains the acoustic part. Because in this test the temperature difference is higher than in the first test the maximum wind speed is $61\ \mathrm{ms^{-1}}$. The time step sizes correspond to advection CFL numbers of 2.1 for $\Delta t = 3.5$ s and 1.5 for $\Delta t = 2.5$ s. In the test with the dynamically adapted Jacobian the threshold of $40\ \mathrm{ms^{-1}}$ corresponds to an advection CFL number of 1.4. So the CFL numbers are close to the maximum of 1.7 from linear stability theory as in the first test. Fig. 7 shows the results: Every solution shows 3 eddies at each side where the backward directed (right) eddies are better pronounced. This phenomenon was also documented in [31,9] (there the horizontal mean wind came from the right). While the backward directed eddies look rather similar for the 3 settings the forward directed eddies are better pronounced when computing with the smaller time step size and the partial Jacobian. So as in the first test the solution computed with the partial Jacobian is more accurate. Despite of the used CFL numbers close to the maximum of 1.7 all 3 solutions are in agreement with the solution presented in [22] (where the tests were performed without the background wind) with the exception of the asymmetries caused by the lateral transport. Furthermore no noise comes from adapting the Jacobian at high wind speeds, i.e. the peer methods with full and partial Jacobian harmonize very well.

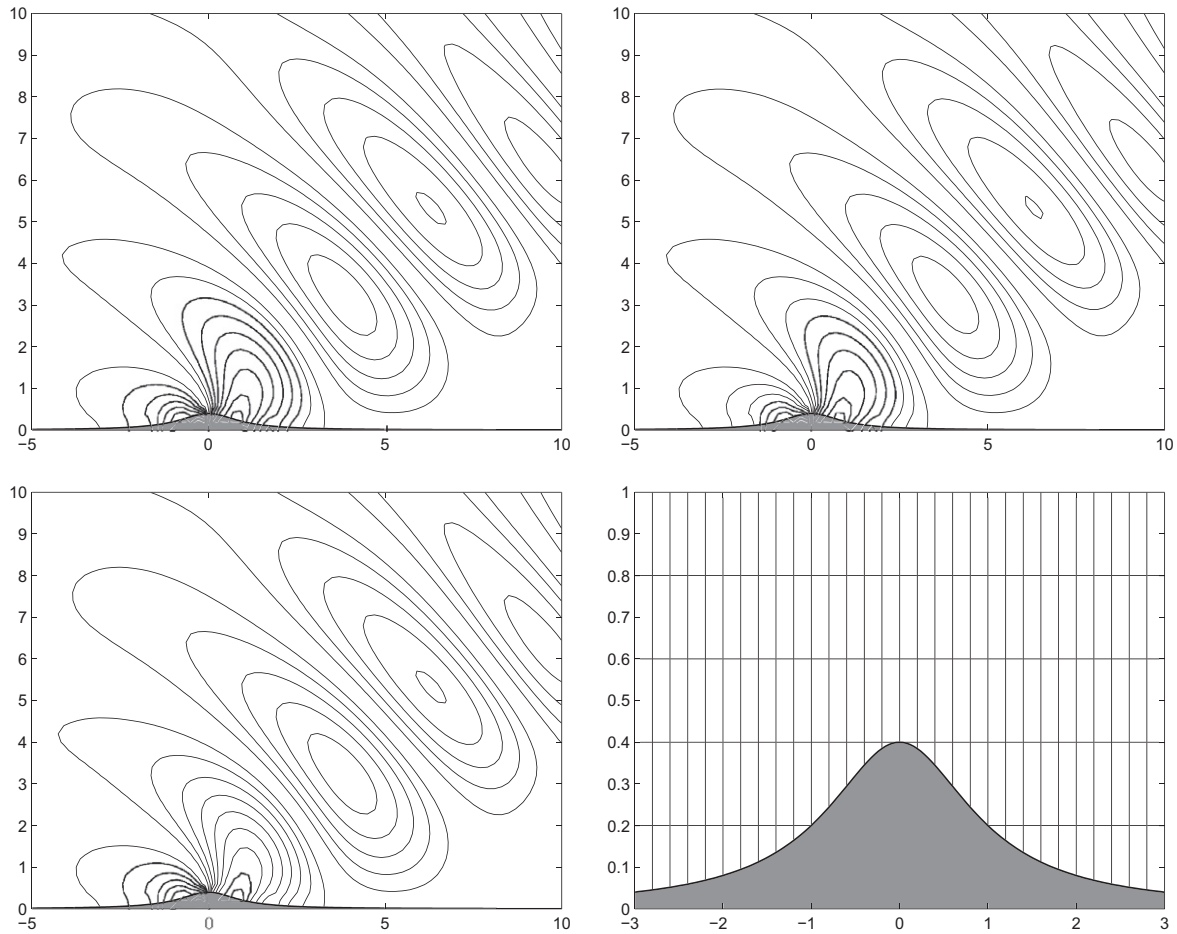

Fig. 8. Vertical wind after 2160 s at the flow over mountain test computed with the implicit peer method (left) and ROS3Pw (top right) with full Jacobian in all cells (top) and full Jacobian in cut cells only (bottom left), all with $\Delta t = 20$ s, and the Witch of Agnesi mountain with cut cells (bottom right).

