



Dissertation

Explicit and Linearly Implicit Peer Methods for the Solution of the Compressible Euler Equations

Martin-Luther-Universität Halle-Wittenberg
Naturwissenschaftliche Fakultät II
Institut für Mathematik

von Stefan Jebens
geboren am 17.03.1984
in der Hansestadt Hamburg

Gutachter:

Prof. Dr. Rüdiger Weiner (Martin-Luther-Universität Halle-Wittenberg)

Prof. Dr. Jens Lang (Technische Universität Darmstadt)

Verteidigung am 28. Oktober 2011

Für Ines

Dank

An dieser Stelle möchte ich mich herzlich bei Prof. Dr. Rüdiger Weiner für seine fortwährende Unterstützung während meiner Promotionszeit bedanken. Seine Ratschläge waren immer eine große Hilfe beim Erstellen der wissenschaftlichen Publikationen und der Dissertation. Weiterer Dank gebührt Dr. Oswald Knoth dafür, dass er mit seinem umfassenden Wissen über Strömungsmechanik geholfen hat und die hier präsentierten Verfahren in sein eigenes Strömungsmodell implementiert hat. Außerdem möchte ich den Kollegen der Arbeitsgruppe numerische Mathematik an der Martin-Luther-Universität und denen der Abteilung Modellierung am Institut für Troposphärenforschung für das stets freundliche und angenehme Arbeitsklima danken. Des Weiteren danke ich der Deutschen Forschungsgemeinschaft für die Finanzierung im Rahmen des Schwerpunktprogramms MetStröm (Skalenübergreifende Modellierung in der Strömungsmechanik und Meteorologie) ohne die ich das Promotionsstudium gar nicht hätte durchführen können.

Ich bedanke mich bei OStR Arndt Pawelczik, der die vorliegende Arbeit auf sprachliche und grammatikalische Fehler untersucht hat. Außerdem danke ich OStD' Anni Schulz-Krause dafür, dass sie es mir trotz meiner neuen Tätigkeit ermöglicht hat, die Dissertation fertigzustellen.

Ich bin Ines Lichtenecker für die uneingeschränkte Unterstützung in allen Lebenslagen außerordentlich dankbar. Und last but not least: Die guten Freunde in Halle, Leipzig und Köln. Danke!

Zusammenfassung

Wir entwickeln neue Verfahren für die Lösung der kompressiblen Euler-Gleichungen, welche die zu Grunde liegenden Gleichungen bei der numerischen Wettervorhersage sind. Zu diesem Zweck untersuchen wir die zwei Hauptschwierigkeiten, die bei der Integration der kompressiblen Gleichungen auftreten.

Der erste Aspekt ist, dass Schallwellen auf Grund der Kompressibilität auftreten. Man ist an einer genauen Wiedergabe von physikalischen Prozessen wie Advektion und Schwerewellen interessiert, welche viel langsamer als Schallwellen sind. Wenn man ein explizites Verfahren benutzt, beschränkt also die Akustik auf Grund der CFL-Bedingung die maximale Zeitschrittweite. Um diese Einschränkung durch die meteorologisch unwichtigen Moden zu vermeiden, werden split-explizite Verfahren benutzt, d.h., die rechte Seite der Euler-Gleichungen wird in langsame und schnelle Prozesse aufgeteilt, die mit unterschiedlichen Verfahren integriert werden. Operator-Splitting ist in vielen operationellen Wetter-Modellen implementiert, aber die benutzten split-expliziten Methoden benötigen eine künstliche Dämpfung, um stabil zu sein. Wir haben ein split-explizites Verfahren der Ordnung 2 entwickelt, das auch ohne diese künstliche Dämpfung stabil ist, d.h., im Gegensatz zu den verbreiteten Modellen können wir die originalen Euler-Gleichungen mit einem split-expliziten Verfahren lösen, das große Zeitschritte benutzt.

Der zweite Aspekt ist die Implementierung der Orografie. Während alle operationellen Wetter-Modelle bodenfolgende Koordinaten benutzen, nutzen wir ein kartesisches Gitter bei dem die Orografie aus dem Gitter herausgeschnitten ist. Einerseits hat dieser Ansatz den Vorteil, dass keine künstlichen Kräfte wie bei bodenfolgenden Koordinaten auftreten, wo der Druckgradient in der Nähe von Bergen groß wird. Andererseits können beliebig kleine Zellen auftauchen, wenn man angeschnittene Zellen benutzt. Das führt auf Grund des CFL-Kriteriums zu strengen Zeitschrittbegrenzungen, falls man explizite Verfahren benutzt. Aber angeschnittene Zellen gibt es nur in einem kleinen Bereich des Rechengebiets. Deshalb haben wir partiell-implizite Verfahren entwickelt. In angeschnittenen Zellen enthält die Jacobimatrix alle Prozesse wie Advektion, Diffusion und Akustik, während in den vollen Zellen der freien Atmosphäre nur die Akustik in der Jacobimatrix benutzt wird. Somit sind diese Verfahren linear-implizit in den angeschnittenen und semi-explizit in den vollen Zellen. Sie können stabil mit Zeitschrittweiten rechnen, die durch das CFL-Kriterium in der freien Atmosphäre beschränkt sind. Des Weiteren benutzen wir in den angeschnittenen Zellen eine vereinfachte Jacobimatrix, um Speicherplatz und Rechenzeit zu sparen. Diese Verfahren sind in der freien Atmosphäre genauso stabil und genau wie das split-explizite Verfahren aber sie können außerdem fast ohne Mehraufwand mit angeschnittenen Zellen rechnen.

Die meisten der Ergebnisse, die in dieser Dissertation vorgestellt werden, sind in [13], [14] und [15] publiziert.

Abstract

We develop new methods for the solution of the compressible Euler equations which are the governing equations in numerical weather prediction. For this purpose we investigate the two main difficulties when integrating the compressible equations.

The first aspect is that sound waves occur as a consequence of the compressibility of the model. While one is interested in an accurate representation of physical processes like advection and gravity waves, these processes are much slower than sound waves. If an explicit method is used, the acoustics restricts the maximum time step size due to the CFL condition. In order to avoid this restriction from those meteorologically irrelevant modes split-explicit methods are used, i.e. the right-hand side of the Euler equations is split up into the slow and fast processes which are integrated with different methods. This operator splitting is implemented in several operational weather models but the existing split-explicit methods need artificial damping in order to be stable. We developed a second-order split-explicit method that is stable without any artificial damping, i.e. in contrast to the widely used models we can stably integrate the original compressible Euler equations with a split-explicit method which uses large time steps.

The second aspect is the implementation of orography. While all operational weather models use terrain-following coordinates, we use a Cartesian grid where the orography is cut out. On the one hand this ansatz has the advantage that no artificial forces occur as is the case with terrain-following coordinates because near mountains the pressure derivative becomes large. On the other hand arbitrary small cells can occur when using cut cells. This results in very harsh time step restrictions for explicit methods due to the CFL criterion. But cut cells only appear in a small region of the domain. Therefore we developed partially implicit methods. 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 methods are linearly implicit in the cut cell regions and semi-explicit in the free regions. They can stably compute with time step sizes restricted only by the CFL condition in the free atmosphere. In addition we use a simplified Jacobian in the cut cell regions in order to save memory and gain computational efficiency. In the free atmosphere these methods are as stable and accurate as the split-explicit method but furthermore they can compute with cut cells with nearly no additional effort.

Most of the results presented in this thesis are published in [13], [14] and [15].

Contents

1	Introduction	13
2	Formulation of the methods	18
2.1	Split-explicit Runge-Kutta methods	18
2.2	Split-explicit peer methods	19
2.3	Linearly implicit peer methods	20
3	Order theory	23
3.1	Order of consistency for split-explicit peer methods	23
3.2	Order of convergence for split-explicit peer methods	27
3.3	Order of consistency for linearly implicit peer methods	30
3.4	Order of convergence for linearly implicit peer methods	33
4	Stability investigations	38
4.1	Linearization of the Euler equations	38
4.2	Linear stability analysis	40
4.3	Stability properties of split-explicit methods	43
4.4	Stability properties of linearly implicit methods	45
4.5	Amplitude and phase properties	47
5	Implementation	51
5.1	The compressible Euler equations	51
5.2	Implementation with finite volumes	53
5.3	Computation of the Jacobian	55
5.4	Implementation of split-explicit methods	57
5.5	Initialization of orography	59
5.6	Method search	60
5.7	Coefficients of the methods	62
5.7.1	The split-explicit Runge-Kutta method RK3	62
5.7.2	The split-explicit peer method explPeer	63
5.7.3	The Rosenbrock method ROS3Pw	64
5.7.4	The linearly implicit peer method implPeer2	65
5.7.5	The linearly implicit peer method implPeer3	66
6	Numerical tests	67
6.1	Order test 1: Burgers' equation	67
6.2	Order test 2: Burgers' equation with acoustics	67
6.3	Order test 3: Rising bubble	68
6.4	Test case 1: Rising bubble	70
6.5	Test case 2: Density current	73
6.6	Test case 3: Flow over a mountain	76
6.7	Test case 4: Zeppelin test	79
6.8	Test case 5: Dam-break test	82
7	Conclusions	85

List of Figures

1	Orography modelled with terrain-following coordinates (top left) and cut cells (top middle). Pressure (top right) and cloud (bottom left) in a model with terrain-following coordinates. Mont Blanc modelled with resolutions of 7 km (bottom middle) and 1 km (bottom right), Mont Blanc images from [4]. . . .	15
2	Variables defined on a one-dimensional staggered grid.	40
3	Eigenvalues of M (left) and \widetilde{M} (right).	42
4	Stability diagrams for RK3-FB without (top left) and with (top right) divergence damping, for RK3-TR without (middle left) and with (middle right) off-centering and for explPeer-FB (bottom left) and explPeer-TR (bottom right). Unstable regions in grey with contour interval 0.1.	44
5	Stability diagrams for ROS3Pw (top), implPeer2 (middle) and implPeer3 (bottom) with partial Jacobian (left) and simplified Jacobian (right). Unstable regions in grey with contour interval 0.1.	46
6	Amplitude for the $4\Delta x$ wave (top), the $7\Delta x$ wave (middle) and the $10\Delta x$ wave (bottom) for split-explicit methods and linearly implicit methods with partial Jacobian (left) and for linearly implicit methods with simplified Jacobian (right). The analytic amplitude in black.	48
7	Relative phase speed for the $4\Delta x$ wave (top), the $7\Delta x$ wave (middle) and the $10\Delta x$ wave (bottom) for split-explicit methods and linearly implicit methods with partial Jacobian (left) and for linearly implicit methods with simplified Jacobian (right). The analytic relative phase speed in black.	49
8	The positions of the variables on the Arakawa C grid.	52
9	One cell with orography (left) and four cells (right) of the grid.	54
10	Pressure perturbation after 20 s (left) and initialized wind field (right).	60
11	Error of the numerical solution for the Burgers' equation (left) and for the Burgers' equation with acoustics (right) in L^2 norm against CFL number.	68
12	Error of the numerical solution for the rising bubble test in L^2 norm against CFL number (left) and CPU time (right).	69
13	Potential temperature after 0 s, 143 s, 286 s, 429 s, 572 s, 715 s, 858 s and 1000 s for the rising bubble test case.	71
14	Potential temperature after 1000 s for the rising bubble test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with partial (left) and simplified (right) Jacobian.	72
15	Potential temperature after 0 s, 180 s, 360 s, 540 s, 720 s and 900 s for the density current test case.	74
16	Potential temperature after 900 s for the density current test computed with (from top to bottom) RK3 and explPeer with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with partial Jacobian and implPeer2 with simplified Jacobian in cells with high wind speeds only.	75
17	Witch of Agnesi mountain with cut cells and vertical wind after 0 s, 360 s, 720 s, 1080 s, 1440 s, 1800 s and 2160 s for the flow over a mountain test case. Contour interval is 0.25 m s^{-1}	77

18	Vertical wind after 2160 s for the flow over a mountain test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with simplified Jacobian in cut cells only (left) and everywhere (right). Contour interval is 0.25 m s^{-1} . . .	78
19	Zeppelin obstacle with cut cells and potential temperature after 0 s, 208 s, 416 s, 624 s, 832 s, 1040 s and 1250 s for the zeppelin test case.	80
20	Potential temperature after 1250 s for the zeppelin test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with simplified Jacobian in cut cells only (left) and everywhere (right).	81
21	Depth after 0 s, 7 s, 14 s, 21 s, 28s, 35 s, 42 s and 50 s for the dam-break test case.	83
22	Depth after 50 s for the dam-break test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with simplified Jacobian in cut cells only (left) and everywhere (right).	84

List of Tables

1	Numbers of entries per grid cell of the different Jacobians.	55
2	Properties of the considered methods. s is the number of stages, p the order of convergence, $\ \alpha\ _1$ the sum of the fast integration intervals for split-explicit methods, C_{adv} and C_{sound} are the advection and acoustic stability limits for split-explicit methods and for linearly implicit methods with partial/simplified Jacobian under the restriction $\bar{u} < \frac{c_s}{6}$. Properties of RK3 with divergence damping respectively off-centering in parentheses. $n_s = 30$ for FB respectively $n_s = 1$ for TR.	63

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 the Courant-Friedrichs-Lewy (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.

A widespread method was introduced in [18] which uses a leapfrog time discretization for the slow part and the computationally very efficient forward-backward scheme (FB) from [5] for the high-frequency modes. In [29] divergence damping was introduced which stabilizes split-explicit schemes by damping acoustic modes. Then the time-splitting idea was improved in [44] by using a second-order two-stage Runge-Kutta method (RK2) for the low-frequency modes instead of the leapfrog method together with the same forward-backward scheme for the high-frequency modes and divergence damping for stabilization. It was shown that RK2 is stable and as accurate as the leapfrog method but computationally more efficient. In [45] the two-stage Runge-Kutta method was replaced by a three-stage Runge-Kutta method (RK3) in order to use this method together with higher-order spatial discretizations (RK2 does not run stably with even-order spatial discretizations) and to improve stability and accuracy. In [24] it was demonstrated that RK3 is second-order in time for nonlinear problems. These two split-explicit Runge-Kutta methods RK2 and RK3 by Wicker and Skamarock are widely used, e.g. in the Weather Research and Forecasting Model WRF of the National Center for Atmospheric Research NCAR ([28], [31]) and in the operational model COSMO provided by the German Weather Service DWD.

The disadvantage of these Runge-Kutta methods is the fact that the high-frequency modes still constrain the maximum time step size if no additional damping term such as divergence damping is used. In a linear test equation, even with the use of the analytical solution for the high-frequency modes (i.e. with an infinite number of small time steps), there is the constraint $c_s \Delta t / \Delta x < \pi$ where c_s is the speed of sound, Δt the time step size and Δx the spatial step size. In contrast the maximum CFL number of advection for RK3 is $\sqrt{3}$, which means that in case of maximum wind speeds below 190 m s^{-1} the acoustic modes constrain the maximum time step size independent of the number of small time steps per large time step. One approach to avoid this problem if one does not want to use an additional damping term is to use another fast integrator. We found out that using a method introduced in [27] removes the stability constraints caused by the acoustic modes if the number of small time steps per large time step is sufficiently large. On the other hand this fast integrator is at least twice as expensive as the forward-backward integrator and the scheme is stable only for large time steps, i.e. there is a region of medium time steps for which the scheme becomes unstable. Another approach was done in [16] where

RK3 was combined with the trapezoidal rule (TR) for the fast modes instead of the forward-backward scheme and the same time step sizes for both parts were used which is possible because the trapezoidal scheme is unconditionally stable. Tests with the shallow water equations in [16] showed that this approach is stable and only a little less accurate in comparison to RK3. But in [16] a rather simple test equation was used to determine stability. In contrast we use a more sophisticated test equation which shows that RK3 is not stable when TR is used as an integrator for the acoustics. In spite of the fact that it seems to be stable for the shallow water equations our tests with the compressible Euler equations verify our results from linear stability: RK3 with TR is not suitable for the application in atmospheric models. A possible ansatz to use RK3 anyhow with the guarantee of unconditional stability with respect to the acoustics, is to employ some off-centering to TR, i.e. to use a θ method. But then the order goes down to 1. In [43] the idea of split-explicit Runge-Kutta methods was generalized. The scheme also incorporates linear combinations of numerical solutions and the lengths of the integration intervals for the fast part is not determined by the nodes of the underlying Runge-Kutta method. These generalizations result in a method which has a CFL condition with respect to acoustics which is twice as large as the condition for RK3, if no artificial damping is used. That ansatz is quite similar to our ansatz but in contrast to the method published in [43], which is third-order and still has a limitation from the acoustics, we constructed a second-order method that has no limitations from acoustics with that ansatz (not presented in this thesis).

The split-explicit method which we present in this thesis is stable without the use of any artificial damping. It can use the computationally very efficient forward-backward scheme with small time steps as integrator for the high-frequency modes like Klemp, Wilhelmson, Wicker and Skamarock did before. The scheme is also stable when the trapezoidal rule is used with the same time step size as for the advection. In contrast to [44], [45] and [43] there is no limitation for the time step size of the split-explicit method which arises from the acoustics if TR or FB with sufficiently small time steps is used as fast integrator, only the advection determines the maximum time step size. The underlying method comes from the class of peer methods which is a subclass of general linear methods. General linear methods were introduced in [3], good overviews can be found in [9] and [12]. 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 everyone of the stages of a peer method is a linear multi-step method. The new feature of peer methods is that they possess several stages like Runge-Kutta methods, but all of these stages have the same properties, no extraordinary solution variable is used. These methods combine positive features of both, Runge-Kutta and linear multi-step methods. They have the same order in every stage so they have no order reduction even for very stiff systems and are implemented as one option for integration methods in the KARDOS code ([8]). Because of the generality of this class of methods they proved to be applicable to many different kinds of problems. Linearly-implicit peer methods were introduced in [41], their order does not depend on what is used as Jacobian. Explicit peer methods were successfully used in [42] and [26] amongst others. Since this is common for weather codes we consider peer methods with constant time step sizes. The split-explicit

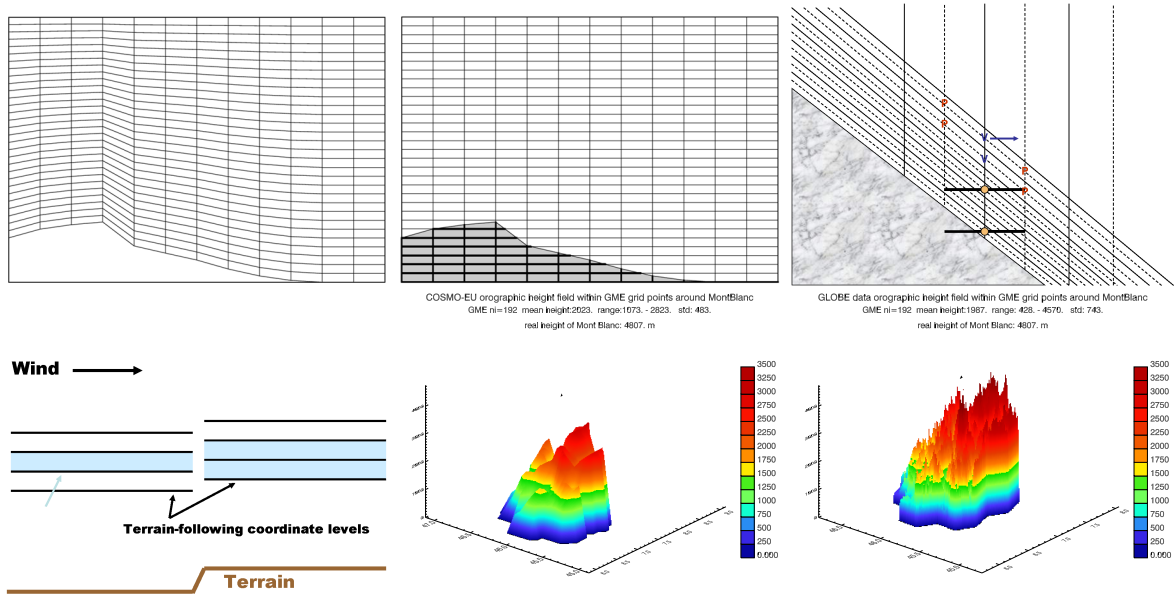


Figure 1: Orography modelled with terrain-following coordinates (top left) and cut cells (top middle). Pressure (top right) and cloud (bottom left) in a model with terrain-following coordinates. Mont Blanc modelled with resolutions of 7 km (bottom middle) and 1 km (bottom right), Mont Blanc images from [4].

peer method, which we use for the integration of the compressible Euler equations, was introduced in [13].

For the representation of orography terrain-following coordinates are established in operational weather models. Figure 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 terrain-following coordinates can cause anomalous vertical dispersion. This is illustrated by Figure 1: The horizontal wind should transport the cloud to the right. But the different levels of the terrain-following grid over the mountain force the cloud to distribute over two cells, i.e. the density of the cloud decreases only because of the grid structure. Because the volume which is occupied by the cloud increases the number of aerosols in the cloud increases. Aerosols act as condensation nuclei in clouds. The increased number of aerosols in the cloud results in the formation of more but smaller droplets. This artificial behaviour induced by the grid can prevent precipitation or forces the cloud to vanish due to the smaller density of the water vapour. Another disadvantage of terrain-following coordinates is that the stratified atmosphere generates artificial forces due to the grid: The horizontal pressure derivative in a vertically stratified atmosphere is zero, i.e. it is small in 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 Figure 1. As proposed in [34] 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 ([36]). In the course of time the spatial resolutions of numerical weather prediction models increase so that terrain will be resolved better in the future which will result in steeper gradients. This is illustrated by Figure 1, Mont Blanc is modelled with the resolution of the COSMO-EU model which is used by the DWD for the operative weather prediction over Europe and with a three times higher resolution than the DWD uses for the weather prediction over Germany. So this high resolution might be used in some years' time for numerical weather prediction. As can be seen Mont Blanc is much smoother with the COSMO-EU resolution but despite that coarse resolution Mont Blanc has steep gradients. With the high resolution the gradients become even steeper and the described problems with terrain-following coordinates 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 thesis. 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 ([34]). For these reasons the DWD developed a cut cell model called LM-Z besides its operational model. The Regional Atmospheric Modeling System RAMS ([22]) originally used terrain-following coordinates and a cut cells option was added later. The Ocean-Land-Atmosphere Model OLAM ([38]) which was developed from RAMS uses cut cells only. The All Scale Atmospheric Model ASAM ([11]) 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 size is restricted by the CFL condition for these small cells. A similar problem occurs when computing on the sphere with a latitude-longitude grid because of the pole singularities. One ansatz to avoid this problem is filtering as done in [34], [7], [17] and in OLAM. In contrast to that ansatz we integrate the compressible equations without any filtering. For this purpose we use partially implicit peer methods. 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 if cut cells are present. The presented class of partially implicit peer methods in this thesis 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 of partially implicit methods is that the Jacobian is only used where it is necessary for stability, i.e. they are a mixture of semi-explicit and linearly implicit methods. Because we 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 use a Jacobian which uses a lower-order spatial discretization of advection in order to save memory and gain computational efficiency. Because we use peer methods which retain the order independently of what is used as Jacobian, 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 methods but they require a careful consideration of the linear stability theory in order to derive methods which are stable despite these simplifications. One of the derived partially implicit peer methods was presented in [15].

To sum up, this thesis describes how the two main problems when integrating the compressible Euler equations are solved: The first problem, the appearance of fast sound waves which should not restrict the maximum time step size, is solved with a split-explicit peer method which can stably integrate the compressible equations without any artificial damping. The second problem of arbitrary small cells due to the representation of orography with cut cells is solved by using partially implicit peer methods which are as accurate and efficient as the split-explicit peer method because they use the full Jacobian in cut cells only, while in the remaining region the Jacobian only incorporates acoustics. This thesis is organized as follows:

In Section 2 we formulate the classes of split-explicit and linearly implicit peer methods.

In Section 3 we derive the order conditions for the considered peer methods with constant time step sizes.

In Section 4 we describe the linear stability theory and show the amplitude and phase properties of the presented peer methods and some other methods for comparison.

In Section 5 we give detailed information on how the compressible Euler equations are implemented.

Three order tests and five numerical tests with the compressible Euler equations and the shallow water equations are presented in Section 6.

Finally we give some conclusions in Section 7.

2 Formulation of the methods

2.1 Split-explicit Runge-Kutta methods

In order to make the idea of split-explicit methods easier to understand and because the lack of stability of split-explicit Runge-Kutta methods is our main motivation for the consideration of split-explicit peer methods, we consider split-explicit Runge-Kutta methods as a preliminary step to the introduction of split-explicit peer methods. With these we want to solve autonomous split differential equations

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

where N represents the nonlinear slow part and L the linear fast part. We introduce split-explicit Runge-Kutta methods in a more technical way than Wicker and Skamarock did in [44] and [45] because we do not restrict ourselves to one integrator for the fast part of (1) but our formulation of the split-explicit scheme contains the fast part as an initial value problem. With this view we can focus on the properties of the main solver for the slow part, i.e. on the underlying Runge-Kutta and peer methods. Furthermore this approach will simplify the derivation of order conditions for split-explicit peer methods.