The third test problem is the flow over a mountain. This problem is described in [3]. There is a mountain with half-width 1 km and height 400 m as shown in Fig. 8. The spatial resolution is 200 m in both directions. A horizontal flow of 10 ms^{-1} from the left results in gravity waves in the lee of the mountain. Fig. 8 shows the vertical wind after 2160 s. Because of the representation of the orography with cut cells very small cells appear, the smallest have a size of 0.17% of the cells in the free regions. Because the initial values which are needed for the peer method are computed with an explicit method the time step size needed for the initialization would be 600 times smaller than the time step size used by the peer method. Therefore we close all cells which are smaller than 1% than the cells in the free atmosphere, i.e. we pretend that the mountain completely cuts out these cells instead of only 99% of them and assume that this has no significant influence on the solution. We use a time step size for the initialization with the explicit method which is only 100 times smaller. The time step size used for the implicit methods is $\Delta t = 20$ s. The maximum wind speed that occurs is 14 ms^{-1} which results in an advection CFL number of 1.4 in the free atmosphere. In the cut cells the advection CFL number is up to 140 and the CFL number for the acoustics is up to 4800. Despite that large CFL numbers all solution are in very good agreement with the results published in [3] and no clear difference is visible between the solutions computed with full and partial Jacobian. These results verify the linear stability theory and confirm that the full Jacobian is needed in cut cells only.

The fourth test is a more escapist one and combines the rising bubble test with cut cells. It provides a more stringent environment to test the effects of different Jacobians because the first test showed how sensitive the shape of the bubble is to the use of different Jacobians. In this test parts of the bubble will be located in cut cells, i.e. parts of the bubble will be computed with the full Jacobian while other parts are updated with the partial Jacobian. A detailed description of the initial conditions can be found in [11] but our setup differs a little bit from that one. There is the same warm bubble as in the first test with the same grid but the bubble is located 1 km in the left and there is no background wind. The main difference is that there is a region in the center of the domain which is cut out, i.e. the grey region shown in Fig. 9 is an obstacle. While the inner cells are completely cut out cut cells appear at the boundaries of this zeppelin, the smallest cells with a size of 0.13% of the full cells. For the same reason as in the third test we close tiny cells so that the smallest cells have a minimum size of 1% of the cells in

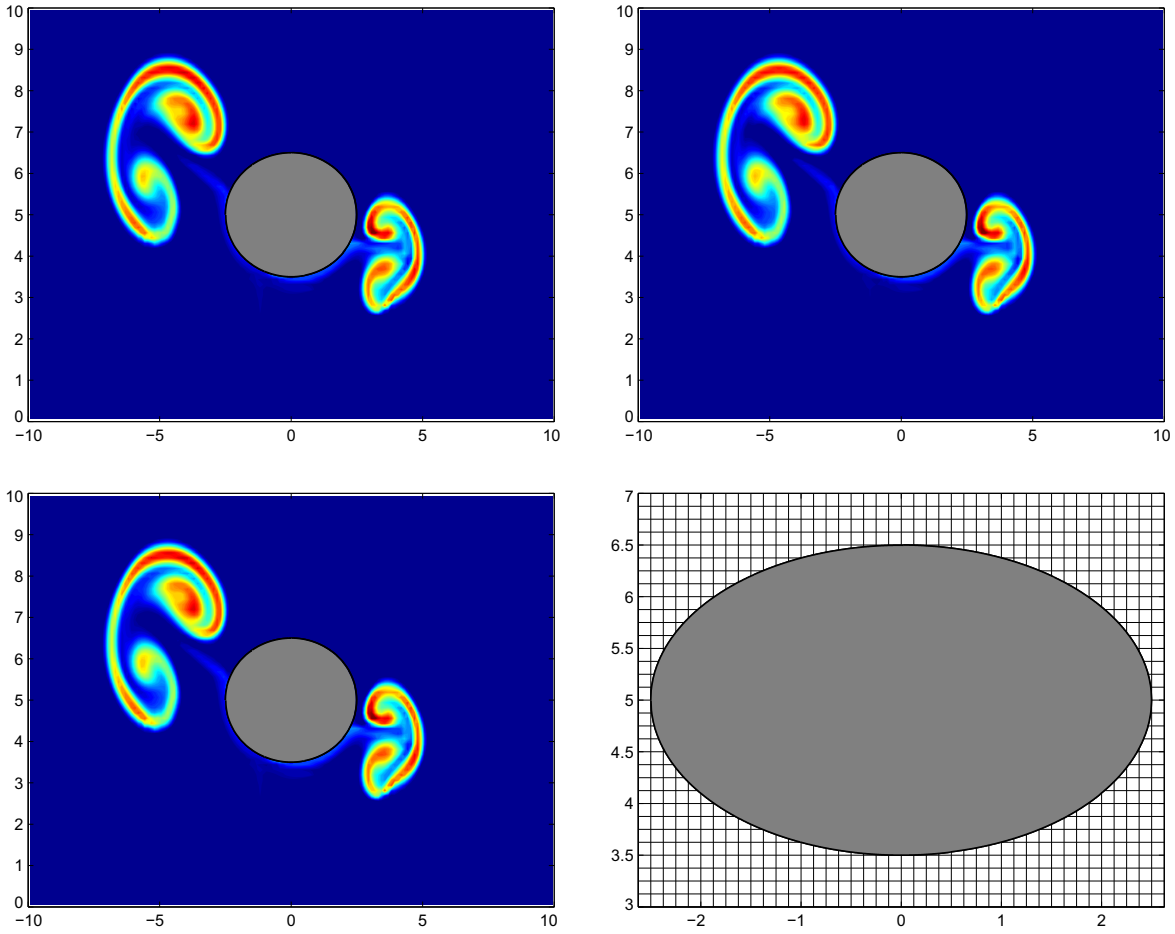

Fig. 9. Potential temperature after 1250 s at the zeppelin test computed with the implicit peer method (left) and ROS3Pw (top right) with full Jacobian in all cells (top) and full Jacobian in cut cells only (bottom left), all with $\Delta t = 10$ s, and the zeppelin with cut cells (bottom right).