To solve split differential equation (1) we propose a scheme where an explicit Runge-Kutta method is used for the slow part N of the right-hand side while the solution of the linear ordinary differential equation with a constant slow part, i.e. $\dot{y} = C + Ly$, is defined implicitly by a differential equation so that in the absence of L the explicit Runge-Kutta method is recovered. For simplicity of notation we will denote the weights b_i as the $s + 1$ -st row of A from the Butcher tableau, e.g. for RK3 the Butcher tableau is

$$\begin{array}{c|ccc} 0 & & & \\ 1/3 & 1/3 & & \\ 1/2 & 0 & 1/2 & \\ \hline & 0 & 0 & 1 \end{array}$$

and therefore we have $a_{ij} = 0$ except $a_{21} = 1/3$, $a_{32} = 1/2$ and $a_{43} = 1$. The nodes $c_1 = 0$, $c_2 = 1/3$, $c_3 = 1/2$ and $c_4 = 1$ are the points on the time axis where the numerical solution approximates the analytical solution, Δt is the time step size. The i th stage (for $2 \leq i \leq s + 1$ with $y_m^{(1)} := y_{m-1}^{(s+1)}$) of a split-explicit Runge-Kutta method is then:

$$\text{Solve} \quad \dot{Z}_i = \frac{1}{c_i} \sum_{j=1}^{i-1} a_{ij} N(y_m^{(j)}) + LZ_i \quad (2)$$

$$\text{with initial value} \quad Z_i(0) = y_{m-1}^{(s+1)} \quad (3)$$

$$\text{and define} \quad y_m^{(i)} = Z_i(c_i \Delta t). \quad (4)$$

The considered split-explicit Runge-Kutta methods in [44] and [45] fit into this formulation if the forward-backward scheme FB is used for the integration of (2) as will be explained in detail in Section 5.4. Another approach is used in [19] where the integration of (2) does not start at $y_{m-1}^{(s+1)}$ but at $y_m^{(i-1)}$ for the i th stage. This approach has the

advantage that the sum of the integration intervals is not $\sum c_i$ but only 1 in case of non-negative ascending nodes. Therefore it is computationally more efficient but on the other hand it has worse stability properties. However, we will focus on the generalization of the Wicker and Skamarock approach in the remainder. As we will see in Section 4.3 the use of Runge-Kutta methods with this ansatz results in split-explicit methods which need artificial damping like diffusion to compute stably. This is the main motivation for considering peer methods as underlying methods for the slow part of split differential equations in order to obtain a stable method which does not need any additional damping.

2.2 Split-explicit peer methods

Explicit two-step peer methods for first-order differential equations $\dot{y} = f(y)$ were considered in [1], [39] and [42], in parallel form also in [40] and [26]. In this thesis we also imply values of the numerical solution from the current time step. For constant step sizes these methods are given by

$$Y_m = BY_{m-1} + SY_m + \Delta t AF_{m-1} + \Delta t RF_m \quad (5)$$

with the notations

$$\begin{aligned} Y_m &:= (Y_{mi})_{i=1}^s \in \mathbb{R}^{s \times n} & \text{with} & & Y_{mi} &\approx y(t_{mi}) = y(t_m + c_i \Delta t), \\ F_m &:= \left(f(Y_{mi}) \right)_{i=1}^s \in \mathbb{R}^{s \times n} \end{aligned}$$

and $B, S, A, R \in \mathbb{R}^{s \times s}$ where S and R are strictly lower triangular matrices to obtain explicit methods.

Remark 1 *For the integration of $\dot{y} = f(y)$ the use of the numerical solution from the current time level, i.e. $S \neq 0$, does not lead to a more general class of peer methods than the case $S = 0$ because every peer method with $S \neq 0$ can be generated from a peer method with $S = 0$. This can be seen by subtracting SY_m from (5) and then multiplying both sides with $(I - S)^{-1}$. Therefore in former applications of peer methods $S = 0$ was used. However, because of the following splitting approach the use of $S \neq 0$ leads to a more general class of split-explicit peer methods and therefore we use these additional degrees of freedom.*

To solve split differential equation (1) we propose a scheme where the explicit peer method is used for the slow part N while the solution of $\dot{y} = C + Ly$ is defined implicitly by a differential equation so that in the absence of L the explicit peer method is recovered. A further generalization is done by allowing the lengths of the integration intervals α_i to be independent of the nodes c_i in contrast to split-explicit Runge-Kutta methods where

$\alpha_i = c_i$. The i th stage (for $1 \leq i \leq s$) of such a method reads:

$$\text{Solve} \quad \dot{Z}_i = \frac{1}{\alpha_i} \left(\sum_{j=1}^s a_{ij} N(Y_{m-1,j}) + \sum_{j=1}^{i-1} r_{ij} N(Y_{mj}) \right) + LZ_i \quad (6)$$

$$\text{with initial value} \quad Z_i(0) = \sum_{j=1}^s b_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} s_{ij} Y_{mj} \quad (7)$$

$$\text{and define} \quad Y_{mi} = Z_i(\alpha_i \Delta t). \quad (8)$$

In case of $L = 0$ such a method obviously is an explicit peer method with $B = (b_{ij})$, $S = (s_{ij})$, $A = (a_{ij})$ and $R = (r_{ij})$.

Remark 2 *Every split-explicit Runge-Kutta method can be written as a split-explicit peer method. For example RK3 reads*

$$c = \alpha = \begin{pmatrix} 1/3 \\ 1/2 \\ 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad S = 0, \quad A = \begin{pmatrix} 0 & 0 & 1/3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

as a peer method. The split-explicit scheme with the leapfrog method for the slow part, used in [29] and [30], also can be written as a peer method with

$$c = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix}, \quad \alpha = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad S = 0, \quad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

The same holds for the generalized split-explicit Runge-Kutta methods from [43], they are a subclass of split-explicit peer methods, too.

2.3 Linearly implicit peer methods

Linearly implicit peer methods were discussed in [41], [23] and [8] amongst others. With them we want to solve autonomous first-order differential equations

$$\dot{y} = f(y), \quad t \in [t_0, t_e], \quad y(t_0) = y_0 \in \mathbb{R}^n. \quad (9)$$

For the derivation of linearly implicit peer methods we start with explicit peer methods and add the function evaluation of the solution from the current stage:

$$Y_{mi} = \sum_{j=1}^s \tilde{b}_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} \tilde{s}_{ij} Y_{mj} + \Delta t \sum_{j=1}^s \tilde{a}_{ij} f(Y_{m-1,j}) + \Delta t \sum_{j=1}^{i-1} \tilde{r}_{ij} f(Y_{mj}) + \Delta t \gamma f(Y_{mi}).$$

Because we consider singly implicit methods, i.e. use the same γ in every stage, we add no index to it. We solve the occurring nonlinear system of equations with one Newton step, i.e. we apply Newton's method to $g(y) = 0$ with

$$g(y) = y - \Delta t \gamma f(y) - \sum_{j=1}^s \tilde{b}_{ij} Y_{m-1,j} - \sum_{j=1}^{i-1} \tilde{s}_{ij} Y_{mj} - \Delta t \sum_{j=1}^s \tilde{a}_{ij} f(Y_{m-1,j}) - \Delta t \sum_{j=1}^{i-1} \tilde{r}_{ij} f(Y_{mj}),$$

$$g'(y) = I - \Delta t \gamma J$$

where $J = f'(y)$ and I is the identity matrix. We have to solve the linear system

$$g'(Y_{mi}^{start})(Y_{mi} - Y_{mi}^{start}) = -g(Y_{mi}^{start}),$$

for the starting value we use

$$Y_{mi}^{start} := -\frac{1}{\gamma} \left(\sum_{j=1}^s g_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} h_{ij} Y_{mj} \right).$$

Instead of using the function evaluation and Jacobian of this starting value we use

$$f(Y_{mi}^{start}) = \frac{1}{\Delta t \gamma} \left(\sum_{j=1}^s \hat{b}_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} \hat{s}_{ij} Y_{mj} + \Delta t \sum_{j=1}^s \hat{a}_{ij} f(Y_{m-1,j}) + \Delta t \sum_{j=1}^{i-1} \hat{r}_{ij} f(Y_{mj}) \right),$$

$$J = f'(y(t_m)).$$

These choices lead to the equation

$$(I - \Delta t \gamma J) \left(Y_{mi} + \frac{1}{\gamma} \left(\sum_{j=1}^s g_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} h_{ij} Y_{mj} \right) \right) = \frac{1}{\gamma} \left(\sum_{j=1}^s g_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} h_{ij} Y_{mj} \right)$$

$$+ \sum_{j=1}^s (\tilde{b}_{ij} + \hat{b}_{ij}) Y_{m-1,j} + \sum_{j=1}^{i-1} (\tilde{s}_{ij} + \hat{s}_{ij}) Y_{mj} + \Delta t \sum_{j=1}^s (\tilde{a}_{ij} + \hat{a}_{ij}) f(Y_{m-1,j}) + \Delta t \sum_{j=1}^{i-1} (\tilde{r}_{ij} + \hat{r}_{ij}) f(Y_{mj}).$$

With $b_{ij} := \tilde{b}_{ij} + \hat{b}_{ij}$, $s_{ij} := \tilde{s}_{ij} + \hat{s}_{ij}$, $a_{ij} := \tilde{a}_{ij} + \hat{a}_{ij}$ and $r_{ij} := \tilde{r}_{ij} + \hat{r}_{ij}$ we finally obtain the considered class of linearly implicit peer methods

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

$$+ \Delta t \sum_{j=1}^s g_{ij} J Y_{m-1,j} + \Delta t \sum_{j=1}^{i-1} h_{ij} J Y_{mj}.$$

In compact matrix notation the considered methods read

$$Y_m (I - \Delta t \gamma J)^T = B Y_{m-1} + S Y_m + \Delta t A F_{m-1} + \Delta t R F_m + \Delta t G Y_{m-1} J^T + \Delta t H Y_m J^T \quad (10)$$

with $B = (b_{ij})$, $S = (s_{ij})$, $A = (a_{ij})$, $R = (r_{ij})$, $G = (g_{ij})$ and $H = (h_{ij})$.

Remark 3 While we use formulation (10) for the derivation of order conditions and stability theory the methods are implemented in the equivalent formulation

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

$$Y_{mi} = \hat{Y}_{mi} - \sum_{j=1}^s \frac{g_{ij}}{\gamma} Y_{m-1,j} - \sum_{j=1}^{i-1} \frac{h_{ij}}{\gamma} Y_{mj}$$

in order to avoid the matrix-vector multiplications with the Jacobian in (10).

Remark 4 *When applying a split-explicit peer method to the split differential equation (1) and using the trapezoidal rule for the integration of the occurring differential equations this results in*

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

where $f(y) = N(y) + Ly$, i.e. a 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 we will see in Section 4.3 the presented split-explicit peer method is suitable for the integration of the compressible Euler equations in the free atmosphere. But when cut cells occur it is not sufficiently stable even if we use the above formulation with the Jacobian incorporating not only acoustics but also advection. This and the fact that this formulation is not singly implicit are the main reasons why we use the split-explicit peer method as underlying method and only change the implicit coefficients γ , g_{ij} and h_{ij} in order to derive a linearly implicit peer method which has the same good properties of the split-explicit peer method in the free atmosphere but furthermore it is stable in the cut cell regions, too.

3 Order theory

3.1 Order of consistency for split-explicit peer methods

We now derive order conditions for split-explicit peer methods. Consistency of split-explicit peer methods is discussed by considering the local residuals obtained by substituting the analytical solution into the method, i.e. we consider

$$\text{Solve} \quad \dot{\tilde{Z}}_i = \frac{1}{\alpha_i} \left(\sum_{j=1}^s a_{ij} N(y(t_{m-1,j})) + \sum_{j=1}^{i-1} r_{ij} N(y(t_{m_j})) \right) + L \tilde{Z}_i \quad (11)$$

$$\text{with initial value} \quad \tilde{Z}_i(0) = \sum_{j=1}^s b_{ij} y(t_{m-1,j}) + \sum_{j=1}^{i-1} s_{ij} y(t_{m_j}) \quad (12)$$

where y is the analytical solution. We are interested only in the error of the slow part of the split differential equation (1) and therefore we assume that we can solve (11) exactly. Then the residuals are

$$\Delta_{mi} := y(t_{mi}) - \tilde{Z}_i(\alpha_i \Delta t), \quad i = 1, \dots, s.$$

We define the order of consistency of a split-explicit peer method by the order of the local residuals:

Definition 1 *A split-explicit peer method (8) has order of consistency p if*

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

Because a peer method has the same order in every stage the derivation of order conditions is much easier than for split-explicit Runge-Kutta methods. On the other hand the splitting ansatz makes the derivation of order conditions more complicated than for ordinary peer methods. Furthermore our splitting ansatz only allows the construction of reasonable methods of order 2.

Theorem 1 *A split-explicit peer method (8) has order of consistency $p \leq 2$ if*

$$\begin{aligned} AB(k) &= 0, & k &= 0, \dots, p, \\ \widehat{AB}(k) &= 0, & k &= 1, \dots, p \end{aligned}$$

with

$$\begin{aligned} AB_i(k) &:= c_i^k - \sum_{j=1}^s b_{ij} (c_j - 1)^k - \sum_{j=1}^{i-1} s_{ij} c_j^k - k \sum_{j=1}^s a_{ij} (c_j - 1)^{k-1} - k \sum_{j=1}^{i-1} r_{ij} c_j^{k-1}, \\ \widehat{AB}_i(k) &:= (c_i - \alpha_i)^k - \sum_{j=1}^s b_{ij} (c_j - 1)^k - \sum_{j=1}^{i-1} s_{ij} c_j^k \quad \text{for } i = 1, \dots, s. \end{aligned}$$

The condition $\alpha_i = 0$ for $i = 1, \dots, s$ is necessary for order 3.

Proof: For simplicity of notation we consider only values from the current time step, i.e. s_{ij} and r_{ij} . The inclusion of values from the other time level, i.e. terms with b_{ij} and a_{ij} , is straightforward. For $\tilde{Z}_i(t)$ we obtain the expansion

$$\tilde{Z}_i(\alpha_i \Delta t) = \tilde{Z}_i(0) + \alpha_i \Delta t \dot{\tilde{Z}}_i(0) + \frac{(\alpha_i \Delta t)^2}{2} \ddot{\tilde{Z}}_i(0) + \frac{(\alpha_i \Delta t)^3}{6} \overset{\cdot\cdot\cdot}{\tilde{Z}}_i(0) + \mathcal{O}((\Delta t)^4). \quad (13)$$

In order to get shorter equations we use the abbreviations $\underline{y} := y(t_m)$, $\dot{\underline{y}} := \dot{y}(t_m)$, $\underline{N} := N(y(t_m))$, $\underline{N}' := N'(y(t_m))$, etc. Because for the definition of split-explicit peer methods the slow part of the ordinary differential equation (6) is constant it holds

$$\begin{aligned} \dot{\tilde{Z}}_i &= C + L\tilde{Z}_i & \text{with } C &= \frac{1}{\alpha_i} \left(\sum_{j=1}^{i-1} r_{ij} N(y(t_{mj})) \right), \\ \ddot{\tilde{Z}}_i &= L\dot{\tilde{Z}}_i, \\ \overset{\cdot\cdot\cdot}{\tilde{Z}}_i &= L^2 \dot{\tilde{Z}}_i \end{aligned}$$

The expansion of \tilde{Z}_i at 0 is

$$\begin{aligned} \tilde{Z}_i(0) &= \sum s_{ij} y(t_{mj}) \\ &= \sum s_{ij} \underline{y} + \Delta t \sum s_{ij} c_j \dot{\underline{y}} + (\Delta t)^2 \sum s_{ij} \frac{c_j^2}{2} \ddot{\underline{y}} + (\Delta t)^3 \sum s_{ij} \frac{c_j^3}{6} \overset{\cdot\cdot\cdot}{\underline{y}} + \mathcal{O}((\Delta t)^4). \end{aligned}$$

The first-order term in (13) fulfils

$$\begin{aligned} \alpha_i \Delta t \dot{\tilde{Z}}_i(0) &= \Delta t \sum r_{ij} N(y(t_{mj})) + \alpha_i \Delta t L \tilde{Z}_i(0) \\ &= \Delta t \sum r_{ij} \left(\underline{N} + \underline{N}'(y(t_{mj}) - \underline{y}) + \frac{1}{2} \underline{N}''(y(t_{mj}) - \underline{y})^2 \right) + \alpha_i \Delta t L \sum s_{ij} y(t_{mj}) \\ &\quad + \mathcal{O}((\Delta t)^4) \\ &= \Delta t \sum r_{ij} \left(\underline{N} + \underline{N}'(c_j \Delta t \dot{\underline{y}} + \frac{(c_j \Delta t)^2}{2} \ddot{\underline{y}}) + \frac{1}{2} \underline{N}''(c_j \Delta t \dot{\underline{y}})^2 \right) \\ &\quad + \alpha_i \Delta t L \sum s_{ij} \left(\underline{y} + c_j \Delta t \dot{\underline{y}} + \frac{(c_j \Delta t)^2}{2} \ddot{\underline{y}} \right) + \mathcal{O}((\Delta t)^4) \\ &= \Delta t \left(\sum r_{ij} \underline{N} + \alpha_i \sum s_{ij} L \underline{y} \right) + (\Delta t)^2 \left(\sum r_{ij} c_j \underline{N}' \dot{\underline{y}} + \alpha_i \sum s_{ij} c_j L \dot{\underline{y}} \right) \\ &\quad + (\Delta t)^3 \left(\sum r_{ij} \frac{c_j^2}{2} \underline{N}'' \ddot{\underline{y}} + \sum r_{ij} \frac{c_j^2}{2} \underline{N}'' \dot{\underline{y}}^2 + \alpha_i \sum s_{ij} \frac{c_j^2}{2} L \ddot{\underline{y}} \right) + \mathcal{O}((\Delta t)^4). \end{aligned}$$

For the second-order term in (13) it holds

$$\begin{aligned} \frac{(\alpha_i \Delta t)^2}{2} \ddot{\tilde{Z}}_i(0) &= \frac{\alpha_i \Delta t}{2} L \left(\alpha_i \Delta t \dot{\tilde{Z}}_i(0) \right) \\ &= (\Delta t)^2 \left(\frac{\alpha_i}{2} \sum r_{ij} L \underline{N} + \frac{\alpha_i^2}{2} \sum s_{ij} L^2 \underline{y} \right) \\ &\quad + (\Delta t)^3 \left(\alpha_i \sum r_{ij} \frac{c_j}{2} L \underline{N}' \dot{\underline{y}} + \alpha_i^2 \sum s_{ij} \frac{c_j}{2} L^2 \dot{\underline{y}} \right) + \mathcal{O}((\Delta t)^4). \end{aligned}$$

The third-order term in (13) satisfies

$$\begin{aligned} \frac{(\alpha_i \Delta t)^3}{6} \ddot{\tilde{Z}}_i(0) &= \frac{(\alpha_i \Delta t)^2}{6} L^2 \left(\alpha_i \Delta t \dot{\tilde{Z}}_i(0) \right) \\ &= (\Delta t)^3 \left(\frac{\alpha_i^2}{6} \sum r_{ij} L^2 \underline{N} + \frac{\alpha_i^3}{6} \sum s_{ij} L^3 \underline{y} \right) + \mathcal{O}((\Delta t)^4). \end{aligned}$$

The Taylor expansion of the analytical solution at t_m is

$$y(t_{mi}) = \underline{y} + c_i \Delta t \dot{\underline{y}} + \frac{(c_i \Delta t)^2}{2} \ddot{\underline{y}} + \frac{(c_i \Delta t)^3}{6} \dot{\ddot{\underline{y}}} + \mathcal{O}((\Delta t)^4). \quad (14)$$

As the above expansions show the preconsistency condition is

$$\sum s_{ij} = 1.$$

This is $AB(0)$.

The difference in the first-order terms between (13) and (14) is

$$c_i \dot{\underline{y}} - \sum s_{ij} c_j \dot{\underline{y}} - \sum r_{ij} \underline{N} - \alpha_i \sum s_{ij} L \underline{y}.$$

Because y is the analytic solution it satisfies $\dot{\underline{y}} = \underline{N} + L \underline{y}$. Substituting this into the above difference and using the preconsistency condition $AB(0)$ results in

$$c_i \underline{N} + c_i L \underline{y} - \sum s_{ij} c_j \underline{N} - \sum s_{ij} c_j L \underline{y} - \sum r_{ij} \underline{N} - \alpha_i L \underline{y},$$

i.e. from the terms incorporating the slow part \underline{N} the condition $AB(1)$ follows while the terms with the fast part $L \underline{y}$ vanish if $\widehat{AB}(1)$ is fulfilled.

The difference of the second-order terms in (13) and (14) is

$$\frac{c_i^2}{2} \ddot{\underline{y}} - \sum s_{ij} \frac{c_j^2}{2} \ddot{\underline{y}} - \sum r_{ij} c_j \underline{N}' \dot{\underline{y}} - \alpha_i \sum s_{ij} c_j L \dot{\underline{y}} - \frac{\alpha_i}{2} \sum r_{ij} L \underline{N} - \frac{\alpha_i^2}{2} \sum s_{ij} L^2 \underline{y}.$$

If we substitute $\ddot{\underline{y}}$ by $\underline{N}' \dot{\underline{y}} + L \dot{\underline{y}}$ the terms incorporating $\underline{N}' \dot{\underline{y}}$ vanish if $AB(2)$ is satisfied. It remains

$$\frac{c_i^2}{2} L \dot{\underline{y}} - \sum s_{ij} \frac{c_j^2}{2} L \dot{\underline{y}} - \alpha_i \sum s_{ij} c_j L \dot{\underline{y}} - \frac{\alpha_i}{2} \sum r_{ij} L \underline{N} - \frac{\alpha_i^2}{2} \sum s_{ij} L^2 \underline{y}.$$

If the order conditions $AB(0)$ and $\widehat{AB}(1)$ are fulfilled and because of $L \dot{\underline{y}} = L \underline{N} + L^2 \underline{y}$ the above difference is equivalent to

$$\frac{c_i^2}{2} L \dot{\underline{y}} - \sum s_{ij} \frac{c_j^2}{2} L \dot{\underline{y}} - \alpha_i \sum s_{ij} c_j L \dot{\underline{y}} - \frac{\alpha_i^2}{2} L \dot{\underline{y}}.$$

Together with $AB(1)$ this results in the order condition $\widehat{AB}(2)$.

The difference in the third-order terms between (13) and (14) is

$$\begin{aligned} \frac{c_i^3}{6} \dot{\ddot{\underline{y}}} - \sum s_{ij} \frac{c_j^3}{6} \dot{\ddot{\underline{y}}} - \sum r_{ij} \frac{c_j^2}{2} \underline{N}' \ddot{\underline{y}} - \sum r_{ij} \frac{c_j^2}{2} \underline{N}'' \dot{\underline{y}}^2 - \alpha_i \sum s_{ij} \frac{c_j^2}{2} L \ddot{\underline{y}} - \alpha_i \sum r_{ij} \frac{c_j}{2} L \underline{N}' \dot{\underline{y}} \\ - \alpha_i^2 \sum s_{ij} \frac{c_j}{2} L^2 \dot{\underline{y}} - \frac{\alpha_i^2}{6} \sum r_{ij} L^2 \underline{N} - \frac{\alpha_i^3}{6} \sum s_{ij} L^3 \underline{y}. \end{aligned}$$

If we substitute $\dot{\underline{y}}$ by $\underline{N}''\underline{y}^2 + \underline{N}'\underline{y} + \underline{L}\underline{y}$ all terms incorporating $\underline{N}''\underline{y}^2$ and $\underline{N}'\underline{y}$ vanish if $AB(3)$ is satisfied. With $AB(0)$ and $\widehat{AB}(1)$ the last two terms in the above difference simplify to $\frac{\alpha_i^3}{6} \sum L^2 \underline{y}$, i.e. it remains

$$\frac{c_i^3}{6} L \underline{y} - \sum s_{ij} \frac{c_j^3}{6} L \underline{y} - \alpha_i \sum s_{ij} \frac{c_j^2}{2} L \underline{y} - \alpha_i \sum r_{ij} \frac{c_j}{2} L \underline{N}' \underline{y} - \alpha_i^2 \sum s_{ij} \frac{c_j}{2} L^2 \underline{y} - \frac{\alpha_i^3}{6} L^2 \underline{y}.$$

When substituting $L \underline{y}$ by $L \underline{N}' \underline{y} + L^2 \underline{y}$ the terms with $L^2 \underline{y}$ vanish if $\widehat{AB}(3)$ is fulfilled. The remaining terms belong to $L \underline{N}' \underline{y}$, i.e. a third-order split-explicit peer method has to satisfy

$$\frac{c_i^3}{6} - \sum s_{ij} \frac{c_j^3}{6} - \alpha_i \sum s_{ij} \frac{c_j^2}{2} - \alpha_i \sum r_{ij} \frac{c_j}{2} = 0.$$

By computation one can see that this condition together with $AB(2)$, $\widehat{AB}(2)$ and $\widehat{AB}(3)$ is equivalent to $\alpha_i = 0$. \blacksquare

Remark 5 *From the stability point of view a method with $\alpha_i = 0$ is not reasonable, i.e. it is not possible to construct stable third-order split-explicit peer methods with the ansatz (7) because of the order condition $\alpha = 0$. This is caused by the absence of function evaluations in (7), the inclusion of function evaluations allows one to solve the order conditions for order 3 with $\alpha \neq 0$ and therefore the construction of split-explicit peer methods with order 3 and higher. However, we are interested in methods with order 2 because we use low-order integrators for the fast part and because RK3 is second-order for nonlinear problems. Therefore we consider the simpler case where no function evaluations are used as initial values for the fast ordinary differential equation (6) in the remainder.*

The order conditions $AB(k) = 0$ are the traditional order conditions for peer methods while the conditions $\widehat{AB}(k) = 0$ are new order conditions which result from the splitting approach. We now want to solve the order conditions to obtain three-stage split-explicit peer methods which are of order $p = s - 1 = 2$. For this aim we use the coefficient matrix A to fulfil the traditional order conditions as was done in [1], [40], [39], [42] and [26]. Because we only want methods of order $s - 1$ A has s degrees of freedom remaining which are expressed with a vector β . The matrix B is used to fulfil the consistency condition $(B + S)\mathbf{1} = \mathbf{1}$, where $\mathbf{1} = (1, \dots, 1)^T$, and the $s - 1$ additional split order conditions. Therefore B is fully determined by the order conditions in contrast to former applications of peer methods where the coefficients of B were available for stability optimizations or satisfaction of superconvergence conditions.

With the notations

$$C := \text{diag}(c_i), \quad D := \text{diag}(1, \dots, s), \quad V_n := \left((c_i - n)^{j-1} \right)_{i,j=1}^s, \quad V := \left((c_i - \alpha_i)^{j-1} \right)_{i,j=1}^s$$

one obtains methods with order of consistency $p = s - 1 \leq 2$ with

$$B = (V - SV_0)V_1^{-1},$$

$$A = \left(CV_0 - B(C - I)V_1 - SCV_0 - RV_0D - \beta e_s^T \right) D^{-1}V_1^{-1}.$$

This choice of parameters is possible if the nodes are distinct from each other because then the Vandermonde matrix V_1 is regular. Because we choose $c_s = 1$ so that $Y_{ms} \approx y(t_{m+1})$ the remaining parameters are $c_1, c_2, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, s_{21}, s_{31}, s_{32}, r_{21}, r_{31}$ and r_{32} . These will be optimized with respect to stability as explained in Section 5.6.

3.2 Order of convergence for split-explicit peer methods

We now show the convergence of split-explicit peer methods. When a split-explicit peer method is applied to (1) with $N = L = 0$ it holds $Y_{mi} = \sum b_{ij}Y_{m-1,j} + \sum s_{ij}Y_{mj}$ or in compact form $Y_m = BY_{m-1} + SY_m$ which is equivalent to $Y_m = (I - S)^{-1}BY_{m-1}$. This leads us to the definition of zero stability for split-explicit peer methods which is analogously defined as for general linear methods in [9].

Definition 2 A split-explicit peer method (8) is called zero stable if the eigenvalues λ of $\tilde{B} := (I - S)^{-1}B$ satisfy the conditions

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

Definition 3 A split-explicit peer method (8) 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

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

Before we show convergence we need an equation which we prove by induction.

Theorem 2 For all $m \in \mathbb{N}$ it holds

$$1 + \sum_{j=0}^m X_j \prod_{k=0}^{j-1} (1 + X_k) = \prod_{k=0}^m (1 + X_k). \quad (15)$$

Proof: For $m = 0$ the equation obviously is correct. Assuming that it is valid for $m - 1$. For m we have

$$\begin{aligned} 1 + \sum_{j=0}^m X_j \prod_{k=0}^{j-1} (1 + X_k) &= 1 + \sum_{j=0}^{m-1} X_j \prod_{k=0}^{j-1} (1 + X_k) + X_m \prod_{k=0}^{m-1} (1 + X_k) \\ &= \prod_{k=0}^{m-1} (1 + X_k) + X_m \prod_{k=0}^{m-1} (1 + X_k) \\ &= \prod_{k=0}^m (1 + X_k). \quad \blacksquare \end{aligned}$$

We now show the convergence.

Theorem 3 Let the following conditions for the split-explicit peer method (8) be satisfied.

- The method has order of consistency p .
- The 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 method has order of convergence p .

Proof: For simplicity of notation we perform the proof for scalar equations. Furthermore we assume that L is regular, i.e. not 0, and $\alpha_i \neq 0$. These restrictions are not necessary but they allow us to write down the solutions of (6) and (11) with the exponential function instead of its Taylor series: The solution of $\dot{y} = C + Ly$ with $y(0) = y_0$ is

$$y(t) = y_0 + (e^{tL} - 1)(y_0 + L^{-1}C).$$

As before we denote Z_i the solution of (6) and \tilde{Z}_i the solution of (11), i.e. it holds

$$\begin{aligned} Z_i(\alpha_i \Delta t) &= \sum_{j=1}^s b_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} s_{ij} Y_{mj} + (e^{\alpha_i \Delta t L} - 1) \left(\sum_{j=1}^s b_{ij} Y_{m-1,j} + \sum_{j=1}^{i-1} s_{ij} Y_{mj} \right. \\ &\quad \left. + (\alpha_i L)^{-1} \left(\sum_{j=1}^s a_{ij} N(Y_{m-1,j}) + \sum_{j=1}^{i-1} r_{ij} N(Y_{mj}) \right) \right), \\ \tilde{Z}_i(\alpha_i \Delta t) &= \sum_{j=1}^s b_{ij} y(t_{m-1,j}) + \sum_{j=1}^{i-1} s_{ij} y(t_{mj}) + (e^{\alpha_i \Delta t L} - 1) \left(\sum_{j=1}^s b_{ij} y(t_{m-1,j}) + \sum_{j=1}^{i-1} s_{ij} y(t_{mj}) \right. \\ &\quad \left. + (\alpha_i L)^{-1} \left(\sum_{j=1}^s a_{ij} N(y(t_{m-1,j})) + \sum_{j=1}^{i-1} r_{ij} N(y(t_{mj})) \right) \right). \end{aligned}$$

In compact notation with $E_m := \text{diag}(e^{\alpha_i \Delta t L} - 1)$ and $\tilde{E}_m := \text{diag}((e^{\alpha_i \Delta t L} - 1)(\alpha_i L)^{-1})$ they are

$$\begin{aligned} Z &= BY_{m-1} + SY_m + E_m BY_{m-1} + E_m SY_m + \tilde{E}_m AN(Y_{m-1}) + \tilde{E}_m RN(Y_m), \\ \tilde{Z} &= BY(t_{m-1}) + SY(t_m) + E_m BY(t_{m-1}) + E_m SY(t_m) + \tilde{E}_m AN(Y(t_{m-1})) + \tilde{E}_m RN(Y(t_m)). \end{aligned}$$

It holds

$$\begin{aligned} \varepsilon_m &= Y(t_m) - Y_m = Y(t_m) - Z = Y(t_m) - \tilde{Z} + \tilde{Z} - Z \\ &= \Delta_m + B(Y(t_{m-1}) - Y_{m-1}) + S(Y(t_m) - Y_m) + E_m B(Y(t_{m-1}) - Y_{m-1}) \\ &\quad + E_m S(Y(t_m) - Y_m) + \tilde{E}_m A(N(Y(t_{m-1})) - N(Y_{m-1})) + \tilde{E}_m R(N(Y(t_m)) - N(Y_m)) \\ &= \Delta_m + B\varepsilon_{m-1} + S\varepsilon_m + E_m B\varepsilon_{m-1} \\ &\quad + E_m S\varepsilon_m + \tilde{E}_m A(N(Y(t_{m-1})) - N(Y_{m-1})) + \tilde{E}_m R(N(Y(t_m)) - N(Y_m)). \end{aligned}$$

The differences of the function values can be eliminated with the mean value theorem

$$N(y(t_{mi})) - N(Y_{mi}) = N_{mi} \varepsilon_{mi}$$

where

$$N_{mi} := \int_0^1 N' \left(y(t_{mi}) + \theta(Y_{mi} - y(t_{mi})) \right) d\theta.$$

With $N_m := \text{diag}(N_{mi})$ we obtain the equation

$$\varepsilon_m = \Delta_m + B\varepsilon_{m-1} + S\varepsilon_m + E_m B\varepsilon_{m-1} + E_m S\varepsilon_m + \tilde{E}_m AN_{m-1}\varepsilon_{m-1} + \tilde{E}_m RN_m \varepsilon_m$$

which is equivalent to

$$\begin{aligned}\varepsilon_m = & (I - S)^{-1}\Delta_m + \tilde{B}\varepsilon_{m-1} + (I - S)^{-1}E_m B\varepsilon_{m-1} + (I - S)^{-1}E_m S\varepsilon_m \\ & + (I - S)^{-1}\tilde{E}_m AN_{m-1}\varepsilon_{m-1} + (I - S)^{-1}\tilde{E}_m RN_m\varepsilon_m.\end{aligned}$$

By recursively substituting the global error in the second term at \tilde{B} with errors from previous time steps one obtains

$$\begin{aligned}\varepsilon_m = & \sum_{j=0}^{m-1} \tilde{B}^j (I - S)^{-1} \Delta_{m-j} + \tilde{B}^m \varepsilon_0 + \sum_{j=0}^{m-1} \tilde{B}^j (I - S)^{-1} E_{m-j} B \varepsilon_{m-1-j} \\ & + \sum_{j=0}^{m-1} \tilde{B}^j (I - S)^{-1} E_{m-j} S \varepsilon_{m-j} + \sum_{j=0}^{m-1} \tilde{B}^j (I - S)^{-1} \tilde{E}_{m-j} AN_{m-1-j} \varepsilon_{m-1-j} + \sum_{j=0}^{m-1} \tilde{B}^j (I - S)^{-1} \tilde{E}_{m-j} RN_{m-j} \varepsilon_{m-j}.\end{aligned}$$

Because the method is zero stable, i.e. \tilde{B} is power-bounded, there exists a constant \tilde{C} so that

$$\begin{aligned}\|\tilde{B}^j (I - S)^{-1} \Delta_{m-j}\| & \leq \tilde{C} \|\Delta_{m-j}\|, \\ \|\tilde{B}^m \varepsilon_0\| & \leq \tilde{C} \|\varepsilon_0\|, \\ \|\tilde{B}^j (I - S)^{-1} E_{m-j} B \varepsilon_{m-1-j}\| & \leq \tilde{C} \Delta t \|\varepsilon_{m-1-j}\|, \\ \|\tilde{B}^j (I - S)^{-1} E_{m-j} S \varepsilon_{m-j}\| & \leq \tilde{C} \Delta t \|\varepsilon_{m-j}\|, \\ \|\tilde{B}^j (I - S)^{-1} \tilde{E}_{m-j} AN_{m-1-j} \varepsilon_{m-1-j}\| & \leq \tilde{C} \Delta t \|\varepsilon_{m-1-j}\|, \\ \|\tilde{B}^j (I - S)^{-1} \tilde{E}_{m-j} RN_{m-j} \varepsilon_{m-j}\| & \leq \tilde{C} \Delta t \|\varepsilon_{m-j}\|\end{aligned}$$

for sufficiently small time step sizes. The Δt in the last four inequalities originates from the first-order terms of the expansions of E_{m-j} and \tilde{E}_{m-j} . The method has order of consistency p , i.e. $\|\Delta_j\| \leq \bar{C}(\Delta t)^{p+1}$. Because of

$$\sum_{j=0}^{m-1} \tilde{C} \|\Delta_{m-j}\| \leq \sum_{j=0}^{m-1} \tilde{C} \bar{C} (\Delta t)^{p+1} \leq (t_e - t_0) \tilde{C} \bar{C} (\Delta t)^p$$

and because the initial values have order p we obtain the inequality for the global error

$$\|\varepsilon_m\| \leq \sum_{j=0}^m \check{C} \Delta t \|\varepsilon_j\| + \check{C} (\Delta t)^p$$

where \check{C} is the maximum of the above constants, i.e. the maximum of the constant from the initial values, $(t_e - t_0) \tilde{C} \bar{C}$ and $4\tilde{C}$. Subtracting $\check{C} \Delta t \|\varepsilon_m\|$ and dividing by $1 - \check{C} \Delta t$ results in

$$\|\varepsilon_m\| \leq \sum_{j=0}^{m-1} \hat{C} \Delta t \|\varepsilon_j\| + \hat{C} (\Delta t)^p \quad (16)$$

where $\hat{C} := \frac{\check{C}}{1 - \check{C} \Delta t}$. With (16) and equation (15) from Theorem 2 with $X_j = \hat{C} \Delta t$ we prove the inequality

$$\|\varepsilon_m\| \leq \left(\prod_{j=0}^{m-1} (1 + \hat{C} \Delta t) \right) \hat{C} h^p$$

by induction: For $m = 0$ the inequality is correct because the initial values have order p . Let the inequality holds for all j with $0 \leq j \leq m - 1$. For m we have

$$\begin{aligned} \|\varepsilon_m\| &\leq \sum_{j=0}^{m-1} \widehat{C}\Delta t \|\varepsilon_j\| + \widehat{C}(\Delta t)^p \\ &\leq \left(1 + \sum_{j=0}^{m-1} \widehat{C}\Delta t \prod_{k=0}^{j-1} (1 + \widehat{C}\Delta t)\right) \widehat{C}(\Delta t)^p \\ &= \left(\prod_{j=0}^{m-1} (1 + \widehat{C}\Delta t)\right) \widehat{C}(\Delta t)^p. \end{aligned}$$

Finally we can show convergence of order p :

$$\begin{aligned} \|\varepsilon_m\| &\leq \left(\prod_{j=0}^{m-1} (1 + \widehat{C}\Delta t)\right) \widehat{C}(\Delta t)^p \\ &\leq e^{\widehat{C}\Delta tm} \widehat{C}(\Delta t)^p \\ &\leq e^{\widehat{C}(t_e - t_0)} \widehat{C}(\Delta t)^p, \end{aligned}$$

i.e. order of convergence p with constant $C := e^{\widehat{C}(t_e - t_0)} \widehat{C}$. ■

Remark 6 While we used the analytical solution of (6) in order to prove convergence, in practical applications numerical schemes are used for the integration of (6). Numerical tests show that the split-explicit methods are second-order if a first-order method with sufficiently small time step sizes or a second-order method is applied to (6).

3.3 Order of consistency for linearly implicit peer methods

We now present order conditions for linearly implicit peer methods. Thereby we make no assumptions for J . Consistency of the linearly implicit peer methods is discussed by considering the local residuals Δ_{mi} obtained by substituting the exact solution y of (9) into the method.

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

Definition 4 A linearly implicit peer method (10) has order of consistency p , if

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

Taylor expansions at t_m lead to the following theorem.

Theorem 4 A linearly implicit peer method (10) has order of consistency p , if

$$\begin{aligned} AB(k) &= 0, & k &= 0, 1, \dots, p, \\ \widehat{AB}(k) &= 0, & k &= 1, \dots, p \end{aligned}$$

where

$$\begin{aligned} AB_i(k) &:= c_i^k - \sum_{j=1}^s b_{ij}(c_j - 1)^k - \sum_{j=1}^{i-1} s_{ij}c_j^k - k \sum_{j=1}^s a_{ij}(c_j - 1)^{k-1} - k \sum_{j=1}^{i-1} r_{ij}c_j^{k-1}, \\ \widehat{AB}_i(k+1) &:= \gamma c_i^k + \sum_{j=1}^s g_{ij}(c_j - 1)^k + \sum_{j=1}^{i-1} h_{ij}c_j^k \quad \text{for } i = 1, \dots, s. \end{aligned}$$

Proof: Using Taylor expansions for $y(t_{mi})$, $y(t_{m-1,j})$, $y(t_{mj})$, $\dot{y}(t_{m-1,j})$ and $\dot{y}(t_{mj})$ at t_m results in

$$\begin{aligned} \Delta_{mi} &= (I - \Delta t \gamma J)y(t_{mi}) - \sum_{j=1}^s b_{ij}y(t_{m-1,j}) - \sum_{j=1}^{i-1} s_{ij}y(t_{mj}) - \Delta t \sum_{j=1}^s a_{ij}\dot{y}(t_{m-1,j}) \\ &\quad - \Delta t \sum_{j=1}^{i-1} r_{ij}\dot{y}(t_{mj}) - \Delta t \sum_{j=1}^s g_{ij}Jy(t_{m-1,j}) - \Delta t \sum_{j=1}^{i-1} h_{ij}Jy(t_{mj}) \\ &= (I - \Delta t \gamma J) \sum_{k=0}^{\infty} \frac{(c_i \Delta t)^k}{k!} y^{(k)}(t_m) \\ &\quad - \sum_{j=1}^s b_{ij} \sum_{k=0}^{\infty} \frac{((c_j - 1) \Delta t)^k}{k!} y^{(k)}(t_m) - \sum_{j=1}^{i-1} s_{ij} \sum_{k=0}^{\infty} \frac{(c_j \Delta t)^k}{k!} y^{(k)}(t_m) \\ &\quad - \Delta t \sum_{j=1}^s a_{ij} \sum_{k=0}^{\infty} \frac{((c_j - 1) \Delta t)^k}{k!} y^{(k+1)}(t_m) - \Delta t \sum_{j=1}^{i-1} r_{ij} \sum_{k=0}^{\infty} \frac{(c_j \Delta t)^k}{k!} y^{(k+1)}(t_m) \\ &\quad - \Delta t \sum_{j=1}^s g_{ij}J \sum_{k=0}^{\infty} \frac{((c_j - 1) \Delta t)^k}{k!} y^{(k)}(t_m) - \Delta t \sum_{j=1}^{i-1} h_{ij}J \sum_{k=0}^{\infty} \frac{(c_j \Delta t)^k}{k!} y^{(k)}(t_m) \\ &= (I - \Delta t \gamma J) \sum_{k=0}^{\infty} \frac{(c_i \Delta t)^k}{k!} y^{(k)}(t_m) \\ &\quad - \sum_{j=1}^s b_{ij} \sum_{k=0}^{\infty} \frac{((c_j - 1) \Delta t)^k}{k!} y^{(k)}(t_m) - \sum_{j=1}^{i-1} s_{ij} \sum_{k=0}^{\infty} \frac{(c_j \Delta t)^k}{k!} y^{(k)}(t_m) \\ &\quad - \sum_{j=1}^s a_{ij} \sum_{k=0}^{\infty} \frac{k(c_j - 1)^{k-1} (\Delta t)^k}{k!} y^{(k)}(t_m) - \sum_{j=1}^{i-1} r_{ij} \sum_{k=0}^{\infty} \frac{k c_j^{k-1} (\Delta t)^k}{k!} y^{(k)}(t_m) \\ &\quad - \Delta t \sum_{j=1}^s g_{ij}J \sum_{k=0}^{\infty} \frac{((c_j - 1) \Delta t)^k}{k!} y^{(k)}(t_m) - \Delta t \sum_{j=1}^{i-1} h_{ij}J \sum_{k=0}^{\infty} \frac{(c_j \Delta t)^k}{k!} y^{(k)}(t_m). \end{aligned}$$

We now separate the parts which incorporate J , i.e. it holds

$$\begin{aligned}
\Delta_{mi} &= \sum_{k=0}^{\infty} \left(c_i^k - \sum_{j=1}^s b_{ij} (c_j - 1)^k - \sum_{j=1}^{i-1} s_{ij} c_j^k \right. \\
&\quad \left. - k \sum_{j=1}^s a_{ij} (c_j - 1)^{k-1} - k \sum_{j=1}^{i-1} r_{ij} c_j^{k-1} \right) \frac{(\Delta t)^k}{k!} y^{(k)}(t_m) \\
&\quad - J \sum_{k=0}^{\infty} \left(\gamma c_i^k + \sum_{j=1}^s g_{ij} (c_j - 1)^k + \sum_{j=1}^{i-1} h_{ij} c_j^k \right) \frac{(\Delta t)^{k+1}}{k!} y^{(k)}(t_m) \\
&= \sum_{k=0}^{\infty} \left(AB_i(k) \right) \frac{(\Delta t)^k}{k!} y^{(k)}(t_m) - J \sum_{k=0}^{\infty} \left(\widehat{AB}_i(k+1) \right) \frac{(\Delta t)^{k+1}}{k!} y^{(k)}(t_m). \quad \blacksquare
\end{aligned}$$

Remark 7 *The last equation of the previous proof demonstrates that the order conditions $AB(k) = 0$ are the conditions of the explicit part of the linearly implicit 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 linearly implicit peer methods have the order independently of what is used as J .*

The collected order conditions $AB(k) = 0$ and $\widehat{AB}(k) = 0$ for order $p = s - 1$ can be written in compact matrix form analogously to the ansatz in Section 3.1. For this purpose we use the matrix $\Gamma = \gamma I$ and the matrices C, D, V_0 and V_1 defined in Section 3.1. The first condition $AB(0) = 0$ simply is

$$(B + S)\mathbf{1} = \mathbf{1}.$$

The other conditions, i.e. $1 \leq k \leq s - 1$, lead to the matrix equations

$$\begin{aligned}
CV_0 - B(C - I)V_1 - SCV_0 - AV_1D - RV_0D - \beta e_s^T D &= 0, \\
\Gamma V_0 + GV_1 + HV_0 + \tilde{\beta} e_s^T &= 0
\end{aligned}$$

where β and $\tilde{\beta}$ represent the remaining degrees of freedom. Because V_1 is a Vandermonde matrix it is regular if the nodes c_i are different from each other. Therefore a linearly implicit peer method with

$$\begin{aligned}
(B + S)\mathbf{1} &= \mathbf{1}, \\
A &= \left(CV_0 - B(C - I)V_1 - SCV_0 - RV_0D - \beta e_s^T D \right) D^{-1} V_1^{-1}, \\
G &= - \left(\Gamma V_0 + HV_0 + \tilde{\beta} e_s^T \right) V_1^{-1}
\end{aligned}$$

has order of consistency $p = s - 1$ independently of J . In the remainder we will concentrate on three-stage methods with order of consistency 2.

In [13] a second-order three-stage split-explicit peer method was used for the integration of the compressible Euler equations. That method is presented in this thesis,

too. It is quite accurate and has good stability properties. Therefore we use it for the explicit part of a linearly implicit peer method. Applying that split-explicit peer method to the split-differential equation (1) and using the trapezoidal rule for the integration of the fast part results in 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 explained in Remark 4. We keep the coefficients which belong to the explicit part, i.e. c_i , b_{ij} , s_{ij} , a_{ij} and r_{ij} , and use new implicit coefficients γ_i , g_{ij} and h_{ij} in order to find a method with better stability properties. Because the parameters g_{ij} are used to fulfil the order conditions for order 2 the remaining degrees of freedom for the stability optimization are γ , $\tilde{\beta}_1$, $\tilde{\beta}_2$, $\tilde{\beta}_3$, h_{21} , h_{31} and h_{32} .

Another linearly implicit peer method presented in this thesis is generated from a completely new set of parameters. It also has $s = 3$ stages and order of consistency 2 but it additionally fulfils conditions which guarantee superconvergence, i.e. it has order of convergence 3.

3.4 Order of convergence for linearly implicit peer methods

We now show convergence for the class of linearly implicit peer methods. We also give a condition which guarantees superconvergence. The definitions of zero stability and order of convergence for linearly implicit peer methods are analogous to the definitions for split-explicit peer methods. Furthermore we define strong zero stability.

Definition 5 *A linearly implicit peer method (10) is called zero stable if the eigenvalues λ of $\tilde{B} := (I - S)^{-1}B$ satisfy the conditions*

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

If one eigenvalue is 1 and the moduli of the other eigenvalues are strictly smaller than 1 the method is called strongly zero stable.

Definition 6 *A linearly implicit peer method (10) 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*

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

Because we consider methods with constant time step sizes, convergence follows straightforwardly from consistency and zero stability.

Theorem 5 *Let the following conditions for the linearly implicit peer method (10) be satisfied.*

- *The method has order of consistency p .*
- *The 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 method has order of convergence p .

Proof: The proof is similar to the proof of convergence for split-explicit peer methods. ■

Under rather general additional assumptions order of convergence p follows from order of consistency p . For the case of constant step sizes the concept of quasi-consistency was introduced in [32]. It was shown that a quasi-consistent method of order $p + 1$ is also convergent of order $p + 1$. The assumption of quasi-consistency of order $p + 1$ is less stringent than consistency of the same order. It requires consistency of order p and an additional condition. Formulated for linearly implicit peer methods this additional condition reads

$$\tilde{B}^\infty \Delta_m = \mathcal{O}((\Delta t)^{p+1})$$

where the matrix

$$\tilde{B}^\infty = \mathbb{1}\tilde{v}^T$$

is defined in Theorem 6. We will show that this condition together with order of consistency p implies convergence of order $p + 1$ which therefore is called superconvergence. This idea was applied to peer methods in [42] with the difference that the methods in [42] are optimally zero stable, i.e. B has one eigenvalue 1 and the other eigenvalues are 0. Our ansatz is a generalization because we only demand strong zero stability, i.e. the moduli of all but one eigenvalue have to be strictly smaller than 1. As preliminary step to the proof of superconvergence we need the following theorem.

Theorem 6 *If a linearly implicit peer method (10) is strongly zero stable and fulfils the preconsistency condition $(B + S)\mathbb{1} = \mathbb{1}$ the limit*

$$\tilde{B}^\infty := \lim_{j \rightarrow \infty} \tilde{B}^j = \lim_{j \rightarrow \infty} ((I - S)^{-1}B)^j$$

exists and $\tilde{B}^\infty = \mathbb{1}\tilde{v}^T$ where \tilde{v} is a left eigenvector of \tilde{B}^∞ to the eigenvalue 1 with $\tilde{v}\mathbb{1} = 1$. Furthermore the series

$$\sum_{j=0}^{\infty} (\tilde{B}^j - \tilde{B}^\infty)$$

converges.