the free atmosphere. The bubble will crash against the obstacle and will asymmetrically divide into two separate parts. The implicit schemes compute with $\Delta t = 10$ s and the maximum wind speed is approximately 17. 5 ms^{-1} what results in CFL numbers of 1.4 for advection and 38 for acoustics in the free atmosphere respectively 140 and 3800 in the smallest cut cells. As in the third test there is no qualitative difference between the different methods so again it is sufficient to use the full Jacobian in cut cells only.

5. Conclusions

We presented a class of linearly implicit peer methods for the integration of the compressible Euler equations. We derived order conditions which allow the construction of methods that retain the full order independently of what is used as Jacobian. The presented partially implicit peer method is as stable and accurate as the underlying split-explicit method but furthermore it can compute with cut cells with nearly no additional effort if the number of cut cells is sufficiently small what is true in numerical weather prediction.

In order to save storage and gain computational efficiency we used a simplified Jacobian which corresponds to the advection form of the Euler equation while the conservative form was used as right-hand side to conserve mass, momentum and entropy. The simplified Jacobian uses a lower-order spatial discretization. Furthermore the simplified Jacobian is used in cut cells only while in the remaining domain a partial Jacobian which only incorporates the acoustic part of the Euler equations is used. Table 1 shows the amount of storage which is needed for the Jacobian per grid cell, i.e. the number of non-zero entries in 3 rows (respectively 4 rows in 2D and 5 rows in 3D without the consideration of gravity, diffusion, Coriolis force, etc.). While central differences and the first-order upwind scheme need 2 entries per derivative the third-order upwind scheme needs at least 4 entries. Therefore the simplified Jacobian only needs 41% of the storage needed for the exact Jacobian in 2D. If there are no cut cells the partial Jacobian is used which only has 43% of the number of entries in comparison to the simplified Jacobian. Because in numerical weather prediction models orography appears at ground cut cells can be located only there, i.e. in a model with 50 vertical layers only 2% of the cells in the whole domain are cut cells which is negligible from the storage point of view. So the right column in Table 1 shows the saving of storage when using the simplified Jacobian in cut cells only instead of everywhere. Furthermore there is a theoretical speed-up of 2.3 for a matrix–vector multiplication when using the partial Jacobian instead of the simplified Jacobian. The practical speed-up might be even larger because there have to be computed more entries and the systems might be worse conditioned if the full Jacobian is used. In comparison to the exact Jacobian the speed-up would be even 5.7. The computing time for the zeppelin test with the full but simplified Jacobian (without the time needed for computing the initial values) is 608 s on an Intel Core2 Quad Q9550 @ 2833 MHz with 3 GB RAM. When using the simplified Jacobian in cut cells only the computation takes 405 s, i.e. 33% of the computing time is saved what corresponds to a speed-up of 1.5. The times needed to set up and solve the linear systems of equations are 310 s respectively 113 s, i.e. the speed-up for this part of the solver is 2.7. When applying the split-explicit peer method with the trapezoidal rule to the flow over a mountain test case with a time step size of 0.2 s the computation takes 13146 s (3812 s) where the time in parentheses is needed for setting up and solving the linear systems of equations. The times for the implicit peer method with the same time step size are 13251 s (3975 s) when the full Jacobian is used in cut cells only and 20011 s (10560 s) when it is used everywhere. So the speed-up when using the full Jacobian in cut cells only is 1.5 (2.6). The difference in computing time between the split-explicit method and the partially implicit method is just about 1% (4%), i.e. the partially implicit method is as efficient as the split-explicit method. As mentioned before the systems were solved with the built-in MATLAB solver which uses LU decomposition so the results may differ in 3D applications where iterative solvers are used. On the other hand the theoretical speed-up increases in higher dimensions. Nevertheless these values give a good insight in what computational time can be saved when computing the full Jacobian in cut cells only. Furthermore the partial and simplified Jacobians are easier to implement than the exact Jacobian due to the smaller stencils what makes the implementation of block-structured grids and parallelization more comfortable. We gave a detailed insight into the stability analysis which incorporates the influences of these simplifications.

The peer method we presented has three stages, is second-order and A-stable in the common sense (not shown). It originates from the split-explicit method from [9]. For wind speeds smaller than $c_s/6$ it is stable for arbitrary large advection and acoustic CFL numbers if the simplified Jacobian is used. It is as stable as the underlying split-explicit method, i.e. stable for arbitrary large acoustic CFL numbers and $C_{adv} < 1.7$, if the Jacobian only incorporates the acoustic part of the compressible Euler equations. We found the presented peer method with a genetic algorithm where we optimized the degrees of freedom with respect to small amplitude and phase errors with the desired stability criteria as side conditions. We applied it to several test cases. Besides the presented tests we also applied it to the 2D flow around a cylinder test which is described in [16] and to the dam-break problem for the shallow water equations published in [24]. As for the flow around a mountain test and the zeppelin test the results were all the same for the tests with cut cells involved: No differences between the solutions computed with the full Jacobian in cut cells only and the full Jacobian everywhere are visible. Despite of the large CFL numbers in cut cells of $\mathcal{O}(100)$ for advection respectively $\mathcal{O}(5000)$ for sound waves and advection CFL numbers in the free atmosphere which are close to 1.7 the solutions show no noise or instabilities, i.e. the results from linear stability theory are valid for the nonlinear Euler equations. Furthermore the peer method with partial and full Jacobian harmonizes very well even if the Jacobian is dynamically adapted as in the density current test where the wind speed determines whether advection and diffusion are incorporated in the Jacobian or not. Because of the high wind speeds in the rising bubble and

the density current test there are some differences visible between the solutions computed with partial and full Jacobians. The solutions where advection is treated explicitly are qualitatively better, i.e. not only from the efficiency point of view but also because of the accuracy the Jacobian should incorporate advection only where it is necessary for stability. There are 2 reasonable employments for treating advection implicitly not only in cut cells but also in cells with high wind speeds: Firstly in numerical weather prediction models the jet stream has to be implemented which can reach velocities of more than $100\,\mathrm{ms}^{-1}$. This is much faster than the wind speeds in the remaining atmosphere, i.e. in explicit models which use no multi-rate schemes the CFL condition for the jet stream limits the maximum time step size. If the Jacobian incorporates advection in regions where the jet stream is known to be or in regions with high wind speeds the time step size is not restricted by the speed of the jet stream what results in computational efficiency. Secondly in numerical weather prediction the used time step sizes are not close to the CFL condition because the occurring wind speeds are not known a priori. A dynamical adaption of the Jacobian in case of higher wind speeds allows larger time steps. Furthermore this strategy guarantees that the CFL condition can not be violated, i.e. this ansatz not only results in computational efficiency but also in reliability of the model. Furthermore the successful test with the dynamically adapted Jacobian gives the hint that our approach is suitable for the combination of partially implicit peer methods with grids that use adaptive mesh refinement techniques. If for example adaptive grids are used to track and better resolve a hurricane with smaller spatial step sizes a method with a dynamically adaptive Jacobian could use larger time step sizes if it is implicit in the better resolved regions. Another conceivable application are latitude–longitude grids because the pole singularities would no longer result in time step restrictions if the method is implicit in these regions with small spatial step sizes.

Acknowledgements

This work is a contribution to the project MetStröm (Skalenübergreifende Modellierung in der Stromungsmechanik und Meteorologie) funded by the German Research Foundation (DFG).

Appendix A. Coefficients of the peer method

The coefficients of the second-order three-stage peer method presented in this paper are:

$$c = (\,-0.0899531627878552 \quad 0.4676428830697650 \quad 1\,)^T,$$

$$\gamma = (\,0.5167851598100672 \quad 0.5167851598100672 \quad 0.5167851598100672\,)^T,$$

$$B = \begin{pmatrix} -0.0967059983845656 & 0.4915598645202344 & 0.6051461338643311 \\ -0.0470929826281593 & 0.2169946581702936 & 0.5720815963722115 \\ -0.0891437312845480 & 0.1573830315884013 & 0.1973233392586685 \end{pmatrix},$$

$$S = \begin{pmatrix} 0 & 0 & 0 \\ 0.2580167280856541 & 0 & 0 \\ 0.3269306113397434 & 0.4075067490977347 & 0 \end{pmatrix},$$

$$A = \begin{pmatrix} 0.0721007322008575 & -0.1322804288331288 & 0.1265069173192104 \\ 0.0478238719665258 & -0.4831372398722279 & -0.1163332106046261 \\ 0.0325906971440313 & 0.0702440095890842 & 0.1286761505892647 \end{pmatrix},$$

$$R = \begin{pmatrix} 0 & 0 & 0 \\ 1.1066883875756954 & 0 & 0 \\ -0.5020271673748957 & 1.0959786066300778 & 0 \end{pmatrix},$$

$$G = \begin{pmatrix} -0.2873500776711067 & 0.5010014103242811 & -0.7304364924632416 \\ -0.5198912797002499 & 1.5416287759361538 & -1.4010177204289034 \\ -0.3557339562292955 & 0.7879731539945604 & -0.9973424995819653 \end{pmatrix},$$

$$H = \begin{pmatrix} 0 & 0 & 0 \\ -0.1375049356170674 & 0 & 0 \\ 0.9103920894669218 & -0.8620739474602885 & 0 \end{pmatrix}.$$

Appendix B. Implementation of the Jacobian

We now present the implementation of the Jacobian. For this purpose we start with the discretization of the compressible Euler equations with finite volumes. We introduce three notations for the variables: If they are written in black without a tilde they are defined at the locations of their indices which are shown in Fig. 10. Tilde variables are obtained by averaging, e.g.

$$\widetilde{\rho}_{23} := \frac{\rho_{22} + \rho_{24}}{2}.$$

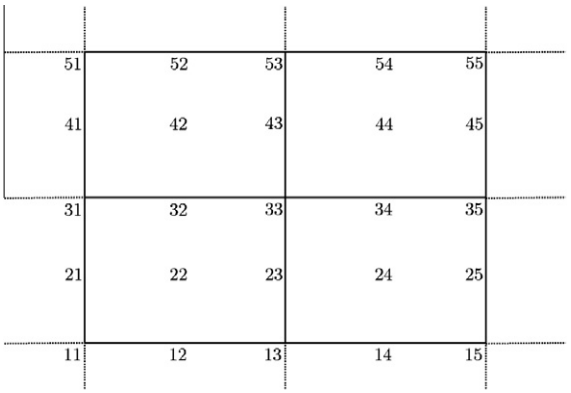

Fig. 10. Four cells of the grid with indices.