Proof: Because S is a strictly lower triangular matrix \tilde{B} exists and the order condition $(B + S)\mathbb{1} = \mathbb{1}$ is equivalent to $\tilde{B}\mathbb{1} = \mathbb{1}$, i.e. \tilde{B} has one eigenvalue 1. The other eigenvalues have moduli strictly smaller than 1 because the method is strongly zero stable. With the Jordan normal form $W = Q^{-1}\tilde{B}Q$ of \tilde{B} the matrix powers can be written in the form

$$\tilde{B}^j = (QWQ^{-1}) \cdots (QWQ^{-1}) = QW^jQ^{-1}.$$

Because of the block structure of W every block

$$W_k = \lambda_k I_m + D_m,$$

where λ_k is the eigenvalue, I_m the identity matrix, D_m the secondary diagonal with ones and m the dimension of the block, can be considered separately. It holds

$$W_k^j = (\lambda_k I_m + D_m)^j = \lambda_k^j I_m + \sum_{l=1}^{m-1} \binom{j}{l} \lambda_k^{j-l} D_m^l.$$

For $\lambda_k = 1$, which originates from the consistency condition, $W_k^j = 1$. Because the other eigenvalues have moduli strictly smaller than 1 it holds

$$\lim_{j \rightarrow \infty} W_k^j = \lim_{j \rightarrow \infty} \left(\lambda_k^j I_m + \sum_{l=1}^{m-1} \binom{j}{l} \lambda_k^{j-l} D_m^l \right) = 0.$$

So W^∞ exists and only has one 1 on the main diagonal, i.e. it has rank 1. Therefore \tilde{B}^∞ has rank 1 at most, each of the rows of \tilde{B}^∞ is a multiple of the other rows. From $\tilde{B} \mathbf{1} = \mathbf{1}$ follows $\tilde{B}^\infty \mathbf{1} = \mathbf{1}$, so the sum of each row of \tilde{B}^∞ is 1 and all rows must be equal. They are denoted by the vector \tilde{v}^T , i.e. $\tilde{B}^\infty = \mathbf{1} \tilde{v}^T$ and $\tilde{v}^T \mathbf{1} = 1$. Furthermore it holds

$$\tilde{v}^T \tilde{B}^\infty = \tilde{v}^T \mathbf{1} \tilde{v}^T = \tilde{v}^T,$$

i.e. \tilde{v}^T is a left eigenvector of \tilde{B}^∞ to the eigenvalue 1.

Because of

$$\sum_{j=0}^{\infty} (\tilde{B}^j - \tilde{B}^\infty) = \sum_{j=0}^{\infty} Q(W^j - W^\infty)Q^{-1}$$

we consider each of the blocks separately. The block which corresponds to the eigenvalue 1 vanishes. For the other blocks there exist constants C_k so that

$$\|W_k^j - W_k^\infty\| = \|(\lambda_k I_m + D_m)^j - 0\| = \|\lambda_k^j I_m + \sum_{l=1}^{m-1} \binom{j}{l} \lambda_k^{j-l} D_m^l\| \leq C_k j^{m-1} |\lambda_k|^{j-m+1}$$

holds for sufficiently large j . The series

$$\sum_{j=0}^{\infty} j^{m-1} |\lambda_k|^{j-m+1}$$

converges because $|\lambda_k|$ is strictly smaller than 1, i.e. there exists a positive constant which is strictly smaller than 1 and an upper bound for

$$\frac{(j+1)^{m-1} |\lambda_k|^{j+1-m+1}}{j^{m-1} |\lambda_k|^{j-m+1}} = |\lambda_k| \left(\frac{j+1}{j} \right)^{m-1}$$

if j is sufficiently large. Therefore the ratio criterion is satisfied, which is sufficient for the convergence of the series. Because these arguments are valid for all blocks the convergence of

$$\sum_{j=0}^{\infty} (\tilde{B}^j - \tilde{B}^\infty)$$

is shown. ■

With the previous theorem we now can show that two additional conditions guarantee superconvergence.

Theorem 7 *Let the following conditions for the linearly implicit peer method (10) be satisfied.*

- The method has order of consistency p .
- The method is strongly zero stable.
- The starting values have order $p+1$, i.e. $y(t_{0i}) - Y_{0i} = \mathcal{O}((\Delta t)^{p+1})$ for $i = 1, \dots, s$.
- \tilde{v} from Theorem 6 satisfies $\tilde{v}^T(I - S)^{-1}AB(p+1) = \tilde{v}^T(I - S)^{-1}\widehat{AB}(p+1) = 0$.

Then this method has order of convergence $p+1$.

Proof: For simplicity of notation we again perform the proof for scalar equations. It holds

$$\begin{aligned}
(I - \Delta t \gamma J) \varepsilon_m &= (I - \Delta t \gamma J)(Y(t_m) - Y_m) = (I - \Delta t \gamma J)Y(t_m) \\
&\quad - BY(t_{m-1}) - SY(t_m) - \Delta t Af(Y(t_{m-1})) - \Delta t Rf(Y(t_m)) - \Delta t GJY(t_{m-1}) - \Delta t HJY(t_m) \\
&\quad + BY(t_{m-1}) + SY(t_m) + \Delta t Af(Y(t_{m-1})) + \Delta t Rf(Y(t_m)) + \Delta t GJY(t_{m-1}) + \Delta t HJY(t_m) \\
&\quad - BY_{m-1} - SY_m - \Delta t AF_{m-1} - \Delta t RF_m - \Delta t GJY_{m-1} - \Delta t HJY_m \\
&= \Delta_m + B(Y(t_{m-1}) - Y_{m-1}) + S(Y(t_m) - Y_m) + \Delta t A(f(Y(t_{m-1})) - F_{m-1}) \\
&\quad + \Delta t R(f(Y(t_m)) - F_m) + \Delta t GJ(Y(t_{m-1}) - Y_{m-1}) + \Delta t HJ(Y(t_m) - Y_m).
\end{aligned}$$

The differences of the function values can be eliminated with the mean value theorem

$$f(y(t_{mi})) - F_{mi} = G_{mi} \varepsilon_{mi}$$

where

$$G_{mi} := \int_0^1 f'(y(t_{mi}) + \theta(Y_{mi} - y(t_{mi}))) d\theta.$$

With $G_m := \text{diag}(G_{mi})$ we obtain the equation

$$(I - \Delta t \gamma J) \varepsilon_m = \Delta_m + B \varepsilon_{m-1} + S \varepsilon_m + \Delta t A G_{m-1} \varepsilon_{m-1} + \Delta t R G_m \varepsilon_m + \Delta t G J \varepsilon_{m-1} + \Delta t H J \varepsilon_m$$

which is equivalent to

$$\varepsilon_m = (I - S)^{-1} \Delta_m + \tilde{B} \varepsilon_{m-1} + \Delta t \tilde{A} G_{m-1} \varepsilon_{m-1} + \Delta t \tilde{R} G_m \varepsilon_m + \Delta t \tilde{G} J \varepsilon_{m-1} + \Delta t \tilde{H} J \varepsilon_m$$

where $\tilde{B} = (I - S)^{-1}B$, $\tilde{A} = (I - S)^{-1}A$, $\tilde{R} = (I - S)^{-1}R$, $\tilde{G} = (I - S)^{-1}G$ and $\tilde{H} = (I - S)^{-1}(H + \gamma I)$. By recursively substituting the global error in the second term of the above equation with errors from previous time steps one obtains

$$\begin{aligned}
\varepsilon_m &= \sum_{j=0}^{m-1} \tilde{B}^j (I - S)^{-1} \Delta_{m-j} + \tilde{B}^m \varepsilon_0 + \Delta t \sum_{j=0}^{m-1} \tilde{B}^j \tilde{A} G_{m-1-j} \varepsilon_{m-1-j} \\
&\quad + \Delta t \sum_{j=0}^{m-1} \tilde{B}^j \tilde{R} G_{m-j} \varepsilon_{m-j} + \Delta t \sum_{j=0}^{m-1} \tilde{B}^j \tilde{G} J \varepsilon_{m-1-j} + \Delta t \sum_{j=0}^{m-1} \tilde{B}^j \tilde{H} J \varepsilon_{m-j}.
\end{aligned}$$

Because the method is zero stable, i.e. \tilde{B} is power-bounded, there exists a constant \tilde{C} so that

$$\begin{aligned}\|\tilde{B}^m \varepsilon_0\| &\leq \tilde{C} \|\varepsilon_0\|, \\ \|\tilde{B}^j \tilde{A} G_{m-1-j} \varepsilon_{m-1-j}\| &\leq \tilde{C} \|\varepsilon_{m-1-j}\|, \\ \|\tilde{B}^j \tilde{R} G_{m-j} \varepsilon_{m-j}\| &\leq \tilde{C} \|\varepsilon_{m-j}\|, \\ \|\tilde{B}^j \tilde{G} J \varepsilon_{m-1-j}\| &\leq \tilde{C} \|\varepsilon_{m-1-j}\|, \\ \|\tilde{B}^j \tilde{H} J \varepsilon_{m-j}\| &\leq \tilde{C} \|\varepsilon_{m-j}\|.\end{aligned}$$

This results in

$$\begin{aligned}\|\varepsilon_m\| &\leq \left\| \sum_{j=0}^{m-1} \tilde{B}^j (I - S)^{-1} \Delta_{m-j} \right\| + \tilde{C} \|\varepsilon_0\| + 4\tilde{C} \Delta t \sum_{j=0}^m \|\varepsilon_{m-j}\| \\ &\leq \left\| \sum_{j=0}^{m-1} (\tilde{B}^j - \tilde{B}^\infty) (I - S)^{-1} \Delta_{m-j} \right\| + \left\| \sum_{j=0}^{m-1} \tilde{B}^\infty (I - S)^{-1} \Delta_{m-j} \right\| \\ &\quad + \tilde{C} \|\varepsilon_0\| + 4\tilde{C} \Delta t \sum_{j=0}^m \|\varepsilon_{m-j}\|.\end{aligned}$$

Because of Theorem 6 and the order of consistency p of the method the inequality

$$\left\| \sum_{j=0}^{m-1} (\tilde{B}^j - \tilde{B}^\infty) (I - S)^{-1} \Delta_{m-j} \right\| \leq \left\| \sum_{j=0}^{m-1} (\tilde{B}^j - \tilde{B}^\infty) \right\| \|(I - S)^{-1}\| C_1 (\Delta t)^{p+1} \leq C_2 (\Delta t)^{p+1}$$

holds with some constants C_1 (resulting from the consistency) and C_2 (resulting from the convergence of the series, the norm of $(I - S)^{-1}$ and C_1). The conditions

$$\tilde{v}^T (I - S)^{-1} A B(p+1) = \tilde{v}^T (I - S)^{-1} \widehat{A} B(p+1) = 0$$

result in

$$\tilde{v}^T (I - S)^{-1} \Delta_{m-j} = \mathcal{O}((\Delta t)^{p+2})$$

and together with $\tilde{B}^\infty = \mathbb{1} \tilde{v}^T$ this leads to

$$\left\| \sum_{j=0}^{m-1} \tilde{B}^\infty (I - S)^{-1} \Delta_{m-j} \right\| = \left\| \sum_{j=0}^{m-1} \mathbb{1} \tilde{v}^T (I - S)^{-1} \Delta_{m-j} \right\| \leq \sum_{j=0}^{m-1} C_3 (\Delta t)^{p+2} \leq C_4 (\Delta t)^{p+1}.$$

Because the initial values have order $p+1$ it holds $\tilde{C} \|\varepsilon_0\| \leq C_5 (\Delta t)^{p+1}$. Denoting \check{C} the maximum of the constants $3C_2$, $3C_4$, $3C_5$ and $4\tilde{C}$ and using the previous inequalities results in the inequality for the global error

$$\|\varepsilon_m\| \leq \sum_{j=0}^m \check{C} \Delta t \|\varepsilon_{m-j}\| + \check{C} (\Delta t)^{p+1}.$$

The remainder of the proof is analogous to the proof of convergence for split-explicit peer methods with the exception that the order is $p+1$ instead of p . \blacksquare

Remark 8 *In the remainder of this thesis order means order of convergence.*

4 Stability investigations

4.1 Linearization of the Euler equations

We now discuss the stability properties of the considered methods. In [44], [45] and [13] amongst others the authors used a 2×2 -system incorporating wind and pressure variables in order to consider the effects of the different physical processes, wind and sound waves, on the stability of the investigated methods. In contrast to that ansatz we consider a 3×3 -system which also incorporates the density. This approach is closer to the compressible Euler equations and better takes the effects of a simplified Jacobian into account. We start with the one-dimensional compressible Euler equations in conservative form. In some of the numerical tests diffusion is incorporated and the split-explicit Runge-Kutta method which we consider for comparison with the peer methods needs divergence damping, in one dimension this is the same as diffusion, in order to be stable. Therefore we also include a diffusion term with a constant diffusion coefficient ν .

$$\begin{aligned}\dot{\rho} &= -\frac{\partial \rho u}{\partial x}, \\ (\dot{\rho}u) &= -\frac{\partial \rho u u}{\partial x} - \frac{\partial p}{\partial x} + \nu \rho \frac{\partial^2 u}{\partial x^2}, \\ (\dot{\rho}\theta) &= -\frac{\partial \rho u \theta}{\partial x}.\end{aligned}$$

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

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

where R_d is the gas constant for dry air, $\kappa = R_d/c_p$, c_p the heat capacity of dry air at constant pressure and p_0 is the pressure at ground level. To linearize these nonlinear equations 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

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

where

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

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

$$\begin{aligned}\frac{\partial \rho u u}{\partial x} &= -u^2 \frac{\partial \rho}{\partial x} + 2u \frac{\partial \rho u}{\partial x}, \\ \frac{\partial \rho u \theta}{\partial x} &= -u \theta \frac{\partial \rho}{\partial x} + \theta \frac{\partial \rho u}{\partial x} + u \frac{\partial \rho \theta}{\partial x}, \\ \rho \frac{\partial^2 u}{\partial x^2} &= \frac{\partial^2 \rho u}{\partial x^2} - u \frac{\partial^2 \rho}{\partial x^2} - \frac{2}{\rho} \frac{\partial \rho}{\partial x} \frac{\partial \rho u}{\partial x} + \frac{2u}{\rho} \frac{\partial \rho}{\partial x} \frac{\partial \rho}{\partial x}.\end{aligned}$$

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

$$\begin{pmatrix} \dot{\rho}' \\ (\rho u)' \\ \frac{1}{\theta}(\rho\theta)' \end{pmatrix} = - \begin{pmatrix} 0 & 1 & 0 \\ -\bar{u}^2 & 2\bar{u} & c_s^2 \\ -\bar{u} & 1 & \bar{u} \end{pmatrix} \begin{pmatrix} \rho'_x \\ (\rho u)'_x \\ \frac{1}{\theta}(\rho\theta)'_x \end{pmatrix} + \nu \begin{pmatrix} 0 & 0 & 0 \\ -\bar{u} & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \rho'_{xx} \\ (\rho u)'_{xx} \\ \frac{1}{\theta}(\rho\theta)'_{xx} \end{pmatrix} \quad (17)$$

where we divided the potential temperature by the background potential temperature for simplicity of notation. When we compute with split-explicit methods all terms in the first matrix in (17) with \bar{u} belong to the advection part (red), i.e. after the spatial discretization this is N in (1). The other terms (black) in the first matrix and the diffusion terms (blue) are treated as the fast part, i.e. L . While divergence damping and diffusion are the same in one dimension this is not the case in two or more dimensions. However, divergence damping and diffusion will be treated both as fast parts for split-explicit methods, i.e. in contrast to the advection terms these terms will be advanced with smaller time steps or an implicit method.

When using linearly implicit methods we will not use the spatial discretization of (17) as Jacobian but we will make three simplifications for J in order to save memory and gain computational efficiency:

- We use the conservative form of the Euler equations, i.e. equations with prognostic variables ρu and $\rho\theta$, as the right-hand side because this guarantees conservation properties, namely conservation of mass, momentum and entropy. Contrariwise we use the Jacobian which arises 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. For the Jacobian we use the first-order upwind scheme. Therefore we will call it simplified Jacobian in the remainder of this thesis.
- We will use the 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 thesis we will call it partial Jacobian.

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

$$\begin{pmatrix} \dot{\rho}' \\ (\bar{\rho} u)' \\ \frac{1}{\theta}(\rho\theta)' \end{pmatrix} = - \begin{pmatrix} \bar{u} & 1 & 0 \\ 0 & \bar{u} & c_s^2 \\ 0 & 1 & \bar{u} \end{pmatrix} \begin{pmatrix} \rho'_x \\ (\bar{\rho} u)'_x \\ \frac{1}{\theta}(\rho\theta)'_x \end{pmatrix} + \nu \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \rho'_{xx} \\ (\bar{\rho} u)'_{xx} \\ \frac{1}{\theta}(\rho\theta)'_{xx} \end{pmatrix}. \quad (18)$$

This set of equations (18) is simpler than (17) with respect to two aspects: It has less terms than (17) which results in computational efficiency when solving the resulting

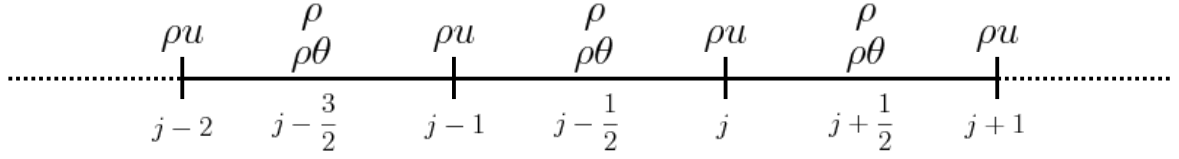


Figure 2: Variables defined on a one-dimensional staggered grid.

linear system of equations. Table 1 in Section 5.2 shows the memory savings as well in 2D as in 3D. Furthermore it is easier to implement because there 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 (18) 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 (17) as the right-hand side but the derivative of (18) as Jacobian despite the fact that it originates from the advection form of the Euler equations. In order to investigate the influence of the second simplification of the Jacobian we perform a von Neumann stability analysis.

4.2 Linear stability analysis

As this is the case for many weather models we use a staggered grid, i.e. the density and the potential temperature are cell-centered while the wind is defined on the faces. This is illustrated by Figure 2. If the grid is periodic we can use the von Neumann stability analysis by assuming a Fourier series ansatz for the variables, i.e.

$$\begin{aligned}\rho'(t, x_{j+1/2}) &= \rho'(t) \sum_k e^{ikx_{j+1/2}}, \\ (\rho\theta)'(t, x_{j+1/2}) &= (\rho\theta)'(t) \sum_k e^{ikx_{j+1/2}}, \\ (\rho u)'(t, x_j) &= (\rho u)'(t) \sum_k e^{ikx_j}\end{aligned}$$

where $k = \frac{2\pi}{L}$ is the wave number, $L = 2\Delta x, 3\Delta x, 4\Delta x, \dots$ the wavelength and Δx the spatial step size. With this approach we investigate the stability for each wave number separately.

Four different schemes are used for the spatial discretizations of (17) and (18). The first-order derivatives without the wind speed \bar{u} involved, i.e. the black terms, belong to the acoustic part of the compressible equations and are discretized with central differences, e.g. the spatial discretization of the differential equation $\dot{\rho}' = -(\rho u)'_x$ reads

$$\dot{\rho}'(t)e^{ikx_{j+1/2}} = -\frac{(\rho u)'(t)e^{ikx_{j+1}} - (\rho u)'(t)e^{ikx_j}}{\Delta x}.$$

Omitting the time variable and the apostrophe for simplicity of notation and dividing by $e^{ikx_{j+1/2}}$ results in

$$\dot{\rho} = -(\rho u) \frac{e^{\frac{ik\Delta x}{2}} - e^{-\frac{ik\Delta x}{2}}}{\Delta x} = -\mathcal{D}_2(\rho u)$$

with

$$\mathcal{D}_2 := \frac{1}{\Delta x} \left(e^{\frac{ik\Delta x}{2}} - e^{-\frac{ik\Delta x}{2}} \right).$$

The red terms in (17) belong to the advection part. As in the weather codes WRF, COSMO and others we use the third-order upwind scheme described in [44] for its spatial discretization, e.g. we need to discretize the derivative $-2\bar{u}(\rho u)_x$ in the momentum equation of (17). For a positive wind \bar{u} the third-order upwind scheme reads

$$(\rho u)_{j+1/2} := \frac{1}{6} \left(2(\rho u)_{j+1} + 5(\rho u)_j - (\rho u)_{j-1} \right). \quad (19)$$

Therefore the discretization of $-2\bar{u}(\rho u)_x$ at j is

$$-2\bar{u} \frac{(\rho u)_{j+1/2} - (\rho u)_{j-1/2}}{\Delta x} = -\frac{2\bar{u}}{6\Delta x} \left(2(\rho u)_{j+1} + 3(\rho u)_j - 6(\rho u)_{j-1} + (\rho u)_{j-2} \right),$$

i.e. the operator is

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

The second simplification for the Jacobian of linearly implicit peer methods is to use the first-order upwind scheme for the advection part, i.e. for the red terms in (18). For $\bar{u} > 0$ the first-order upwind scheme is

$$(\rho u)_{j+1/2} := (\rho u)_j \quad (20)$$

and therefore

$$-\bar{u} \frac{(\rho u)_{j+1/2} - (\rho u)_{j-1/2}}{\Delta x} = -\frac{\bar{u}}{\Delta x} \left((\rho u)_j - (\rho u)_{j-1} \right),$$

the resulting operator is

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

The diffusion terms, i.e. the blue terms in (17) and (18), will be discretized with second-order central differences, i.e. with the operator

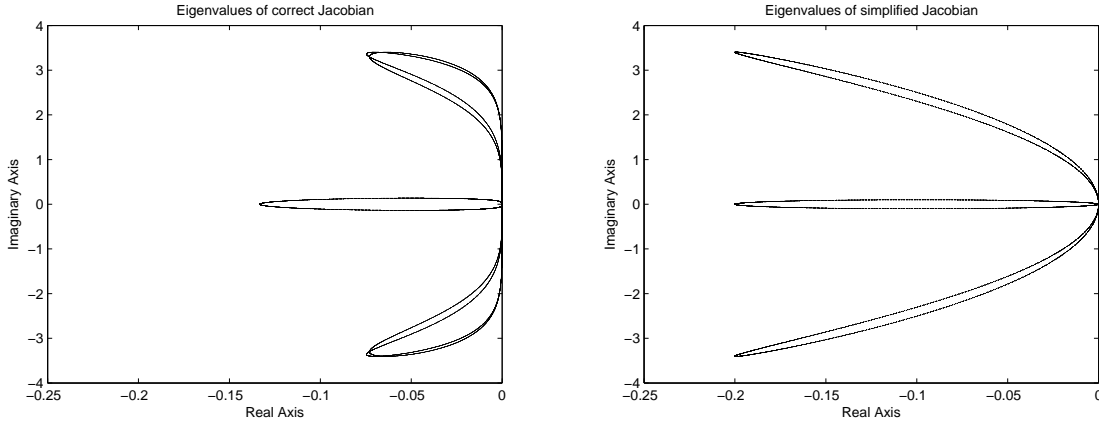
$$\mathcal{D}_\nu := \frac{1}{(\Delta x)^2} \left(e^{ik\Delta x} - 2 + e^{-ik\Delta x} \right).$$

Using these operators as approximations to the spatial derivatives in (17) results in the ordinary differential equation

$$\begin{pmatrix} \dot{\rho} \\ (\dot{\rho}u) \\ \frac{1}{\theta}(\dot{\rho}\theta) \end{pmatrix} = - \begin{pmatrix} 0 & \mathcal{D}_2 & 0 \\ -\bar{u}^2\mathcal{D}_3 + \nu\bar{u}\mathcal{D}_\nu & 2\bar{u}\mathcal{D}_3 - \nu\mathcal{D}_\nu & c_s^2\mathcal{D}_2 \\ -\bar{u}\mathcal{D}_3 & \mathcal{D}_2 & \bar{u}\mathcal{D}_3 \end{pmatrix} \begin{pmatrix} \rho \\ (\rho u) \\ \frac{1}{\theta}(\rho\theta) \end{pmatrix}. \quad (21)$$

For the Jacobian in cut cell regions we instead use the matrix which results from (18) and belongs to the system

$$\begin{pmatrix} \dot{\rho} \\ (\dot{\rho}u) \\ \frac{1}{\theta}(\dot{\rho}\theta) \end{pmatrix} = - \begin{pmatrix} \bar{u}\mathcal{D}_1 & \mathcal{D}_2 & 0 \\ 0 & \bar{u}\mathcal{D}_1 - \nu\mathcal{D}_\nu & c_s^2\mathcal{D}_2 \\ 0 & \mathcal{D}_2 & \bar{u}\mathcal{D}_1 \end{pmatrix} \begin{pmatrix} \rho \\ \bar{\rho}u \\ \frac{1}{\theta}(\rho\theta) \end{pmatrix}. \quad (22)$$

Figure 3: Eigenvalues of M (left) and \widetilde{M} (right).

The third simplification for the Jacobian when using linearly implicit peer methods has an effect in the regions without cut cells. There the Jacobian includes the acoustic part only, i.e. no advection and diffusion. The corresponding system is

$$\begin{pmatrix} \dot{\rho} \\ (\dot{\rho}u) \\ \frac{1}{\theta}(\dot{\rho}\theta) \end{pmatrix} = - \begin{pmatrix} 0 & \mathcal{D}_2 & 0 \\ 0 & 0 & c_s^2 \mathcal{D}_2 \\ 0 & \mathcal{D}_2 & 0 \end{pmatrix} \begin{pmatrix} \rho \\ \rho u \\ \frac{1}{\theta}(\rho\theta) \end{pmatrix}. \quad (23)$$

We now consider the eigenvalues of the above systems. We denote M the matrix of system (21), which is used for the split-explicit methods and as the right-hand side for linearly implicit methods. For linearly implicit methods the matrix \widetilde{M} of the corresponding system (22) is used as Jacobian in cut cells and in the free atmosphere we use the discretization (23) with its matrix named \widehat{M} . While the eigenvalues of the matrices from (17) and (18) are equal if there is no diffusion, namely $-\bar{u}$ and $-\bar{u} \pm c_s$, the eigenvalues of M are

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

whereas the eigenvalues of \widetilde{M} are

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

and \widehat{M} has the eigenvalues

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

if $\nu = 0$.