If the variable is written in red it is generated at the location of its index by the third-order upwind scheme. θ can simply be obtained by dividing $\rho\theta$ with ρ because both variables are defined at the same location. For the wind speeds we divide by the arithmetic mean of the densities, e.g.

$$u_{23} := \frac{2(\rho u)_{23}}{\widetilde{\rho}_{23}} = \frac{2(\rho u)_{23}}{\rho_{22}+\rho_{24}}.$$

The means of the Exner pressure are needed for the pressure derivatives, to obtain them we average the potential temperature, e.g.

$$\widetilde{\pi}_{23} := \left(\frac{R(\widetilde{\rho\theta})_{23}}{p_0}\right)^{\frac{\kappa}{1-\kappa}} = \left(\frac{R((\rho\theta)_{22}+(\rho\theta)_{24})}{2p_0}\right)^{\frac{\kappa}{1-\kappa}}.$$

The spatial discretization of the compressible Euler equations with finite volumes reads

$$\dot{\rho}_{22} = -(\rho u)_{23}\frac{(\Delta z)_{23}}{V_{22}} + (\rho u)_{21}\frac{(\Delta z)_{21}}{V_{22}} - (\rho w)_{32}\frac{(\Delta x)_{32}}{V_{22}} + (\rho w)_{12}\frac{(\Delta x)_{12}}{V_{22}},$$

$$\begin{aligned}(\rho\dot{u})_{23} = &-(\widetilde{\rho u})_{24}u_{24}\frac{(\Delta z)_{24}}{V_{23}} + (\widetilde{\rho u})_{22}u_{22}\frac{(\Delta z)_{22}}{V_{23}} - (\widetilde{\rho w})_{33}u_{33}\frac{(\Delta x)_{33}}{V_{23}} + (\widetilde{\rho w})_{13}u_{13}\frac{(\Delta x)_{13}}{V_{23}} - \frac{R}{1-\kappa}\widetilde{\pi}_{23}(\rho\theta)_{24}\frac{(\Delta z)_{23}}{V_{23}}\\ &+\frac{R}{1-\kappa}\widetilde{\pi}_{23}(\rho\theta)_{22}\frac{(\Delta z)_{23}}{V_{23}} + \nu\widetilde{\rho}_{23}\Delta u,\end{aligned}$$

$$\begin{aligned}(\rho\dot{w})_{32} = &-(\widetilde{\rho u})_{33}w_{33}\frac{(\Delta z)_{33}}{V_{32}} + (\widetilde{\rho u})_{31}w_{31}\frac{(\Delta z)_{31}}{V_{32}} - (\widetilde{\rho w})_{42}w_{42}\frac{(\Delta x)_{42}}{V_{32}} + (\widetilde{\rho w})_{22}w_{22}\frac{(\Delta x)_{22}}{V_{32}} - \frac{R}{1-\kappa}\widetilde{\pi}_{32}(\rho\theta)_{42}\frac{(\Delta x)_{32}}{V_{32}}\\ &+\frac{R}{1-\kappa}\widetilde{\pi}_{32}(\rho\theta)_{22}\frac{(\Delta x)_{32}}{V_{32}} - \widetilde{\rho}_{32}g + \nu\widetilde{\rho}_{32}\Delta w,\end{aligned}$$

$$(\rho\dot{\theta})_{22} = -(\rho u)_{23}\theta_{23}\frac{(\Delta z)_{23}}{V_{22}} + (\rho u)_{21}\theta_{21}\frac{(\Delta z)_{21}}{V_{22}} - (\rho w)_{32}\theta_{32}\frac{(\Delta x)_{32}}{V_{22}} + (\rho w)_{12}\theta_{12}\frac{(\Delta x)_{12}}{V_{22}}.$$

The Laplace operator Δ for the diffusion terms is discretized with finite differences independently whether the cells are cut by orography or not. The reason why we use the above discretization of the pressure derivatives instead of

$$-\frac{R}{1-\kappa}\widetilde{\pi}_{23}(\rho\theta)_{24}\frac{(\Delta z)_{24}}{V_{23}} + \frac{R}{1-\kappa}\widetilde{\pi}_{23}(\rho\theta)_{22}\frac{(\Delta z)_{22}}{V_{23}}$$

and

$$-\frac{R}{1-\kappa}\widetilde{\pi}_{32}(\rho\theta)_{42}\frac{(\Delta x)_{42}}{V_{32}} + \frac{R}{1-\kappa}\widetilde{\pi}_{32}(\rho\theta)_{22}\frac{(\Delta x)_{22}}{V_{32}}$$

is because the latter ansatz would cause artificial pressure perturbations in cut cells due to the different lengths of the edges $(\Delta z)_{24}$ and $(\Delta z)_{22}$ respectively $(\Delta x)_{42}$ and $(\Delta x)_{22}$.

We now state the Jacobian for our spatial discretization with finite volumes. As explained before the advection terms are discretized with the first-order upwind scheme while the acoustics use central differences. The diffusion is treated with second-order finite differences instead of finite volumes. We define

$$J := \begin{pmatrix} J_{\rho\rho} & J_{\rho u} & J_{\rho w} & 0 \\ 0 & J_{uu}+J_{\Delta u} & 0 & J_{u\theta} \\ J_{w\rho} & 0 & J_{ww}+J_{\Delta w} & J_{w\theta} \\ 0 & J_{\theta u} & J_{\theta w} & J_{\theta\theta} \end{pmatrix}$$

and assume negative wind speeds, i.e. if variables are interpolated with the first-order upwind scheme they will be shifted leftwards respectively downwards to the locations where they are needed for the advection terms. For the advection we also need the densities at cell vertices. This is done again with the arithmetic mean, e.g.

$$\widetilde{\rho}_{33} := \frac{\rho_{22}+\rho_{24}+\rho_{42}+\rho_{44}}{4}.$$

The locations of the entries of the Jacobian are given in parentheses behind the Jacobian. For example $-(\rho u)_x$ at location 22 is discretized with

$$-(\rho u)_{23}\frac{(\Delta z)_{23}}{V_{22}} + (\rho u)_{21}\frac{(\Delta z)_{21}}{V_{22}},$$

which is equivalent to

$$\begin{pmatrix} -\frac{(\Delta z)_{23}}{V_{22}} & \frac{(\Delta z)_{21}}{V_{22}} \end{pmatrix} \begin{pmatrix} (\rho u)_{23} \\ (\rho u)_{21} \end{pmatrix}.$$

Therefore $J_{\rho u}(22,23) = -\frac{(\Delta z)_{23}}{V_{22}}$ and $J_{\rho u}(22,21) = \frac{(\Delta z)_{21}}{V_{22}}$, i.e. the Jacobian has these entries in the row which belongs to ρ_{22} and in the columns which belong to $(\rho u)_{23}$ respectively $(\rho u)_{21}$. The whole Jacobian is the sum of these components:

$$J_{\rho\rho}(22,24) = -u_{23}\frac{(\Delta z)_{23}}{V_{22}}, \quad J_{\rho\rho}(22,22) = u_{21}\frac{(\Delta z)_{21}}{V_{22}},$$
$$J_{\rho\rho}(22,42) = -w_{32}\frac{(\Delta x)_{32}}{V_{22}}, \quad J_{\rho\rho}(22,22) = w_{12}\frac{(\Delta x)_{12}}{V_{22}},$$
$$J_{\rho u}(22,23) = -\frac{(\Delta z)_{23}}{V_{22}}, \quad J_{\rho u}(22,21) = \frac{(\Delta z)_{21}}{V_{22}},$$
$$J_{\rho w}(22,32) = -\frac{(\Delta x)_{32}}{V_{22}}, \quad J_{\rho w}(22,12) = \frac{(\Delta x)_{12}}{V_{22}},$$
$$J_{uu}(23,25) = -\frac{(\widetilde{\rho u})_{24}}{\rho_{24}}\frac{(\Delta z)_{24}}{V_{23}}, \quad J_{uu}(23,23) = \frac{(\widetilde{\rho u})_{22}}{\rho_{22}}\frac{(\Delta z)_{22}}{V_{23}},$$
$$J_{uu}(23,43) = -\frac{(\widetilde{\rho w})_{33}}{\widetilde{\rho}_{33}}\frac{(\Delta x)_{33}}{V_{23}}, \quad J_{uu}(23,23) = \frac{(\widetilde{\rho w})_{13}}{\widetilde{\rho}_{13}}\frac{(\Delta x)_{13}}{V_{23}},$$
$$J_{u\theta}(23,24) = -\frac{R}{1-\kappa}\widetilde{\pi}_{23}\frac{(\Delta z)_{23}}{V_{23}}, \quad J_{u\theta}(23,22) = \frac{R}{1-\kappa}\widetilde{\pi}_{23}\frac{(\Delta z)_{23}}{V_{23}},$$
$$J_{w\rho}(32,42) = -\frac{g}{2}, \quad J_{w\rho}(32,22) = -\frac{g}{2},$$
$$J_{ww}(32,34) = -\frac{(\widetilde{\rho u})_{33}}{\widetilde{\rho}_{33}}\frac{(\Delta z)_{33}}{V_{32}}, \quad J_{ww}(32,32) = \frac{(\widetilde{\rho u})_{31}}{\widetilde{\rho}_{31}}\frac{(\Delta z)_{31}}{V_{32}},$$
$$J_{ww}(32,52) = -\frac{(\widetilde{\rho w})_{42}}{\rho_{42}}\frac{(\Delta x)_{42}}{V_{32}}, \quad J_{ww}(32,32) = \frac{(\widetilde{\rho w})_{22}}{\rho_{22}}\frac{(\Delta x)_{22}}{V_{32}},$$
$$J_{w\theta}(32,42) = -\frac{R}{1-\kappa}\widetilde{\pi}_{32}\frac{(\Delta x)_{32}}{V_{32}}, \quad J_{w\theta}(32,22) = \frac{R}{1-\kappa}\widetilde{\pi}_{32}\frac{(\Delta x)_{32}}{V_{32}},$$
$$J_{\theta u}(22,23) = -\theta_{24}\frac{(\Delta z)_{23}}{V_{22}}, \quad J_{\theta u}(22,21) = \theta_{22}\frac{(\Delta z)_{21}}{V_{22}},$$
$$J_{\theta w}(22,32) = -\theta_{42}\frac{(\Delta x)_{32}}{V_{22}}, \quad J_{\theta w}(22,12) = \theta_{22}\frac{(\Delta x)_{12}}{V_{22}},$$
$$J_{\theta\theta}(22,24) = -u_{23}\frac{(\Delta z)_{23}}{V_{22}}, \quad J_{\theta\theta}(22,22) = u_{21}\frac{(\Delta z)_{21}}{V_{22}},$$
$$J_{\theta\theta}(22,42) = -w_{32}\frac{(\Delta x)_{32}}{V_{22}}, \quad J_{\theta\theta}(22,22) = w_{12}\frac{(\Delta x)_{12}}{V_{22}},$$
$$J_{\Delta u}(23,25) = \frac{\nu}{(\Delta x)^2}, \quad J_{\Delta u}(23,23) = -\frac{2\nu}{(\Delta x)^2} - \frac{2\nu}{(\Delta z)^2},$$
$$J_{\Delta u}(23,21) = \frac{\nu}{(\Delta x)^2}, \quad J_{\Delta w}(32,52) = \frac{\nu}{(\Delta z)^2},$$