Figure 3 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 (21), (22) and (23). In contrast the full but 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 into account when searching for stable linearly implicit methods which use the approximate Jacobian.

4.3 Stability properties of split-explicit methods

In order to investigate stability properties of split-explicit methods we apply them to the system (21). The terms with the third-order operators are treated as the slow part, the second-order operators belong to the fast part. So we have to solve (1) with

$$N = - \begin{pmatrix} 0 & 0 & 0 \\ -\bar{u}^2 \mathcal{D}_3 & 2\bar{u} \mathcal{D}_3 & 0 \\ -\bar{u} \mathcal{D}_3 & 0 & \bar{u} \mathcal{D}_3 \end{pmatrix} \quad \text{and} \quad L = - \begin{pmatrix} 0 & \mathcal{D}_2 & 0 \\ \nu \bar{u} \mathcal{D}_\nu & -\nu \mathcal{D}_\nu & c_s^2 \mathcal{D}_2 \\ 0 & \mathcal{D}_2 & 0 \end{pmatrix}.$$

We show the results for four methods which will be described in detail in Section 5.7.

- RK3-FB is a second-order three-stage split-explicit Runge-Kutta method which uses the explicit forward-backward Euler scheme for the integration of the fast part (2) with $n_s = 30$ small time steps $\Delta\tau$ per large time step Δt .
- RK3-TR is the same split-explicit Runge-Kutta method which uses the implicit trapezoidal rule for the integration of the fast part (2) with equal time step sizes for both parts.
- explPeer-FB is a second-order three-stage split-explicit peer method which uses the explicit forward-backward Euler scheme for the integration of the fast part (6) with $n_s = 30$ small time steps $\Delta\tau$ per large time step Δt .
- explPeer-TR is the same split-explicit peer method which uses the implicit trapezoidal rule for the integration of the fast part (2) with equal time step sizes for both parts.

If we apply these methods to the 3×3 -system and perform one integration step with time step size Δt we obtain systems $Y_m = \bar{A} Y_{m-1}$ with \bar{A} called the amplification matrix. It depends on the method, the CFL numbers $C_{adv} := \bar{u} \frac{\Delta t}{\Delta x}$ and $C_{sound} := c_s \frac{\Delta t}{\Delta x}$, the wave number k and the diffusion coefficient ν .

Figure 4 shows the stability diagrams for the considered split-explicit methods. 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 the amplification matrix for wavelengths between $2\Delta x$ and $20\Delta x$, i.e. the method is stable for some coordinates in the C_{adv} - C_{sound} -plane if the maximum is at most 1 at that point. With one exception for all diagrams $\nu = 0$. In each diagram there is a line with a slope of 1/6. We are only interested in the stability regions below of this line because there $\bar{u} < \frac{c_s}{6} \approx 56 \text{ m s}^{-1} \approx 200 \text{ km h}^{-1}$ what is sufficient for practical applications. In [44] the authors argued that even a slope of 1/12 is sufficient.

We can see that the split-explicit peer method is stable for advection CFL numbers smaller than 1.7 without the need for any artificial damping. It is

$$C_{sound} = \frac{c_s \Delta t}{\Delta x} = n_s \frac{c_s \Delta \tau}{\Delta x}$$

and the forward-backward Euler method is stable up to $\frac{c_s \Delta \tau}{\Delta x}$. Because the forward-backward Euler scheme is applied n_s times with small time steps $\Delta\tau$ to the fast part there

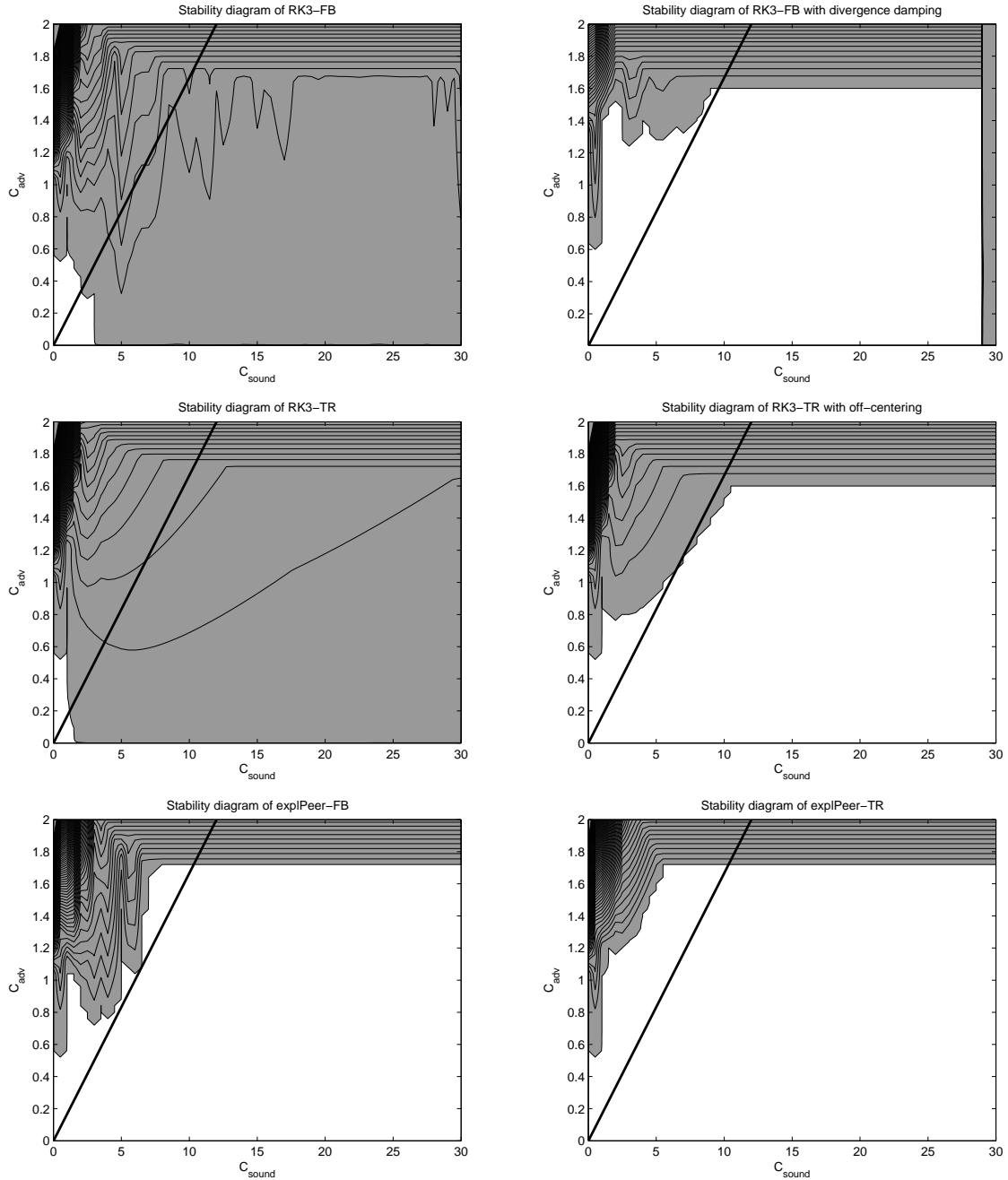


Figure 4: Stability diagrams for RK3-FB without (top left) and with (top right) divergence damping, for RK3-TR without (middle left) and with (middle right) off-centering and for explPeer-FB (bottom left) and explPeer-TR (bottom right). Unstable regions in grey with contour interval 0.1.

is the stability limit $C_{sound} \leq n_s$. The split-explicit peer method in combination with the forward-backward Euler scheme is stable for sound CFL numbers up to this limit. So the combination of peer method and forward-backward Euler scheme harmonizes very well and there is no stability limit for sound CFL numbers if one uses sufficiently small time step sizes for the integration of the fast part. The same holds for the combination of the peer method with the trapezoidal rule, there is no stability restriction from the sound CFL numbers.

As can be seen this is not the case for the split-explicit Runge-Kutta method. Without damping it is not sufficiently stable neither with the forward-backward scheme nor with the trapezoidal rule. It becomes stable if divergence damping is used. We used a fixed value of $0.025 = \nu \frac{\Delta\tau}{\Delta x^2}$, the same as in [44]. This means that the original differential equation is artificially altered by the term νu_{xx} which depends on the spatial and temporal discretization and increases with decreasing time step sizes. Such a method cannot converge in time if the spatial step size is kept constant. Without this damping term the split-explicit Runge-Kutta method is unstable in the whole region where $C_{sound} > \pi$ and therefore the acoustic modes restrict the maximum time step size because $c_s \gg \bar{u}$ in practical applications. In combination with the trapezoidal rule the split-explicit Runge-Kutta method is unstable, too. The trapezoidal rule has no amplitude error which makes it critical for the use with problems with eigenvalues lying on the imaginary axis. It is part of the class of θ -methods where $\theta = 0$ is the explicit Euler scheme, $\theta = 0.5$ is the trapezoidal rule and $\theta = 1$ is the implicit Euler scheme. When using the Runge-Kutta method with some little off-centering, e.g. with $\theta = 0.55$, the resulting split-explicit method is stable. On the other hand the order of this method is reduced to 1 due to the off-centering.

These results for the split-explicit Runge-Kutta method were the main motivation for the construction of a time-splitting method which does not suffer from stability restrictions due to acoustic modes but furthermore does not need any artificial damping term. As shown above these aims are reached with the split-explicit peer method.

4.4 Stability properties of linearly implicit methods

Because the linearly implicit methods should be as efficient as split-explicit methods in regions without cut cells but furthermore should be stable in cut cell regions we present two stability diagrams for each method: One stability diagram shows the results when applying them to (21) but with \widehat{M} originating from (23) as Jacobian for efficiency reasons, i.e.

$$Y_m(I - \Delta t \gamma \widehat{M})^T = BY_{m-1} + SY_m + \Delta t AY_{m-1} M^T + \Delta t RY_m M^T + \Delta t GY_{m-1} \widehat{M}^T + \Delta t HY_m \widehat{M}^T.$$

This case mimics the situation in the free atmosphere where there are no cut cells and the methods should be as stable as the split-explicit peer method. For the other stability diagram the matrix \widetilde{M} from (22) is used as Jacobian, i.e.

$$Y_m(I - \Delta t \gamma \widetilde{M})^T = BY_{m-1} + SY_m + \Delta t AY_{m-1} M^T + \Delta t RY_m M^T + \Delta t GY_{m-1} \widetilde{M}^T + \Delta t HY_m \widetilde{M}^T.$$

This case represents the situation in cut cells and because cut cells can become arbitrarily small the methods should be stable for arbitrary large advection and sound CFL numbers with the only restriction $\bar{u} < \frac{c_s}{6}$.

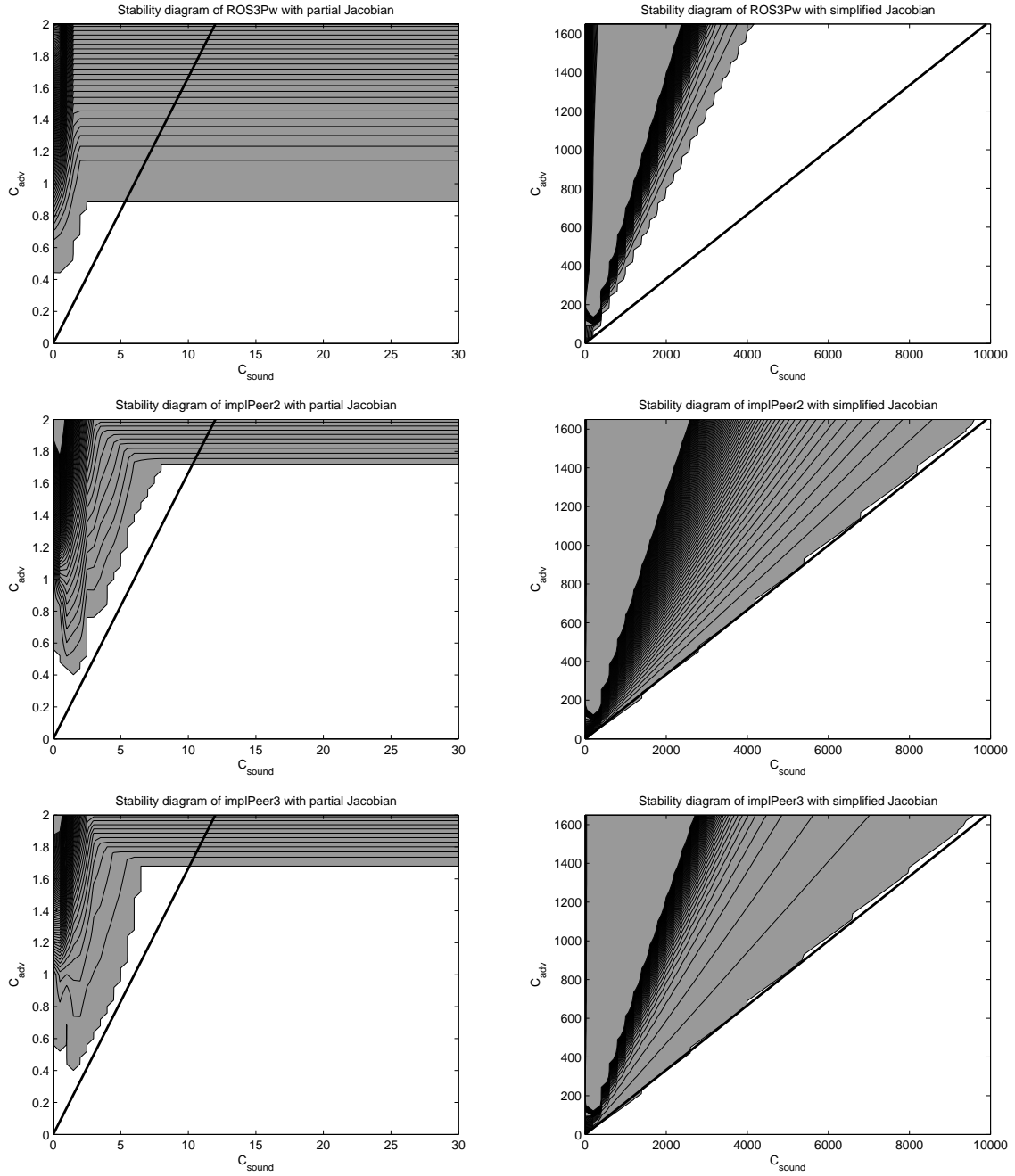


Figure 5: Stability diagrams for ROS3Pw (top), implPeer2 (middle) and implPeer3 (bottom) with partial Jacobian (left) and simplified Jacobian (right). Unstable regions in grey with contour interval 0.1.

We show the results for three methods which will be described in detail in Section 5.7.

- ROS3Pw is a second-order three-stage Rosenbrock W-method.
- implPeer2 is a second-order three-stage linearly implicit peer method which is based on the split-explicit peer method.
- implPeer3 is a third-order three-stage linearly implicit peer method.

Figure 5 shows the stability diagrams for the linearly implicit methods when there is no diffusion, i.e. for $\nu = 0$. As can be seen all methods are stable in the region of interest for arbitrary large CFL numbers if the simplified Jacobian \widetilde{M} is used, i.e. the methods should be stable in regions with cut cells. In the free atmosphere where \widehat{M} is used as Jacobian both linearly implicit peer methods are as stable as the split-explicit peer method, i.e. there is no restriction from the acoustic modes and the limit $C_{adv} \leq 1.7$. The Rosenbrock method has the stability limit $C_{adv} \leq 0.9$, i.e. it can compute stably with only half of the time step size which the peer methods can use.

4.5 Amplitude and phase properties

We now consider the amplitude and phase properties of the split-explicit and linearly implicit methods. The analytical solution of

$$\dot{y} = My$$

with M from (21) is

$$y(t_m) = \exp(\Delta t M)y(t_{m-1}).$$

When we apply the methods to the test equation the amplification matrix \overline{A} advances the solution to the next time level, i.e.

$$Y_m = \overline{A}Y_{m-1}.$$

If there is no diffusion M has one eigenvalue which results from the advection part and two eigenvalues incorporating advection and acoustics. Let λ be the eigenvalue of $\exp(\Delta t M)$ which originates from the pure advection part. Then $|\lambda|$ is the amplification factor and

$$\frac{\arctan \frac{\text{Im}\lambda}{\text{Re}\lambda}}{\text{Im}\lambda}$$

is the relative phase speed. Previously we only considered the maximum of the moduli of the eigenvalues of the amplification matrices \overline{A} in order to obtain the regions where the corresponding methods are stable. Each of these amplification matrices has nine eigenvalues because we consider three-stage methods which are applied to 3×3 -systems. One of these nine eigenvalues corresponds to the advection part which is advanced one time step, i.e. it corresponds to λ of $\exp(\Delta t M)$. For this eigenvalue amplitude and relative phase speed are analogously defined as for λ . These values are compared with the amplitude and relative phase speed of the analytical solution.

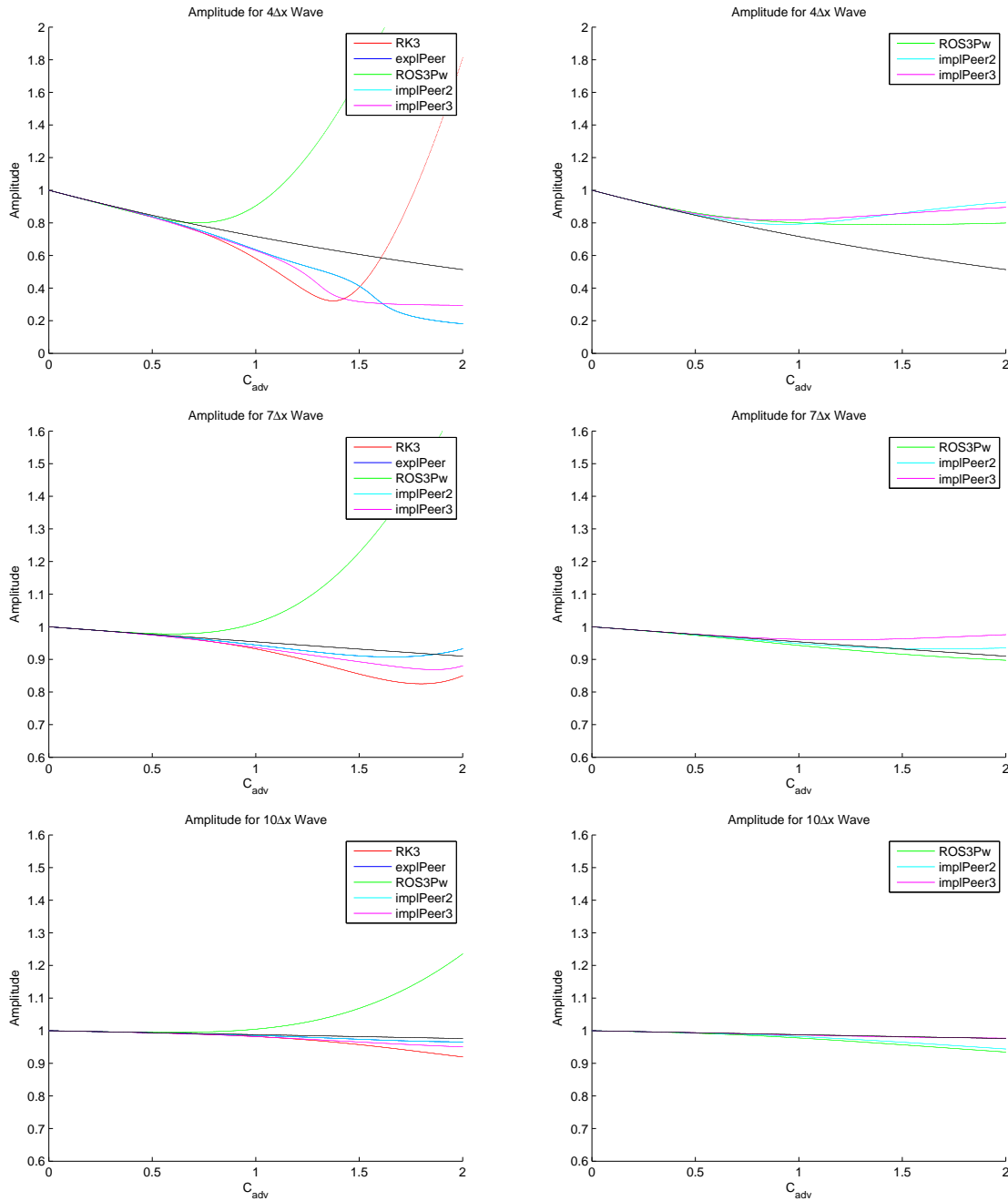


Figure 6: Amplitude for the $4\Delta x$ wave (top), the $7\Delta x$ wave (middle) and the $10\Delta x$ wave (bottom) for split-explicit methods and linearly implicit methods with partial Jacobian (left) and for linearly implicit methods with simplified Jacobian (right). The analytic amplitude in black.

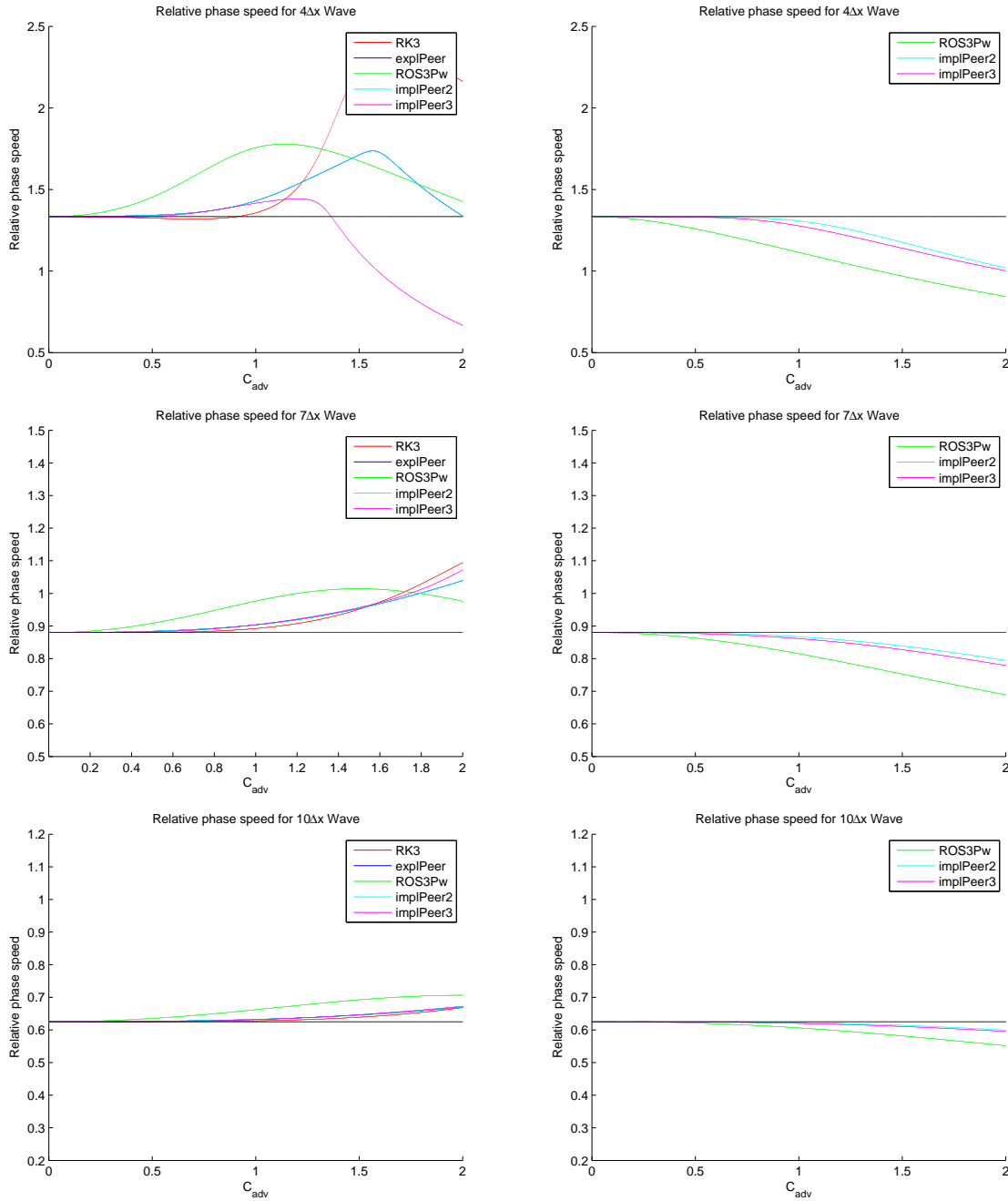


Figure 7: Relative phase speed for the $4\Delta x$ wave (top), the $7\Delta x$ wave (middle) and the $10\Delta x$ wave (bottom) for split-explicit methods and linearly implicit methods with partial Jacobian (left) and for linearly implicit methods with simplified Jacobian (right). The analytic relative phase speed in black.