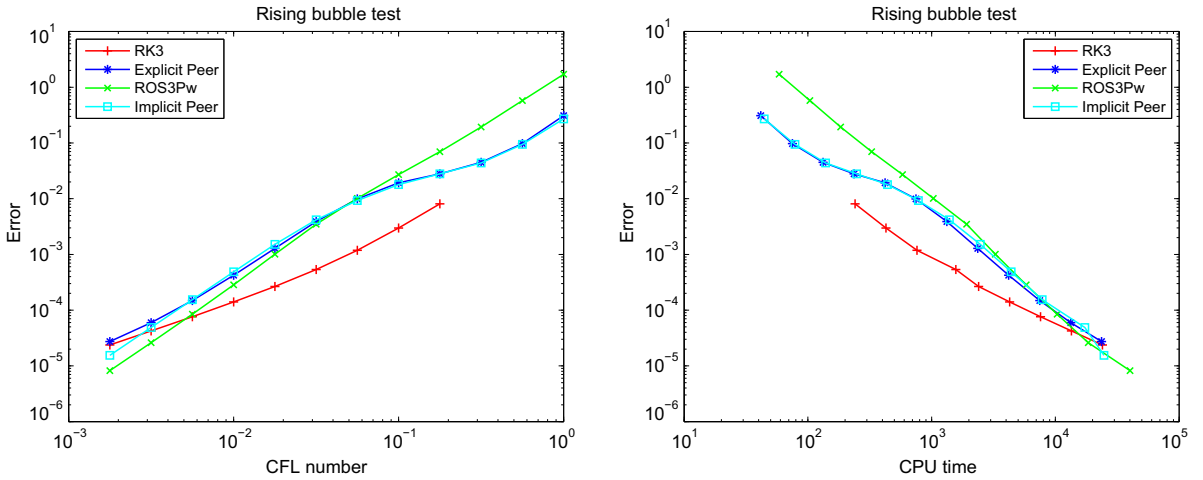

Fig. 11. Error of the numerical solution for the rising bubble test in L^2 norm against CFL number (left) and CPU time (right).

$$J_{\Delta w}(32,32) = -\frac{2v}{(\Delta x)^2} - \frac{2v}{(\Delta z)^2}, \quad J_{\Delta w}(32,12) = \frac{v}{(\Delta z)^2}.$$

Δx and Δz in the discretization of the Laplace operator are the spatial resolutions of the grid in the free atmosphere.

For the presented two-dimensional test cases the size of the Jacobian is small enough to solve the occurring linear systems of equations analytically with the built-in MATLAB solver which uses LU decomposition.

Appendix C. Order test

To test the order we consider the rising bubble test case with the only exception of the doubled spatial step size of $\Delta x = 250$ m. The maximum wind speed which occurs is 27 ms^{-1}. The reference solution has been computed with a third-order linearly implicit peer method with a time step size of $\Delta t = 0.01$ s which corresponds to a CFL number of approximately 1/1000.

With $e_{\rho u}$, $e_{\rho w}$ and $e_{\rho\theta}$ being the errors of ρu, ρw and $\rho\theta$ in the discrete L^2 norm Fig. 11 shows $\sqrt{e_{\rho u}^2 + e_{\rho w}^2 + e_{\rho\theta}^2}$ in dependency of the CFL number and the needed CPU time. We also show the errors for the well known split-explicit Runge–Kutta method RK3 from [31] and the split-explicit peer method from [9] which is the underlying method for the linearly implicit peer method presented in this paper. The split-explicit methods use the forward–backward Euler scheme with 30 small time steps per large time step. This test confirms the results from the order theory. The linearly implicit methods show the expected order, i.e. order 2. They use the partial Jacobian but the results are similar when they compute with the full Jacobian (not shown). For small CFL numbers the order of the split-explicit methods decreases to 1 because the forward–backward Euler scheme is first-order. We show the results for RK3 without divergence damping, i.e. it becomes unstable for large CFL numbers. Therefore the error is not plotted for RK3 with large CFL numbers.

Fig. 11 also shows the error against the CPU time on an Intel Core 2 Quad Q9550 @ 2833 MHz with 3 GB RAM. Qualitatively there is no difference to the error against CFL number plot. But we have to make two restrictions for this statement: Firstly we implemented the three-stage Rosenbrock method ROS3Pw as a four-stage peer method. Secondly the occurring linear systems of equations are solved with the built-in MATLAB solver. With this implementation the effort of using a linearly implicit method with the partial Jacobian is similar to the effort of integrating the fast part with the forward–backward Euler scheme with 30 small time steps per large time step for split-explicit methods. This may be different when using efficient iterative solvers and the linearly implicit methods might get even better.