Figure 6 shows the amplitudes for the considered methods and three wave numbers, Figure 7 shows the relative phase speeds. Because the split-explicit peer method `explPeer` is the underlying method for `implPeer2` and we consider the amplitude and phase properties of the advection part only, `implPeer2` with the partial Jacobian has the same amplitude and phase errors as `explPeer`, i.e. it adopts the good amplitude and phase properties of `explPeer`. We can see that the split-explicit methods and the linearly implicit peer methods with partial Jacobian have quite similar amplitude and phase properties while `ROS3Pw` with partial Jacobian has worse properties and becomes unstable for smaller CFL numbers in comparison to the other methods. When the simplified Jacobian is used for the linearly implicit methods `implPeer2` has the best properties. `implPeer3` has worse amplitude properties than `implPeer2` while `ROS3Pw` has larger phase errors. We will see the effects of these properties in Section 6.4. We can also see that the differences between the methods decrease if waves with larger wavelengths are considered because for them the eigenvalues are closer to 0.

5 Implementation

5.1 The compressible Euler equations

We consider the two-dimensional dry compressible Euler equations in conservative form with diffusion:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial z}, \quad (24)$$

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho u u}{\partial x} - \frac{\partial \rho w u}{\partial z} - \frac{\partial p}{\partial x} + \nu \rho \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right), \quad (25)$$

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho u w}{\partial x} - \frac{\partial \rho w w}{\partial z} - \frac{\partial p}{\partial z} - \rho g + \nu \rho \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial z^2} \right), \quad (26)$$

$$\frac{\partial \rho \theta}{\partial t} = -\frac{\partial \rho u \theta}{\partial x} - \frac{\partial \rho w \theta}{\partial z}. \quad (27)$$

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 p the pressure which is given diagnostically by the equation of state

$$p = \left(\frac{R_d \rho \theta}{p_0^\kappa} \right)^{\frac{1}{1-\kappa}} \quad (28)$$

where R_d is the gas constant for dry air, $\kappa = R_d/c_p$, c_p the heat capacity of dry air at constant pressure and p_0 is the pressure at ground level. The red terms belong to the advection part, the black terms are the acoustics and the diffusion is marked in blue. 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 Figure 8.

Because the atmosphere nearly is in hydrostatic equilibrium and the change of pressure is small in comparison to its absolute value we compute with its deviation from a background state which is in hydrostatic equilibrium, i.e. this background state satisfies

$$\frac{\partial \bar{p}}{\partial z} = -\bar{\rho} g.$$

So instead of computing with the black terms in (26) we implement

$$-\frac{\partial(p - \bar{p})}{\partial z} - (\rho - \bar{\rho})g.$$

In order to obtain linear systems of equations when using implicit methods like the trapezoidal rule for the fast part of split-explicit methods we need a linearized pressure as shown for the one-dimensional case in Section 4.1. Therefore we use the Exner pressure

$$\pi = \left(\frac{R_d \rho \theta}{p_0} \right)^{\frac{\kappa}{1-\kappa}} \quad (29)$$

and obtain

$$-\frac{R_d}{1-\kappa} \pi \frac{\partial \rho \theta}{\partial x} \quad \text{and} \quad -\frac{R_d}{1-\kappa} \pi \frac{\partial \rho \theta}{\partial z}$$

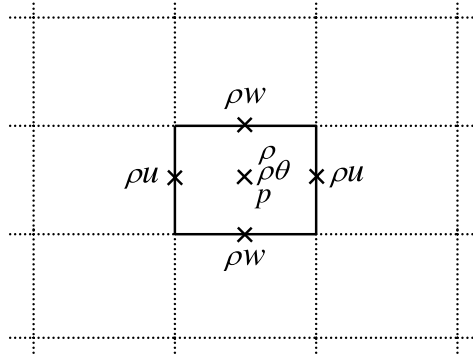


Figure 8: The positions of the variables on the Arakawa C grid.

instead of

$$-\frac{\partial p}{\partial x} \quad \text{and} \quad -\frac{\partial p}{\partial z},$$

i.e. we use the equations (24) to (27) as the right-hand side while the Jacobian originates from

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial z}, \quad (30)$$

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho u u}{\partial x} - \frac{\partial \rho w u}{\partial z} - \frac{R_d}{1-\kappa} \pi \frac{\partial \rho \theta}{\partial x} + \nu \rho \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right), \quad (31)$$

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho u w}{\partial x} - \frac{\partial \rho w w}{\partial z} - \frac{R_d}{1-\kappa} \pi \frac{\partial \rho \theta}{\partial z} - \rho g + \nu \rho \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial z^2} \right), \quad (32)$$

$$\frac{\partial \rho \theta}{\partial t} = -\frac{\partial \rho u \theta}{\partial x} - \frac{\partial \rho w \theta}{\partial z}. \quad (33)$$

As for the pressure p in the vertical momentum equation we compute without the background state, i.e. with

$$-\frac{R_d}{1-\kappa} \pi \frac{\partial(\rho \theta - \bar{\rho} \bar{\theta})}{\partial z} - \frac{R_d}{1-\kappa} (\pi - \bar{\pi}) \frac{\partial \bar{\rho} \bar{\theta}}{\partial z} - (\rho - \bar{\rho}) g.$$

For simplicity of notation we will use the formulation where the background state is not subtracted in the remainder of this thesis.

In Section 6.8 we will also present the application of the methods to the dam-break problem which is a test case for the shallow water equations. These can easily be generated from the compressible Euler equations by setting $\theta = 1$, removing the gravity term from the vertical momentum equation (26) and changing the constants in the equation of state for the pressure. In this formulation the equations for ρ and $\rho \theta$ are redundant and the shallow water equations read

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial z}, \\ \frac{\partial \rho u}{\partial t} &= -\frac{\partial \rho u u}{\partial x} - \frac{\partial \rho w u}{\partial z} - \frac{\partial p}{\partial x}, \\ \frac{\partial \rho w}{\partial t} &= -\frac{\partial \rho u w}{\partial x} - \frac{\partial \rho w w}{\partial z} - \frac{\partial p}{\partial z} \end{aligned}$$

with

$$p = \frac{g}{2} \rho^{\frac{1}{1-\kappa}}.$$

Now ρ is the depth of the water and u and w are the velocities.

The Euler equations are implemented in MATLAB on an Intel Core 2 Quad Q9550 @ 2833 Mhz with 3 GB RAM.

5.2 Implementation with finite volumes

Because we use cut cells for the representation of orography and because of the conservation properties we use a finite volume discretization for the compressible Euler equations as many weather services do. In order to illustrate the implementation of cut cells with finite volumes we use the mass equation (24) as an example for it. Integration of (24) over the volume V of one cell gives

$$\int_V \dot{\rho} dV = - \int_V \operatorname{div} \begin{pmatrix} \rho u \\ \rho w \end{pmatrix} dV.$$

Figure 9 shows this cell. Let n_5 be the normal vector of edge 5. We use free slip boundary conditions at the ground level, i.e. at the ground level the wind blows parallel to it. So it holds

$$\begin{pmatrix} \rho u \\ \rho w \end{pmatrix} \cdot n_5 = 0.$$

By assuming that the variables are constant in the cell Gauss' divergence theorem leads to

$$\begin{aligned} - \int_V \operatorname{div} \begin{pmatrix} \rho u \\ \rho w \end{pmatrix} dV &= - \oint_{\partial V} \begin{pmatrix} \rho u \\ \rho w \end{pmatrix} \cdot n d\partial V \\ &= - \int_{(\Delta x)_1} \begin{pmatrix} \rho u \\ \rho w \end{pmatrix} \cdot \begin{pmatrix} 0 \\ -1 \end{pmatrix} dx - \int_{(\Delta z)_2} \begin{pmatrix} \rho u \\ \rho w \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} dz \\ &\quad - \int_{(\Delta x)_3} \begin{pmatrix} \rho u \\ \rho w \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} dx - \int_{(\Delta z)_4} \begin{pmatrix} \rho u \\ \rho w \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 0 \end{pmatrix} dz \\ &= (\rho w)_1 (\Delta x)_1 - (\rho u)_2 (\Delta z)_2 - (\rho w)_3 (\Delta x)_3 + (\rho u)_4 (\Delta z)_4. \end{aligned}$$

Using

$$\int_V \dot{\rho} dV = \dot{\rho}_6 V$$

and dividing by V results in the spatial discretization of the mass equation

$$\dot{\rho}_6 = -(\rho u)_2 \frac{(\Delta z)_2}{V} + (\rho u)_4 \frac{(\Delta z)_4}{V} - (\rho w)_3 \frac{(\Delta x)_3}{V} + (\rho w)_1 \frac{(\Delta x)_1}{V}.$$

We now present the spatial discretization of (24) to (27) with finite volumes. For this purpose we introduce three notations for the variables: If they are written in black

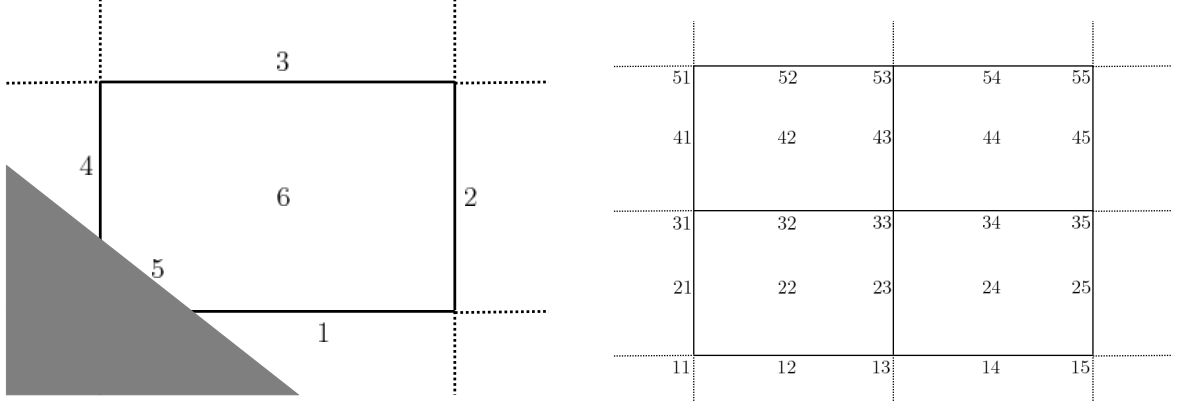


Figure 9: One cell with orography (left) and four cells (right) of the grid.

without a tilde they are defined at the locations of their indices which are shown in Figure 9. Tilde variables are obtained by averaging, e.g.

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

If the variable is written in red it is generated at the location of its index by the third-order upwind scheme (19). θ can simply be obtained by dividing $\rho\theta$ by 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.

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

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

$$\begin{aligned} \dot{\rho}_{22} &= -(\rho u)_{23} \frac{(\Delta z)_{23}}{V_{22}} + (\rho u)_{21} \frac{(\Delta z)_{21}}{V_{22}} - (\rho w)_{32} \frac{(\Delta x)_{32}}{V_{22}} + (\rho w)_{12} \frac{(\Delta x)_{12}}{V_{22}}, \\ (\dot{\rho} u)_{23} &= -(\tilde{\rho} u)_{24} u_{24} \frac{(\Delta z)_{24}}{V_{23}} + (\tilde{\rho} u)_{22} u_{22} \frac{(\Delta z)_{22}}{V_{23}} - (\tilde{\rho} w)_{33} w_{33} \frac{(\Delta x)_{33}}{V_{23}} + (\tilde{\rho} w)_{13} w_{13} \frac{(\Delta x)_{13}}{V_{23}} \\ &\quad - p_{24} \frac{(\Delta z)_{23}}{V_{23}} + p_{22} \frac{(\Delta z)_{23}}{V_{23}} + \nu \tilde{\rho}_{23} \Delta u, \\ (\dot{\rho} w)_{32} &= -(\tilde{\rho} w)_{33} w_{33} \frac{(\Delta z)_{33}}{V_{32}} + (\tilde{\rho} w)_{31} w_{31} \frac{(\Delta z)_{31}}{V_{32}} - (\tilde{\rho} w)_{42} w_{42} \frac{(\Delta x)_{42}}{V_{32}} + (\tilde{\rho} w)_{22} w_{22} \frac{(\Delta x)_{22}}{V_{32}} \\ &\quad - p_{42} \frac{(\Delta x)_{32}}{V_{32}} + p_{22} \frac{(\Delta x)_{32}}{V_{32}} - \tilde{\rho}_{32} g + \nu \tilde{\rho}_{32} \Delta w, \\ (\dot{\rho} \theta)_{22} &= -(\rho u)_{23} \theta_{23} \frac{(\Delta z)_{23}}{V_{22}} + (\rho u)_{21} \theta_{21} \frac{(\Delta z)_{21}}{V_{22}} - (\rho w)_{32} \theta_{32} \frac{(\Delta x)_{32}}{V_{22}} + (\rho w)_{12} \theta_{12} \frac{(\Delta x)_{12}}{V_{22}}. \end{aligned}$$

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

$$-p_{24} \frac{(\Delta z)_{24}}{V_{23}} + p_{22} \frac{(\Delta z)_{22}}{V_{23}} \quad \text{and} \quad -p_{42} \frac{(\Delta x)_{42}}{V_{32}} + p_{22} \frac{(\Delta x)_{22}}{V_{32}}$$

	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%

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

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 $(\Delta x)_{22}$.

This finite volume implementation of the compressible Euler equations in flux form conserves mass and entropy. If we compute without diffusion, with the pressure p and periodic boundary conditions the momentum is conserved, too. This is not the case if we use the Exner pressure π which is the reason why we prefer to use the equations (24) to (27) as the right-hand side instead of (30) to (33).

5.3 Computation of the Jacobian

We now present the details of the implementation of the Jacobian, which is needed for linearly implicit methods and the trapezoidal rule. When adapting the simplifications of the Jacobian from the one-dimensional Euler equations, as explained in Section 4.2, for two dimensions we obtain the matrix which results from the spatial discretization of

$$- \begin{pmatrix} \mathbf{u} & 1 & 0 & 0 \\ 0 & \mathbf{u} & 0 & \frac{R_d}{1-\kappa}\pi \\ 0 & 0 & \mathbf{u} & 0 \\ 0 & \theta & 0 & \mathbf{u} \end{pmatrix} \begin{pmatrix} \rho_x \\ (\rho u)_x \\ (\rho w)_x \\ (\rho \theta)_x \end{pmatrix} - \begin{pmatrix} \mathbf{w} & 0 & 1 & 0 \\ 0 & \mathbf{w} & 0 & 0 \\ 0 & 0 & \mathbf{w} & \frac{R_d}{1-\kappa}\pi \\ 0 & 0 & \theta & \mathbf{w} \end{pmatrix} \begin{pmatrix} \rho_z \\ (\rho u)_z \\ (\rho w)_z \\ (\rho \theta)_z \end{pmatrix} - \begin{pmatrix} 0 \\ -\nu \Delta(\rho u) \\ \rho g - \nu \Delta(\rho w) \\ 0 \end{pmatrix}$$

as Jacobian. The red terms originate from the advection part, the black terms from the acoustics and the blue terms represent the diffusion. So the whole matrix is used as Jacobian for linearly implicit methods in cut cell regions while in the free atmosphere the red and blue terms are neglected. The trapezoidal rule computes with the black and blue terms.

Table 1 shows the amount of memory which is needed for the Jacobian per grid cell, i.e. the number of non-zero entries in four rows, respectively three rows in 1D and five rows in 3D, without the consideration of gravity and diffusion. While central differences and the first-order upwind scheme need two entries per derivative the third-order upwind scheme needs at least four entries. Therefore the simplified Jacobian only needs 41% of the memory 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 level, cut cells can be located only there, e.g. in a model with 50 vertical layers only 2% of the cells in the whole domain are cut cells which is negligible from the memory point of view. So the right column of Table 1 shows the ratio of memory when using the full but simplified Jacobian in cut cells only instead of everywhere.

We now state the Jacobian for our spatial discretization with finite volumes. As explained in Section 4.2 the advection terms are discretized with the first-order upwind

scheme while the acoustics uses central differences. The diffusion is treated with second-order finite differences instead of finite volumes. We define

$$J := \begin{pmatrix} J_{\rho\rho} & J_{\rho u} & J_{\rho w} & 0 \\ 0 & J_{uu} + J_{\Delta u} & 0 & J_{u\theta} \\ J_{w\rho} & 0 & J_{ww} + J_{\Delta w} & J_{w\theta} \\ 0 & J_{\theta u} & J_{\theta w} & J_{\theta\theta} \end{pmatrix} \quad (34)$$

and assume negative wind speeds, i.e. if variables are interpolated with the first-order upwind scheme they will be shifted to the left (or downwards respectively) to the locations where they are needed for the advection terms. The notation for averaged variables and indices is the same as in Section 5.2. For the advection we also need the densities at cell vertices. This is done again with the arithmetic mean, e.g.

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

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

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

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

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

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:

$$\begin{aligned} J_{\rho\rho}(22, 24) &= -u_{23} \frac{(\Delta z)_{23}}{V_{22}}, & J_{\rho\rho}(22, 22) &= u_{21} \frac{(\Delta z)_{21}}{V_{22}}, \\ J_{\rho\rho}(22, 42) &= -w_{32} \frac{(\Delta x)_{32}}{V_{22}}, & J_{\rho\rho}(22, 22) &= w_{12} \frac{(\Delta x)_{12}}{V_{22}}, \\ J_{\rho u}(22, 23) &= -\frac{(\Delta z)_{23}}{V_{22}}, & J_{\rho u}(22, 21) &= \frac{(\Delta z)_{21}}{V_{22}}, \\ J_{\rho w}(22, 32) &= -\frac{(\Delta x)_{32}}{V_{22}}, & J_{\rho w}(22, 12) &= \frac{(\Delta x)_{12}}{V_{22}}, \\ J_{uu}(23, 25) &= -\frac{(\tilde{\rho u})_{24}}{\rho_{24}} \frac{(\Delta z)_{24}}{V_{23}}, & J_{uu}(23, 23) &= \frac{(\tilde{\rho u})_{22}}{\rho_{22}} \frac{(\Delta z)_{22}}{V_{23}}, \\ J_{uu}(23, 43) &= -\frac{(\tilde{\rho w})_{33}}{\tilde{\rho}_{33}} \frac{(\Delta x)_{33}}{V_{23}}, & J_{uu}(23, 23) &= \frac{(\tilde{\rho w})_{13}}{\tilde{\rho}_{13}} \frac{(\Delta x)_{13}}{V_{23}}, \end{aligned}$$

$$\begin{aligned}
J_{u\theta}(23, 24) &= -\frac{R_d}{1-\kappa} \tilde{\pi}_{23} \frac{(\Delta z)_{23}}{V_{23}}, & J_{u\theta}(23, 22) &= \frac{R_d}{1-\kappa} \tilde{\pi}_{23} \frac{(\Delta z)_{23}}{V_{23}}, \\
J_{w\rho}(32, 42) &= -\frac{g}{2}, & J_{w\rho}(32, 22) &= -\frac{g}{2}, \\
J_{ww}(32, 34) &= -\frac{(\tilde{\rho}u)_{33}}{\tilde{\rho}_{33}} \frac{(\Delta z)_{33}}{V_{32}}, & J_{ww}(32, 32) &= \frac{(\tilde{\rho}u)_{31}}{\tilde{\rho}_{31}} \frac{(\Delta z)_{31}}{V_{32}}, \\
J_{ww}(32, 52) &= -\frac{(\tilde{\rho}w)_{42}}{\rho_{42}} \frac{(\Delta x)_{42}}{V_{32}}, & J_{ww}(32, 32) &= \frac{(\tilde{\rho}w)_{22}}{\rho_{22}} \frac{(\Delta x)_{22}}{V_{32}}, \\
J_{w\theta}(32, 42) &= -\frac{R_d}{1-\kappa} \tilde{\pi}_{32} \frac{(\Delta x)_{32}}{V_{32}}, & J_{w\theta}(32, 22) &= \frac{R_d}{1-\kappa} \tilde{\pi}_{32} \frac{(\Delta x)_{32}}{V_{32}}, \\
J_{\theta u}(22, 23) &= -\theta_{24} \frac{(\Delta z)_{23}}{V_{22}}, & J_{\theta u}(22, 21) &= \theta_{22} \frac{(\Delta z)_{21}}{V_{22}}, \\
J_{\theta w}(22, 32) &= -\theta_{42} \frac{(\Delta x)_{32}}{V_{22}}, & J_{\theta w}(22, 12) &= \theta_{22} \frac{(\Delta x)_{12}}{V_{22}}, \\
J_{\theta\theta}(22, 24) &= -u_{23} \frac{(\Delta z)_{23}}{V_{22}}, & J_{\theta\theta}(22, 22) &= u_{21} \frac{(\Delta z)_{21}}{V_{22}}, \\
J_{\theta\theta}(22, 42) &= -w_{32} \frac{(\Delta x)_{32}}{V_{22}}, & J_{\theta\theta}(22, 22) &= w_{12} \frac{(\Delta x)_{12}}{V_{22}}, \\
J_{\Delta u}(23, 25) &= \frac{\nu}{(\Delta x)^2}, & J_{\Delta u}(23, 23) &= -\frac{2\nu}{(\Delta x)^2} - \frac{2\nu}{(\Delta z)^2}, \\
J_{\Delta u}(23, 21) &= \frac{\nu}{(\Delta x)^2}, & J_{\Delta w}(32, 52) &= \frac{\nu}{(\Delta z)^2}, \\
J_{\Delta w}(32, 32) &= -\frac{2\nu}{(\Delta x)^2} - \frac{2\nu}{(\Delta z)^2}, & J_{\Delta w}(32, 12) &= \frac{\nu}{(\Delta z)^2}.
\end{aligned}$$

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.

5.4 Implementation of split-explicit methods

We now show how split-explicit Runge-Kutta methods (8) are applied to the compressible Euler equations (24) to (27) with the forward-backward Euler scheme or the trapezoidal rule for the integration of the fast part (2).

We start with the forward-backward Euler scheme from [5]. A detailed description can be found in [21]. It consists of a forward Euler step for one part of the differential equation followed by a backward Euler step for the other part. The backward step is explicit because only the parts of the Euler equations are needed for it which are known from the forward step. In order to present the concrete implementation for the Euler equations we use two indices for the variables: The lower case index denotes the time level and stage of the underlying split-explicit Runge-Kutta method while the upper case index indicates the time level of the forward-backward Euler scheme. The initial values

(3) are

$$\begin{aligned}\rho^{(0)} &= \rho_{m-1,s+1}, \\ (\rho u)^{(0)} &= (\rho u)_{m-1,s+1}, \\ (\rho w)^{(0)} &= (\rho w)_{m-1,s+1}, \\ (\rho\theta)^{(0)} &= (\rho\theta)_{m-1,s+1}.\end{aligned}$$

One forward Euler step with time step size $\Delta\tau = \frac{\Delta t}{n_s}$ is

$$\begin{aligned}(\rho u)^{(n+1)} &= (\rho u)^{(n)} + \Delta\tau \left(\sum_{j=1}^{i-1} \frac{a_{ij}}{c_i} \left(-\frac{\partial(\rho u u)_{mj}}{\partial x} - \frac{\partial(\rho w u)_{mj}}{\partial z} \right) - \frac{\partial p^{(n)}}{\partial x} + \nu \rho^{(n)} \Delta u^{(n)} \right), \\ (\rho w)^{(n+1)} &= (\rho w)^{(n)} + \Delta\tau \left(\sum_{j=1}^{i-1} \frac{a_{ij}}{c_i} \left(-\frac{\partial(\rho u w)_{mj}}{\partial x} - \frac{\partial(\rho w w)_{mj}}{\partial z} \right) - \frac{\partial p^{(n)}}{\partial z} - \rho^{(n)} g + \nu \rho^{(n)} \Delta w^{(n)} \right)\end{aligned}$$

followed by the backward step

$$\begin{aligned}\rho^{(n+1)} &= \rho^{(n)} + \Delta\tau \left(-\frac{\partial(\rho u)^{(n+1)}}{\partial x} - \frac{\partial(\rho w)^{(n+1)}}{\partial z} \right), \\ (\rho\theta)^{(n+1)} &= (\rho\theta)^{(n)} + \Delta\tau \sum_{j=1}^{i-1} \frac{a_{ij}}{c_i} \left(-\frac{\partial(\rho u)^{(n+1)}\theta_{mj}}{\partial x} - \frac{\partial(\rho w)^{(n+1)}\theta_{mj}}{\partial z} \right).\end{aligned}$$

The number of forward-backward Euler steps per stage of the underlying Runge-Kutta method is

$$\frac{c_i \Delta t}{\Delta\tau}.$$

In fact $\Delta\tau$ is chosen close to $\frac{\Delta t}{n_s}$ so that this number is natural. In practice the spatial derivatives are discretized first and the advection terms, i.e. the terms which are constant during the forward-backward steps, are computed only once per stage of the underlying method. Split-explicit peer methods (8) with the forward-backward Euler scheme are implemented analogously.

When using the trapezoidal rule for integration of the fast parts (2) and (6), the Euler equations (30) to (33) with the Exner pressure are considered. In principle the trapezoidal rule uses the same Jacobian, i.e. the black and blue parts of (34), as linearly implicit methods. Besides the fact that diffusion is treated as the fast part for split-explicit methods in contrast to linearly implicit methods, which only use the acoustic parts of (34) in the free atmosphere, there is another difference: This difference results from the nonlinearity of the pressure derivatives in the momentum equations (31) and (32) and the derivatives in the temperature equation (33). As explained in Section 5.3 the Jacobian for linearly implicit methods has the entries

$$\begin{aligned}J_{u\theta} &= -\frac{R_d}{1-\kappa} \pi \frac{\Delta z}{V}, \\ J_{w\theta} &= -\frac{R_d}{1-\kappa} \pi \frac{\Delta x}{V},\end{aligned}$$

$$J_{\theta u} = -\theta \frac{\Delta z}{V},$$

$$J_{\theta w} = -\theta \frac{\Delta x}{V}.$$

For split-explicit methods we use linear combinations of π and θ which originate from the underlying explicit method, e.g. instead of the above entries split-explicit Runge-Kutta methods use

$$L_{u\theta} := -\frac{R_d}{1-\kappa} \sum_{j=1}^{i-1} \frac{a_{ij}}{c_i} \pi_{mj} \frac{\Delta z}{V},$$

$$L_{w\theta} := -\frac{R_d}{1-\kappa} \sum_{j=1}^{i-1} \frac{a_{ij}}{c_i} \pi_{mj} \frac{\Delta x}{V},$$

$$L_{\theta u} := -\sum_{j=1}^{i-1} \frac{a_{ij}}{c_i} \theta_{mj} \frac{\Delta z}{V},$$

$$L_{\theta w} := -\sum_{j=1}^{i-1} \frac{a_{ij}}{c_i} \theta_{mj} \frac{\Delta x}{V}$$

as Jacobian. Because of these differences we use the notation L for the Jacobian for split-explicit methods. The other entries of L are equal to entries of the Jacobian J for linearly implicit methods. Because we use equal time step sizes $\Delta\tau = \Delta t$ we have to solve one linear system of equations

$$\left(I - \frac{c_i \Delta t}{2} L\right) y_{mi} = \left(I + \frac{c_i \Delta t}{2} L\right) y_{m-1, s+1} + \Delta t \sum_{j=1}^{i-1} a_{ij} \begin{pmatrix} 0 & \\ -\frac{\partial(\rho u)_{mj}}{\partial x} & -\frac{\partial(\rho w)_{mj}}{\partial z} \\ \frac{\partial(\rho u)_{mj}}{\partial x} & -\frac{\partial(\rho w)_{mj}}{\partial z} \\ 0 & 0 \end{pmatrix}$$

in each stage of a split-explicit Runge-Kutta method. The implementation of split-explicit peer methods with the trapezoidal rule is analogous.

5.5 Initialization of orography

When computing with orography one problem is the initialization of it. A mountain which is instantaneously inserted into an atmosphere that is at rest will cause perturbations in the pressure. This is illustrated by Figure 10 which shows the pressure perturbation cause by the Witch of Agnesi mountain of test case 3. One widespread ansatz to avoid this problem is to use Rayleigh damping. This means that the computing domain is expanded and in the additional domain the variables will be artificially forced back to the background state. Due to the larger domain this approach is not very efficient. Therefore we use another strategy.

We change the initial atmosphere by computing a wind field that follows the shape of the mountain. This wind field is obtained by performing one backward Euler step with

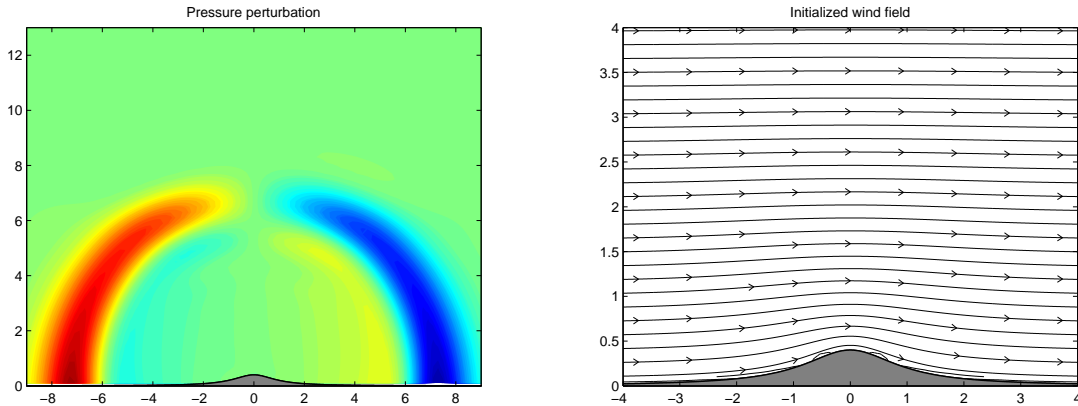


Figure 10: Pressure perturbation after 20 s (left) and initialized wind field (right).

sufficiently large time step size. We solve

$$(I - \Delta t J)y = \begin{pmatrix} 0 \\ \rho u \\ \rho w \\ 0 \end{pmatrix}$$

with J the partial Jacobian which only contains the acoustic parts, i.e. the black terms in (34). Only the winds are updated with this solution, otherwise one would also adopt the disturbance. The result of this ansatz can be seen in Figure 10. It is quite good and we do not need to expand the domain in order to damp the disturbance.

5.6 Method search

We now describe how we obtained the coefficients of the considered peer methods. For the split-explicit peer method our aim was to construct a method (8) with $s = 3$ stages, order $p = s - 1 = 2$ and stability properties comparable to those of the split-explicit Runge-Kutta method RK3. For these aims we have the coefficient matrices B , S , A , R and the vectors c and α . As described in Section 3.1 we use $s(s - 1)$ coefficients of A to satisfy the classical peer order conditions and the remaining s degrees of freedom are expressed with a vector β . The additional split order conditions are fulfilled by B . Furthermore we define $c_s = 1$ so that the last stage is an approximation to the analytical solution at time points t_m . Therefore we have the 14 remaining parameters c_1 , c_2 , α_1 , α_2 , α_3 , β_1 , β_2 , β_3 , s_{21} , s_{31} , s_{32} , r_{21} , r_{31} and r_{32} as degrees of freedom in the optimization process.

For the optimization we use a Monte Carlo search strategy in this 14-dimensional subspace of second-order three-stage split-explicit peer methods: A set of 1000 peer methods is generated from the degrees of freedom. 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 does not change during ten steps the algorithm aborts.

The optimization function consists of two parts: The first aim is stability. The method should be as stable without any damping as RK3 is with divergence damping, i.e. stability for advection CFL numbers up to 1.7 and sound CFL numbers up to 90% of n_s , when the forward-backward Euler scheme is used for integration of the acoustic part. This criterion is verified by considering the maximum eigenvalue of the amplification matrix \bar{A} in dependency of the sound CFL number for a fixed number of small time steps $n_s = 30$, the advection CFL number is chosen as 0 and the minimum of $C_{sound}/6$ and 1.7. It has been shown in practice that methods which are stable for these advection CFL numbers are stable for values between these two extrema, too. Furthermore they are also stable up to $C_{sound} < 0.9n_s$ for other numbers of n_s . The second aim is to obtain methods which have small integration intervals α_i of the fast part (6) in order to be computationally efficient. During the first part of the optimization process the optimization function only contains the stability aim. If it is fulfilled $\|\alpha\|_1$ is optimized while the achieved stability property remains as a side condition. The stability criteria are verified with the functions

$$\tilde{\varphi}(C_{sound}) := \left(\begin{array}{c} \|\text{eig}(\bar{A}(0, C_{sound}))\|_{\infty} \\ \|\text{eig}(\bar{A}(\min(C_{sound}/6, 1.7), C_{sound}))\|_{\infty} \end{array} \right)$$

and

$$\hat{\varphi} := \max\{0 \leq C_{sound} \leq 27 : \|\tilde{\varphi}(C_{sound})\|_{\infty} \leq 1\} - 27$$

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

$$\varphi := \begin{cases} \hat{\varphi} & \text{if } \hat{\varphi} < 0 \\ 1/\|\alpha_1\| & \text{if } \hat{\varphi} = 0 \end{cases}$$

and we optimize with respect to the global maximum of φ . With this search strategy we found the method explPeer.

For the construction of linearly implicit peer methods (10) we optimize the remaining degrees of freedom with respect to amplitude and phase errors which are explained in Section 4.5. As a side condition we claim that the method has to be stable for wind speeds $\bar{u} < c_s/6$ in the following situations:

- In the free atmosphere only the acoustic part \widehat{M} is used as Jacobian. The peer method should be as stable as explPeer, i.e. it should be stable for $C_{adv} < 1.7$ and arbitrary large sound CFL numbers. Because stability is determined numerically this condition is verified for CFL numbers up to 10000.
- In cut cells the matrix \widetilde{M} is used as Jacobian and the peer method should be stable for arbitrary large advection CFL numbers and sound CFL numbers, in practice up to 10000.

Let \widehat{A} be the amplification matrix when the partial Jacobian \widehat{M} is used and \widetilde{A} is its notation when the simplified Jacobian \widetilde{M} is used. For linearly implicit peer methods we use the functions

$$\tilde{\varphi}(C_{sound}) := \left(\begin{array}{c} \|\text{eig}(\widehat{A}(0, C_{sound}))\|_{\infty} \\ \|\text{eig}(\widehat{A}(\min(C_{sound}/6, 1.7), C_{sound}))\|_{\infty} \\ \|\text{eig}(\widetilde{A}(0, C_{sound}))\|_{\infty} \\ \|\text{eig}(\widetilde{A}(C_{sound}/6, C_{sound}))\|_{\infty} \end{array} \right)$$

and

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

to verify stability. 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 optimize with respect to the global maximum of φ . We found the three-stage methods implPeer2 and implPeer3. For the second-order peer method implPeer2 we only optimized the coefficients of the implicit part, i.e. γ , G and H , while the coefficients of the explicit part are those of explPeer in order to adopt its good stability, amplitude and phase properties. For implPeer3 we optimized all coefficients and used the conditions for order of consistency 2 and the additional conditions for superconvergence from Theorem 7 as side conditions, i.e. this method has order of convergence 3.

5.7 Coefficients of the methods

We now give an overview of the properties of the methods considered in this thesis and their coefficients.

The most important properties are shown in Table 2. While all methods have three stages, implPeer3 is the only method which is third-order due to its superconvergence property. ROS3Pw has order 3 if it uses an exact Jacobian but with the partial or the full but simplified Jacobian its order is reduced to 2. In order to be applicable to the compressible Euler equations with reasonable time step sizes RK3 has to use divergence damping when using the forward-backward Euler scheme or off-centering for the trapezoidal rule. Divergence damping alters the Euler equations in such a way that the error of the numerical scheme diverges for sufficiently small time step sizes. The effect of the off-centering is the reduction of the order to 1. In contrast to RK3 the peer method explPeer does not have these disadvantages because it can stably integrate the compressible Euler equations without the need for artificial damping or off-centering. Furthermore the sum of the fast integration intervals is 20% smaller when using explPeer in comparison to RK3. In combination with the forward-backward Euler scheme for the integration of the fast part this results in computational efficiency.

All presented linearly implicit methods are stable for arbitrary large advection and sound CFL numbers if the simplified Jacobian is used, which makes them implementable with cut cell grids. Furthermore they are also stable when the Jacobian only contains the acoustic part, i.e. the full but simplified Jacobian is needed in cut cell regions only. Due to the better stability properties of the linearly implicit peer methods with partial Jacobian they can compute with the double time step size that ROS3Pw can use with the partial Jacobian.

5.7.1 The split-explicit Runge-Kutta method RK3

Because the second-order three-stage split-explicit Runge-Kutta method RK3 is a widely used solver for the compressible Euler equations and implemented in several weather

Method	s	p	$\ \alpha\ _1$	C_{adv}	C_{sound}
RK3-FB	3	2 (-)	1.83	1.7	3 (29)
RK3-TR	3	2 (1)	1.83	1.7	1 (∞)
explPeer-FB	3	2	1.44	1.7	30
explPeer-TR	3	2	1.44	1.7	∞
ROS3Pw	3	2	-	0.9 / ∞	∞ / ∞
implPeer2	3	2	-	1.7 / ∞	∞ / ∞
implPeer3	3	3	-	1.7 / ∞	∞ / ∞

Table 2: Properties of the considered methods. s is the number of stages, p the order of convergence, $\|\alpha\|_1$ the sum of the fast integration intervals for split-explicit methods, C_{adv} and C_{sound} are the advection and acoustic stability limits for split-explicit methods and for linearly implicit methods with partial/simplified Jacobian under the restriction $\bar{u} < \frac{c_s}{6}$. Properties of RK3 with divergence damping respectively off-centering in parentheses. $n_s = 30$ for FB respectively $n_s = 1$ for TR.

codes we compare our split-explicit peer method with it. Written as a peer method (8) its coefficients are

$$\begin{aligned}
c &= \left(\begin{array}{ccc} 0.3333333333333333 & 0.5000000000000000 & 1.0000000000000000 \end{array} \right)^T, \\
\alpha &= \left(\begin{array}{ccc} 0.3333333333333333 & 0.5000000000000000 & 1.0000000000000000 \end{array} \right)^T, \\
B &= \left(\begin{array}{ccc} 0 & 0 & 1.0000000000000000 \\ 0 & 0 & 1.0000000000000000 \\ 0 & 0 & 1.0000000000000000 \end{array} \right), \\
S &= \left(\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right), \\
A &= \left(\begin{array}{ccc} 0 & 0 & 0.3333333333333333 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right), \\
R &= \left(\begin{array}{ccc} 0 & 0 & 0 \\ 0.5000000000000000 & 0 & 0 \\ 0 & 1.0000000000000000 & 0 \end{array} \right).
\end{aligned}$$

5.7.2 The split-explicit peer method explPeer

We constructed the second-order three-stage split-explicit peer method explPeer in order to obtain a method which can stably integrate the compressible Euler equations without the need for artificial damping. Its coefficients are

$$\begin{aligned}
c &= \left(\begin{array}{ccc} -0.0899531627878552 & 0.4676428830697650 & 1.0000000000000000 \end{array} \right)^T, \\
\alpha &= \left(\begin{array}{ccc} 0.0663272206869391 & 0.5550418090653669 & 0.8254622965775626 \end{array} \right)^T,
\end{aligned}$$

$$\begin{aligned}
B &= \begin{pmatrix} -0.0967059983845656 & 0.4915598645202344 & 0.6051461338643311 \\ -0.0470929826281593 & 0.2169946581702936 & 0.5720815963722115 \\ -0.0891437312845480 & 0.1573830315884013 & 0.1973233392586685 \end{pmatrix}, \\
S &= \begin{pmatrix} 0 & 0 & 0 \\ 0.2580167280856541 & 0 & 0 \\ 0.3269306113397434 & 0.4075067490977347 & 0 \end{pmatrix}, \\
A &= \begin{pmatrix} 0.0721007322008575 & -0.1322804288331288 & 0.1265069173192104 \\ 0.0478238719665258 & -0.4831372398722279 & -0.1163332106046261 \\ 0.0325906971440313 & 0.0702440095890842 & 0.1286761505892647 \end{pmatrix}, \\
R &= \begin{pmatrix} 0 & 0 & 0 \\ 1.1066883875756954 & 0 & 0 \\ -0.5020271673748957 & 1.0959786066300778 & 0 \end{pmatrix}.
\end{aligned}$$

5.7.3 The Rosenbrock method ROS3Pw

The three-stage Rosenbrock method ROS3Pw has been developed for parabolic partial differential equations and parabolic partial differential algebraic equations of index 1 with the aim to obtain a third-order Rosenbrock method which is robust even if only an approximation of the Jacobian is used. ROS3Pw has been presented in [25] and was successfully tested with the Navier-Stokes equations amongst others. While popular Rosenbrock methods like RODAS ([10]), RODASP ([33]) or ROS3P ([20]) are ROW-methods, i.e. the order decreases to 1, and they are not stable if an approximate Jacobian is used, ROS3Pw is a W-method. Contrariwise to those ROW-methods ROS3Pw is second-order and stable if the partial Jacobian is used. For these reasons we decided to compare the linearly implicit peer methods with ROS3Pw.

We implemented ROS3Pw as a linearly implicit peer method. While peer methods (10) use the same solution vectors Y_{mi} for the evaluation of the right-hand side $f(Y_{mi})$ and the Jacobian-vector products JY_{mi} Rosenbrock methods

$$\begin{aligned}
(I - \Delta t \gamma J)k_i &= \Delta t f\left(Y_{m-1,s} + \sum_{j=1}^{i-1} a_{ij}k_j\right) + \Delta t \sum_{j=1}^{i-1} \gamma_{ij} Jk_j \\
Y_{ms} &= Y_{m-1,s} + \sum_{j=1}^s b_j k_j
\end{aligned}$$

use different solution vectors $Y_{m-1,s} + \sum_{j=1}^{i-1} a_{ij}k_j$ and k_j for the evaluation of the right-hand side and the Jacobian-vector products. In order to write a Rosenbrock method as a peer method we have to use the solution vectors $Y_{mi} := Y_{m-1,s} + \sum_{j=1}^i a_{i+1,j}k_j$ and express the slopes k_i as linear combinations of $Y_{m-1,i}$ and Y_{mi} . The Butcher tableau of ROS3Pw is

0	
1.577350269189625	1.577350269189625
0.5	0.5
	0.105662432702593 0.049038105676657 0.845299461620748

and therefore

$$Y_m = \mathbb{1}Y_{m-1,s}^T + \begin{pmatrix} 1.577350269189625 & 0 & 0 \\ 0.5 & 0 & 0 \\ 0.105662432702593 & 0.049038105676657 & 0.845299461620748 \end{pmatrix} \begin{pmatrix} k_1^T \\ k_2^T \\ k_3^T \end{pmatrix}.$$

Obviously it is not possible to solve these equations for k_i . For this reason we added an additional stage, i.e. we implemented the three-stage Rosenbrock method ROS3Pw as a four-stage linearly implicit peer method. Its coefficients are

$$\begin{aligned} c &= \begin{pmatrix} 0 & 1.577350269189625 & 0.500000000000000 & 1.000000000000000 \end{pmatrix}^T, \\ \gamma &= \begin{pmatrix} 0.7886751345948128 & 0.7886751345948128 & 0.7886751345948128 & 0.7886751345948128 \end{pmatrix}^T, \\ B &= \begin{pmatrix} 0 & 0 & 0 & 1.000000000000000 \\ 0 & 0 & 0 & 1.000000000000000 \\ 0 & 0 & 0 & 0.6830127018922192 \\ 0 & 0 & 0 & 0.8839745962155612 \end{pmatrix}, \\ S &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.3169872981077807 & 0 & 0 & 0 \\ 0.0669872981077806 & 0.0490381056766579 & 0 & 0 \end{pmatrix}, \\ A &= \begin{pmatrix} 0 & 0 & 0 & 1.5773502691896256 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ R &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1.000000000000000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8452994616207484 & 0 \end{pmatrix}, \\ G &= \begin{pmatrix} 0 & 0 & 0 & -0.7886751345948128 \\ 0 & 0 & 0 & 0.2113248654051871 \\ 0 & 0 & 0 & -0.5386751345948128 \\ 0 & 0 & 0 & -0.1933756729740644 \end{pmatrix}, \\ H &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1.000000000000000 & 0 & 0 & 0 \\ -0.250000000000000 & 0 & 0 & 0 \\ -0.4122867597285291 & -0.1830127018922193 & 0 & 0 \end{pmatrix}. \end{aligned}$$

5.7.4 The linearly implicit peer method implPeer2

The second-order three-stage linearly implicit peer method implPeer2 was found by using explPeer as the underlying method in order to obtain its good stability properties when only the partial Jacobian is used. The remaining parameters have been optimized so that the method is stable for arbitrary large advection CFL numbers when the simplified Jacobian is used. The coefficients of implPeer2 are

$$\begin{aligned} c &= \begin{pmatrix} -0.0899531627878552 & 0.4676428830697650 & 1.000000000000000 \end{pmatrix}^T, \\ \gamma &= \begin{pmatrix} 0.5167851598100672 & 0.5167851598100672 & 0.5167851598100672 \end{pmatrix}^T, \\ B &= \begin{pmatrix} -0.0967059983845656 & 0.4915598645202344 & 0.6051461338643311 \\ -0.0470929826281593 & 0.2169946581702936 & 0.5720815963722115 \\ -0.0891437312845480 & 0.1573830315884013 & 0.1973233392586685 \end{pmatrix}, \\ S &= \begin{pmatrix} 0 & 0 & 0 \\ 0.2580167280856541 & 0 & 0 \\ 0.3269306113397434 & 0.4075067490977347 & 0 \end{pmatrix}, \end{aligned}$$

$$\begin{aligned}
A &= \begin{pmatrix} 0.0721007322008575 & -0.1322804288331288 & 0.1265069173192104 \\ 0.0478238719665258 & -0.4831372398722279 & -0.1163332106046261 \\ 0.0325906971440313 & 0.0702440095890842 & 0.1286761505892647 \end{pmatrix}, \\
R &= \begin{pmatrix} 0 & 0 & 0 \\ 1.1066883875756954 & 0 & 0 \\ -0.5020271673748957 & 1.0959786066300778 & 0 \end{pmatrix}, \\
G &= \begin{pmatrix} -0.2873500776711067 & 0.5010014103242811 & -0.7304364924632416 \\ -0.5198912797002499 & 1.5416287759361538 & -1.4010177204289034 \\ -0.3557339562292955 & 0.7879731539945604 & -0.9973424995819653 \end{pmatrix}, \\
H &= \begin{pmatrix} 0 & 0 & 0 \\ -0.1375049356170674 & 0 & 0 \\ 0.9103920894669218 & -0.8620739474602885 & 0 \end{pmatrix}.
\end{aligned}$$

5.7.5 The linearly implicit peer method implPeer3

For the construction of the third-order three-stage linearly implicit peer method implPeer3 all coefficients have been optimized in order to obtain a method which has the same stability properties as implPeer2 but furthermore is third-order due to its super-convergence property. The coefficients are

$$\begin{aligned}
c &= \left(-0.5777333525760953 \quad 0.3002651533615503 \quad 1.0000000000000000 \right)^T, \\
\gamma &= \left(0.4289052382134347 \quad 0.4289052382134347 \quad 0.4289052382134347 \right)^T, \\
B &= \begin{pmatrix} 0.0298493709790728 & 1.0110127004800935 & -0.0408620714591663 \\ 0.1860016560477698 & 0.3448649676740809 & -0.2085911976769894 \\ -0.0786622735220804 & 0.4003790662974256 & 0.2685200438402068 \end{pmatrix}, \\
S &= \begin{pmatrix} 0 & 0 & 0 \\ 0.6777245739551386 & 0 & 0 \\ 0.3680084510099432 & 0.0417547123745046 & 0 \end{pmatrix}, \\
A &= \begin{pmatrix} 0.1522177815016555 & -0.1749024319658557 & 0.1994864629551681 \\ -0.1144213489989192 & 0.2581057405561797 & 0.4184501342332333 \\ -0.0594148435598668 & 0.0083741953087149 & 0.2433007398619567 \end{pmatrix}, \\
R &= \begin{pmatrix} 0 & 0 & 0 \\ 0.6644497694527152 & 0 & 0 \\ 0.0008670049082676 & 1.1629972665721646 & 0 \end{pmatrix}, \\
G &= \begin{pmatrix} -0.2316113862533708 & 0.1681038871837252 & -0.3653977391437892 \\ 0.2077381806352361 & -0.2186525760535129 & -0.3384189571059057 \\ -0.0444597966294304 & 0.0700417649740132 & -0.2241133953266569 \end{pmatrix}, \\
H &= \begin{pmatrix} 0 & 0 & 0 \\ -0.0795718856892522 & 0 & 0 \\ 0.4337900898424071 & -0.6641639010737678 & 0 \end{pmatrix}.
\end{aligned}$$

6 Numerical tests

6.1 Order test 1: Burgers' equation

We now present some numerical tests to verify the order theory. The first order test is the inviscid Burgers' equation

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x}$$

with u the wind speed. As initial condition we choose

$$u(0, x) = 20 + 10 \sin^{50} \left(\frac{\pi x}{L} \right)$$

where $L = 30$ km is the length of the domain and we use periodic boundary conditions. The spatial step size is $\Delta x = 200$ m and the integration period is $t_e = 200$ s. The reference solution has been computed with implPeer3 with $\Delta t = 1/300$ s, i.e. with a CFL number of 1/2000.

Figure 11 shows the errors of the considered methods for different CFL numbers in the discrete L^2 norm. This test confirms the results from the order theory. All methods show the expected order, i.e. order 2 with the exception of the superconvergent third-order implPeer3. For the linearly implicit methods the partial Jacobian was used but the results are similar when we use the simplified Jacobian (not shown). Because there is no acoustics implPeer2 is equal to explPeer if the partial Jacobian is used. RK3 seems to be third-order for large time step sizes. This may be the case because RK3 fulfils one of the two order conditions for third-order Runge-Kutta methods and is third-order for linear problems. This tests shows that RK3 is the most accurate method for large CFL numbers while implPeer3 becomes the best method for medium and small CFL numbers due to its superconvergence property. The Rosenbrock method ROS3Pw is the most inaccurate method for all time step sizes.

6.2 Order test 2: Burgers' equation with acoustics

To test the order with an example with two different modes the pressure p is added to the inviscid Burgers' equation for the second order test:

$$\begin{aligned} \frac{\partial u}{\partial t} &= -u \frac{\partial u}{\partial x} - \frac{\partial p}{\partial x}, \\ \frac{\partial p}{\partial t} &= -u \frac{\partial p}{\partial x} - c_s^2 \frac{\partial u}{\partial x}. \end{aligned}$$

$c_s = 340$ m s⁻¹ is the speed of sound. These equations are the linearized version of the equations which were used in [44] and [45] to determine stability. The pressure is initialized with

$$p(0, x) = c_s,$$

all other parameters are equal to those of the first order test.