References

[1] G.H. Bryan, J.M. Fritsch, A benchmark simulation for moist nonhydrostatic numerical models, Mon. Wea. Rev. 130 (2002) 2917–2928.
[2] K. Burrage, J.C. Butcher, Non-linear stability of a general class of differential equation methods, BIT 20 (1980) 185–203.
[3] W. Gallus, J.B. Klemp, Behaviour of flow over step orography, Mon. Wea. Rev. 128 (2000) 1153–1164.
[4] A. Gerisch, J. Lang, H. Podhaisky, R. Weiner, High-order linearly implicit two-step peer-finite element methods for time-dependent PDEs, Appl. Numer. Math. 59 (2009) 624–638.
[5] E. Hairer, S.P. Nrsett, G. Wanner, Solving Ordinary Differential Equations I, second ed., Springer-Verlag, 1993. pp. 528.
[6] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, second ed., Springer-Verlag, 1996. pp. 614.
[7] D. Hinneburg, O. Knoth, Non-dissipative cloud transport in Eulerian grid models by the volume-of-fluid (VOF) method, Atmos. Environ. 39 (2005) 4321–4330.
[8] Z. Jackiewicz, General Linear Methods for Ordinary Differential Equations, John Wiley & Sons, 2009. pp. 482.

[9] S. Jebens, O. Knoth, R. Weiner, Explicit two-step peer methods for the compressible Euler equations, Mon. Wea. Rev. 137 (2009) 2380–2392.
[10] S. Jebens, O. Knoth, R. Weiner, Linearly implicit peer methods for the compressible Euler equations, Appl. Numer. Math., inpress .
[11] R. Klein, K.R. Bates, N. Nikiforakis, Well-balanced compressible cut-cell simulation of atmospheric flow, Phil. Trans. R. Soc. 367 (2009) 4559–4575.
[12] J. Lang, J.G. Verwer, ROS3P – an accurate third-order Rosenbrock solver designed for parabolic problems, BIT 41 (2001) 730–737.
[13] R.A. Pielke et al, A comprehensive meteorological modeling system – RAMS, Meteor. Atmos. Phys. 49 (1992) 69–91.
[14] H. Podhaisky, R. Weiner, B.A. Schmitt, Rosenbrock-type 'peer' two-step methods, Appl. Numer. Math. 53 (2005) 409–420.
[15] J. Rang, L. Angermann, New Rosenbrock *W*-methods of order 3 for PDAEs of index 1, BIT 45 (2005) 761–787.
[16] M. Schäfer, S. Turek, Benchmark computations of laminar flow around a cylinder, Notes Numer. Fluid Mech. 52 (1996) 547–566.
[17] B.A. Schmitt, R. Weiner, S. Jebens, Parameter optimization for explicit parallel peer two-step methods, Appl. Numer. Math. 59 (2009) 769–782.
[18] W.C. Skamarock, J. Dudhia, D. Gill, D. Barker, W. Wei, J. Powers, A description of the advanced research WRF version 2, NCAR Tech. Note, NCAR/TN-468+STR, 2008, pp. 88.
[19] W.C. Skamarock, J.B. Klemp, A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, J. Comput. Phys. 227 (2008) 3465–3485.
[20] G. Steinebach, Order-reduction of ROW-methods for DAEs and method of lines applications, Preprint 1741, Technische Universität Darmstadt, Darmstadt, 1995.
[21] J. Steppeler, H.W. Bitzer, M. Minotte, L. Bonaventura, Nonhydrostatic atmospheric modeling using a *z*-coordinate representation, Mon. Wea. Rev. 130 (2002) 2143–2149.
[22] J. Straka, R.B. Wilhelmson, L.J. Wicker, J.R. Anderson, K. Droegemeier, Numerical solutions of a nonlinear density current: a benchmark solution and comparisons, Int. J. Numer. Methods Fluids 17 (1993) 1–22.
[23] H. Sundqvist, On vertical interpolation and truncation in connexion with use of sigma system models, Atmosphere 14 (1976) 37–52.
[24] E. Tadmor, W. Zhong, Energy-preserving and stable approximations for the two-dimensional shallow water equations, Mathematics and Computation, a Contemporary View, Springer-Verlag, 2008. pp. 67–94.
[25] R.L. Walko, A mass- and momentum-conserving cut cell method for C-staggered grids, talk given at cut cell methods for atmosphere and ocean modeling 2010, Berlin, Germany, 05.07.2010.
[26] R.L. Walko, R. Avissar, The ocean–land–atmosphere model (OLAM). Part II: Formulation and tests of the nonhydrostatic dynamic core, Mon. Wea. Rev. 136 (2008) 4045–4062.
[27] R. Weiner, K. Biermann, B.A. Schmitt, H. Podhaisky, Explicit two-step peer methods, Comput. Math. Appl. 55 (2008) 609–619.
[28] R. Weiner, B.A. Schmitt, H. Podhaisky, Parallel 'peer' two-step *W*-methods and their application to MOL-systems, Appl. Numer. Math. 48 (2004) 425–439.
[29] R. Weiner, B.A. Schmitt, H. Podhaisky, S. Jebens, Superconvergent explicit two-step peer methods, J. Comput. Appl. Math. 223 (2009) 753–764.
[30] L.J. Wicker, W.C. Skamarock, A time-splitting scheme for the elastic equations incorporating second-order Runge–Kutta time differencing, Mon. Wea. Rev. 126 (1998) 1992–1999.
[31] L.J. Wicker, W.C. Skamarock, Time-splitting methods for elastic models using forward time schemes, Mon. Wea. Rev. 130 (2002) 2088–2097.