The errors in the discrete L^2 norm of the wind computed with the considered methods for different CFL numbers are illustrated by Figure 11. The split-explicit methods RK3

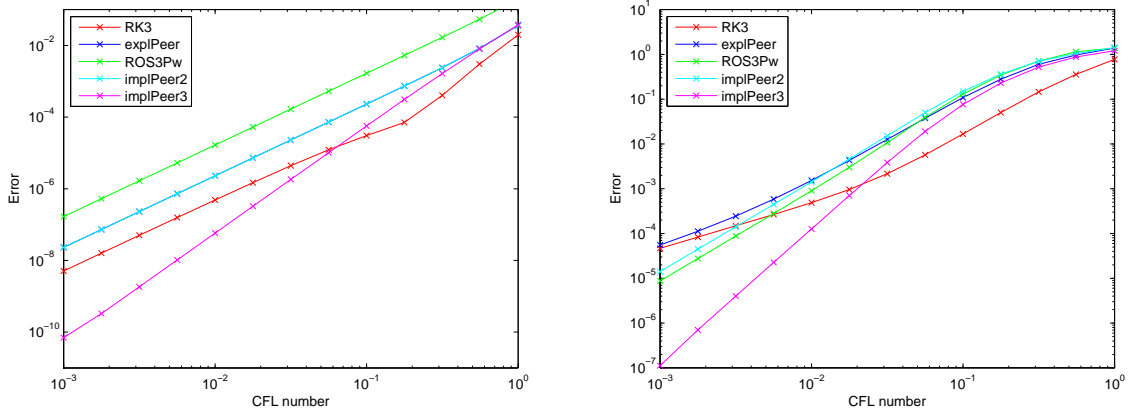


Figure 11: Error of the numerical solution for the Burgers' equation (left) and for the Burgers' equation with acoustics (right) in L^2 norm against CFL number.

and explPeer use the forward-backward Euler scheme with $n_s = 30$ small time steps per large time step. The results are quite similar to the first order test, RK3 is the most accurate method for large CFL numbers while implPeer3 is the best method for small time step sizes. But there is an important difference: For small CFL numbers the order of the split-explicit methods decreases to 1 because the forward-backward Euler scheme is first-order. This behaviour can be prevented by using larger n_s for small Δt . But because we wanted to show the methods with a n_s which is feasible in more realistic test cases we decided to present the order plot for a fixed number of n_s . We show the results for RK3 without divergence damping, i.e. it becomes unstable for large CFL numbers. The reason why this is not visible in the order plot is because the first signs of instability appear after around 900 s but for the order test $t_e = 200$ s.

6.3 Order test 3: Rising bubble

For the third order test we compute with the compressible Euler equations. We consider the rising bubble test case described in Section 6.4 with the only exception of the doubled spatial step size of $\Delta x = 250$ m. The maximum wind speed which occurs is 27 m s^{-1} . The reference solution has been computed with implPeer3 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 Figure 12 shows $\sqrt{e_{\rho u}^2 + e_{\rho w}^2 + e_{\rho \theta}^2}$ in dependency of the CFL number and the required CPU time. The results are very similar to the much simpler order test 2, implPeer3 is the most accurate method and the split-explicit methods are first-order for small CFL numbers because they use a first-order scheme for the integration of the fast part. Because of the longer integration period of 1000 s instabilities occur when RK3 is used without divergence damping. Therefore the error is not plotted for RK3 with large CFL numbers. We now can see that implPeer3 is the best method for most CFL numbers. For large time step sizes RK3 is unstable while for small Δt implPeer3 is more accurate because it is

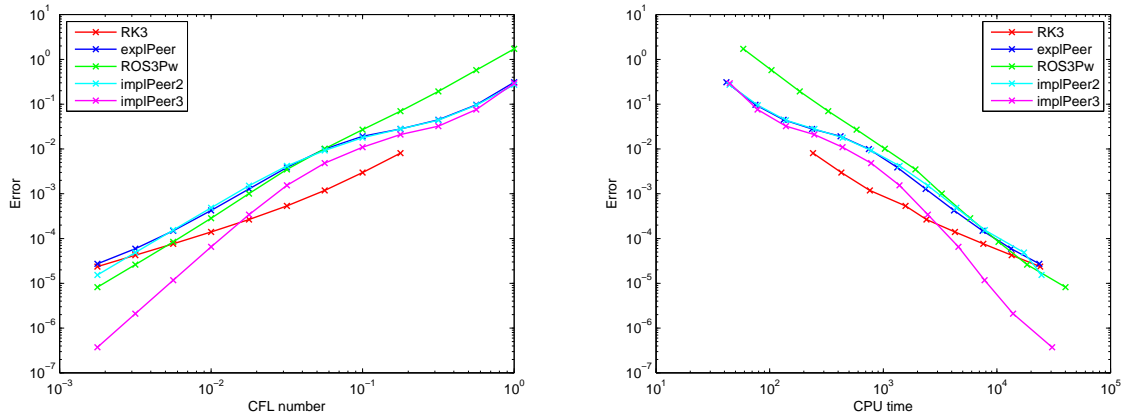


Figure 12: Error of the numerical solution for the rising bubble test in L^2 norm against CFL number (left) and CPU time (right).

superconvergent with order 3. Only for medium time step sizes RK3 is the best method from an accuracy point of view. Figure 12 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 $n_s = 30$ for split-explicit methods. This may be different when using efficient iterative solvers and the linearly implicit methods might get even better.

6.4 Test case 1: Rising bubble

We present the results for five test cases. The first one is the rising bubble which is described in detail in [2]. 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

$$C = \sqrt{\left(\frac{x - x_c}{x_r}\right)^2 + \left(\frac{z - z_c}{z_r}\right)^2}, \quad (35)$$

$x_c = 0$ 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 m s⁻¹ from the left as in [44] 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 as shown by Figure 13.

Figure 14 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 m s⁻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. The split-explicit methods use the forward-backward Euler scheme with $n_s = 30$ small time steps per large time step. Furthermore RK3 has to use divergence damping with $\nu = 0.025$ in order to be stable. The solutions computed with the split-explicit methods and the linearly implicit methods with partial Jacobian are quite good despite the fact that the time step sizes are close to the CFL condition from linear stability theory. ROS3Pw produces a slightly more symmetric solution than the two linearly implicit peer methods, but all solutions are rather similar although the other methods use the double time step size of ROS3Pw. The solutions appear to be much more accurate than the results published in [44] where a second-order split-explicit Runge-Kutta method with a five times smaller CFL number, namely 0.32, was used. When computing with the simplified Jacobian the results are worse than the solutions computed with the partial Jacobian. The solutions show asymmetries but implPeer2 is more accurate than ROS3Pw and implPeer3, as is to be expected when considering the amplitude and phase properties shown in Section 4.5.

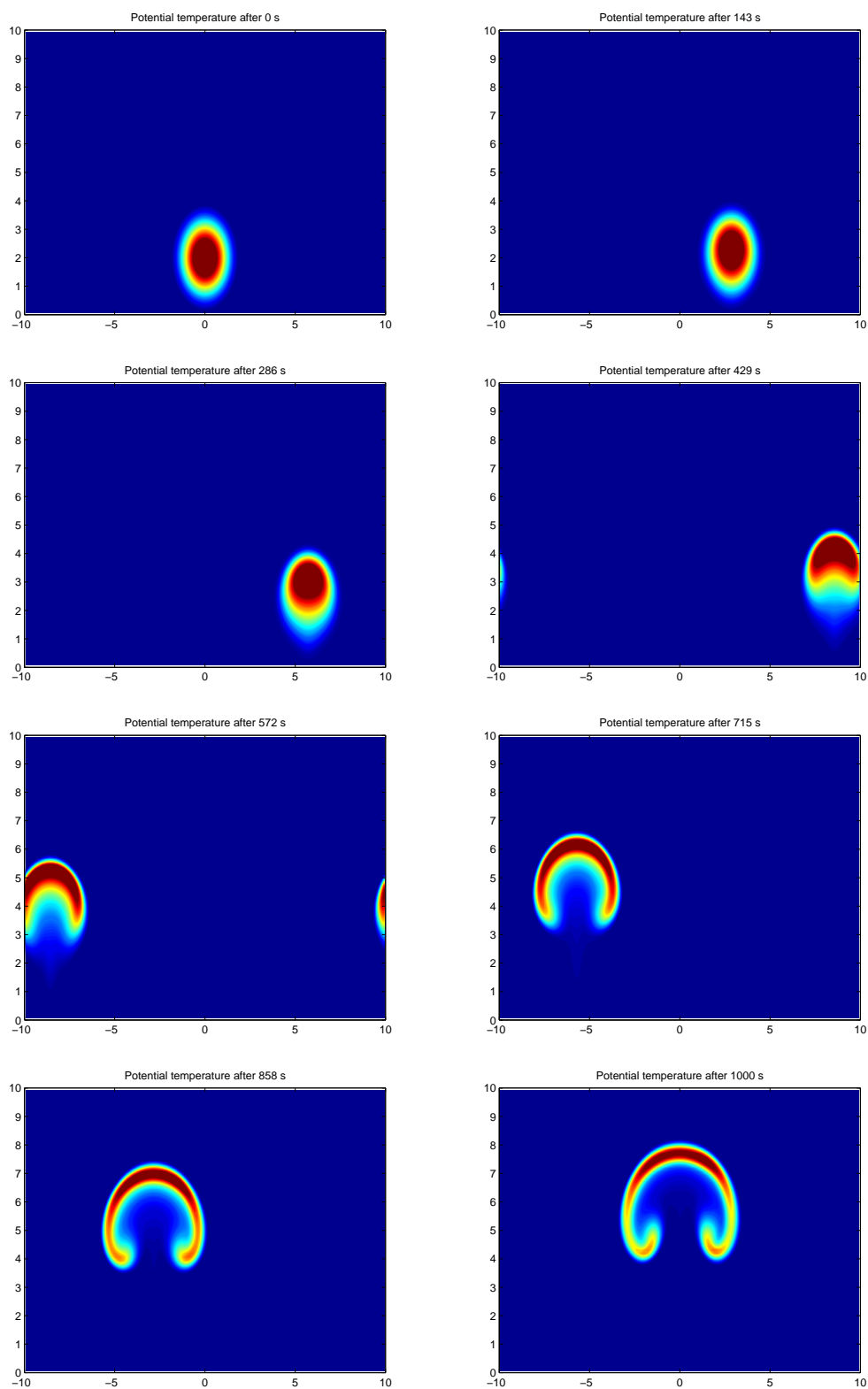


Figure 13: Potential temperature after 0 s, 143 s, 286 s, 429 s, 572 s, 715 s, 858 s and 1000 s for the rising bubble test case.

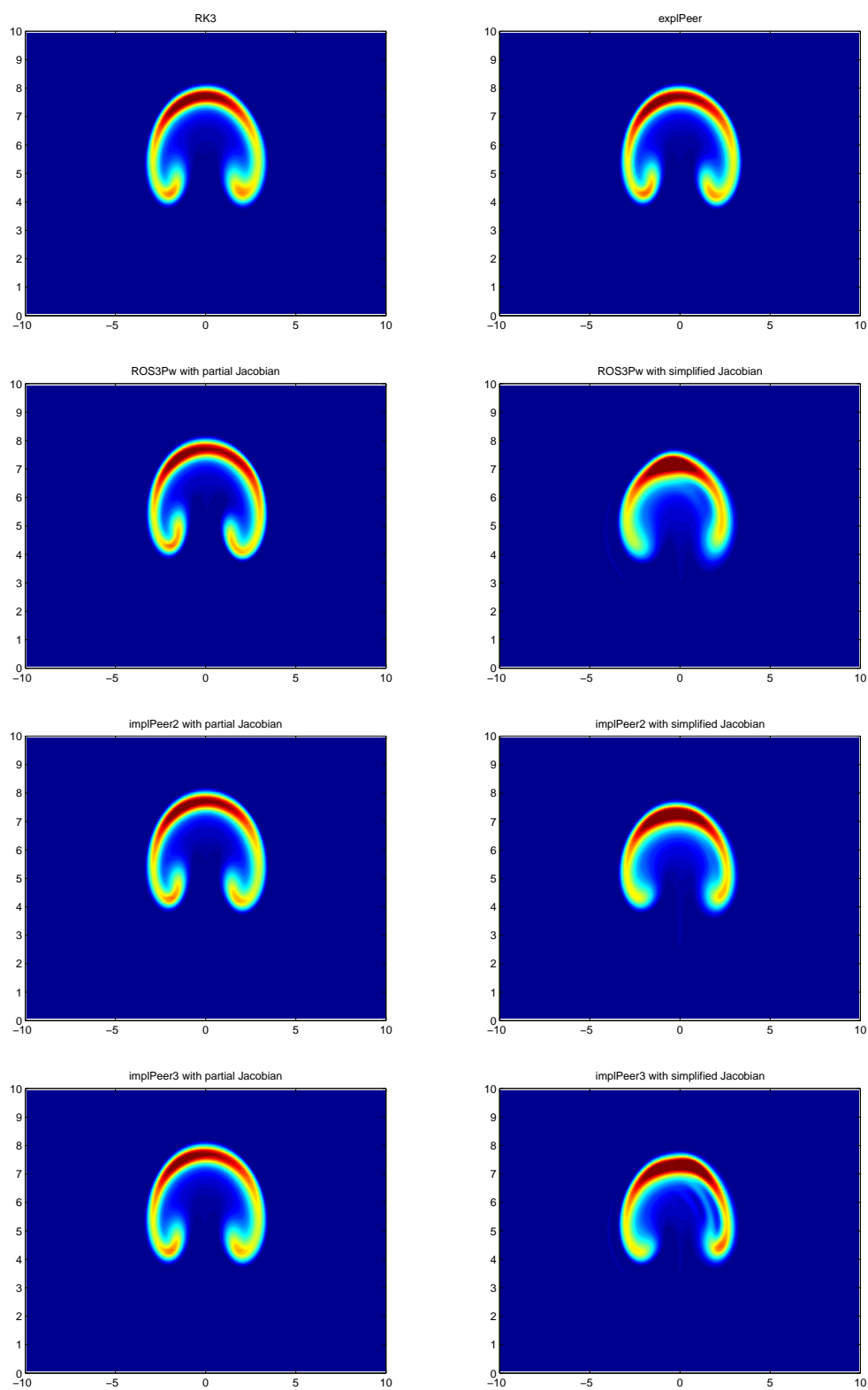


Figure 14: Potential temperature after 1000 s for the rising bubble test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with partial (left) and simplified (right) Jacobian.

6.5 Test case 2: Density current

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 (35) but with $z_c = 3$ km and $x_r = 4$ km. In this test there is diffusion with $\nu = 75 \text{ m}^2 \text{ s}^{-1}$ as described in [35]. Because the bubble is colder it sinks and after crashing to the bottom (with free slip vertical boundary conditions) several eddies form as illustrated by Figure 15. The domain has a width of 36 km and the uniform horizontal flow of 20 m s^{-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 $\Delta t = 2.5$ s, with the exception of ROS3Pw, which used $\Delta t = 1.25$ s because of its worse stability properties. The split-explicit methods use FB with $n_s = 15$ while the linearly implicit methods use the partial Jacobian. Furthermore we compute with implPeer2 and $\Delta t = 3.5$ 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 m s^{-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 rising bubble test the maximum wind speed is 61 m s^{-1} . The time step sizes correspond to advection CFL numbers of 2.1 for $\Delta t = 3.5$ s, 1.5 for $\Delta t = 2.5$ s and 0.8 for $\Delta t = 1.25$ s. In the test with the dynamically adapted Jacobian the threshold of 40 m s^{-1} corresponds to an advection CFL number of 1.4. So the CFL numbers are close to the maxima from linear stability theory just like in the rising bubble test. Figure 16 shows the results: Every solution shows three eddies at each side where the backward directed (right) eddies are better pronounced. This phenomenon was also documented in [45] and [13] (there the horizontal mean wind came from the right). While the backward directed eddies look rather similar for the different methods the forward directed eddies are better pronounced when computing with smaller time step sizes but this is only barely visible. Despite the fact that the used CFL numbers are close to the maxima from stability theory, all solutions are in agreement with the solution presented in [35] (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 partial and simplified Jacobian harmonize very well.

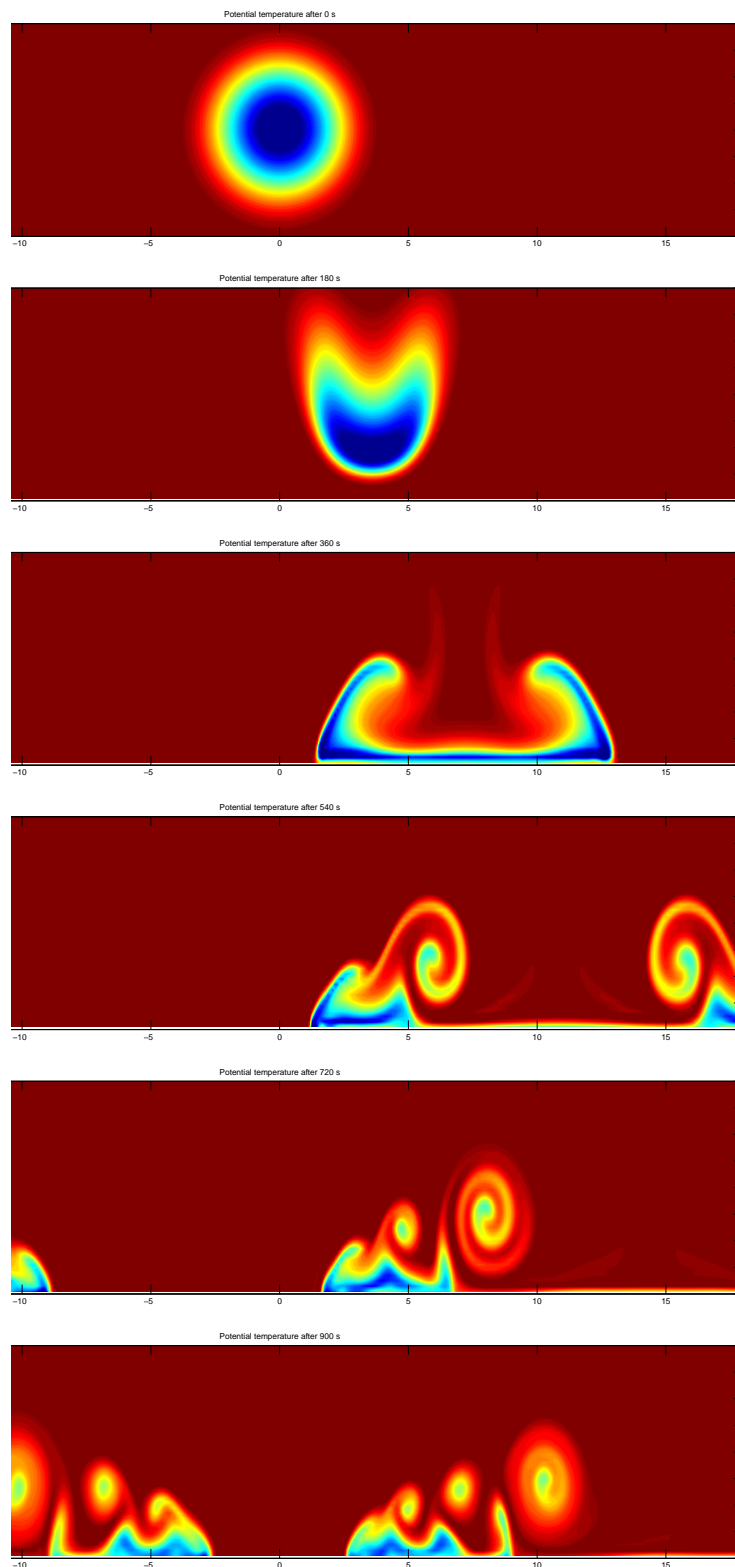


Figure 15: Potential temperature after 0 s, 180 s, 360 s, 540 s, 720 s and 900 s for the density current test case.

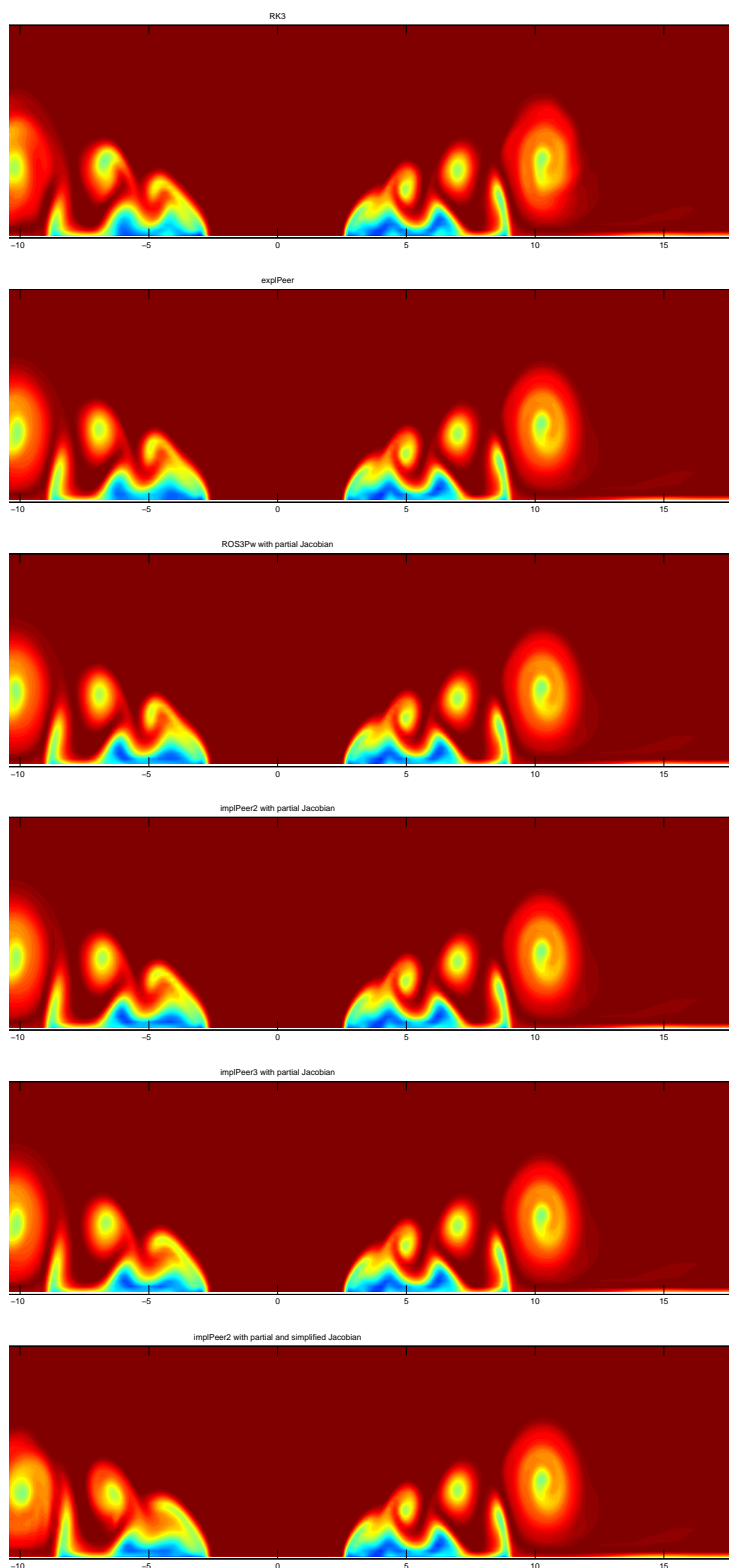


Figure 16: Potential temperature after 900 s for the density current test computed with (from top to bottom) RK3 and explPeer with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with partial Jacobian and implPeer2 with simplified Jacobian in cells with high wind speeds only.

6.6 Test case 3: Flow over a mountain

The third test problem is the flow over a mountain. This problem is described in [6]. There is a mountain with a half-width of 1 km and a height of 400 m as shown in Figure 17. The spatial resolution is 200 m in both directions. A horizontal flow of 10 m s^{-1} from the left results in gravity waves in the lee of the mountain as can be seen in Figure 17.

Figure 18 shows the vertical wind after 2160 s. Because of the representation of the orography with cut cells very small cells appear, the smallest having a size of 0.17% of the cells in the free regions. Because the initial values, which are needed for the peer methods, are computed with the explicit method RK3 the time step size needed for the initialization would be 600 times smaller than the time step size used by the peer methods. Therefore we close all cells which are smaller than 1% of 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 of the peer methods which is 100 times smaller than Δt . The time step size used for ROS3Pw with simplified Jacobian and the linearly implicit peer methods is $\Delta t = 20 \text{ s}$, while ROS3Pw with simplified Jacobian in cut cells only computes with $\Delta t = 10 \text{ s}$ and the split-explicit methods use $\Delta t = 0.2 \text{ s}$ and FB with $n_s = 50$. The maximum wind speed that occurs is 14 m s^{-1} which results in an advection CFL number of 1.4 in the free atmosphere, respectively 0.7 for ROS3Pw with simplified Jacobian in cut cells only. In the cut cells the advection CFL number rises to 140 and the CFL number for the acoustics is up to 4800. Despite those large CFL numbers all solutions are in very good agreement with the results published in [6] and no clear difference is visible between the solutions computed with the full but simplified Jacobian everywhere and in cut cells only. These results verify the linear stability theory and confirm that the simplified Jacobian is needed in cut cells only.

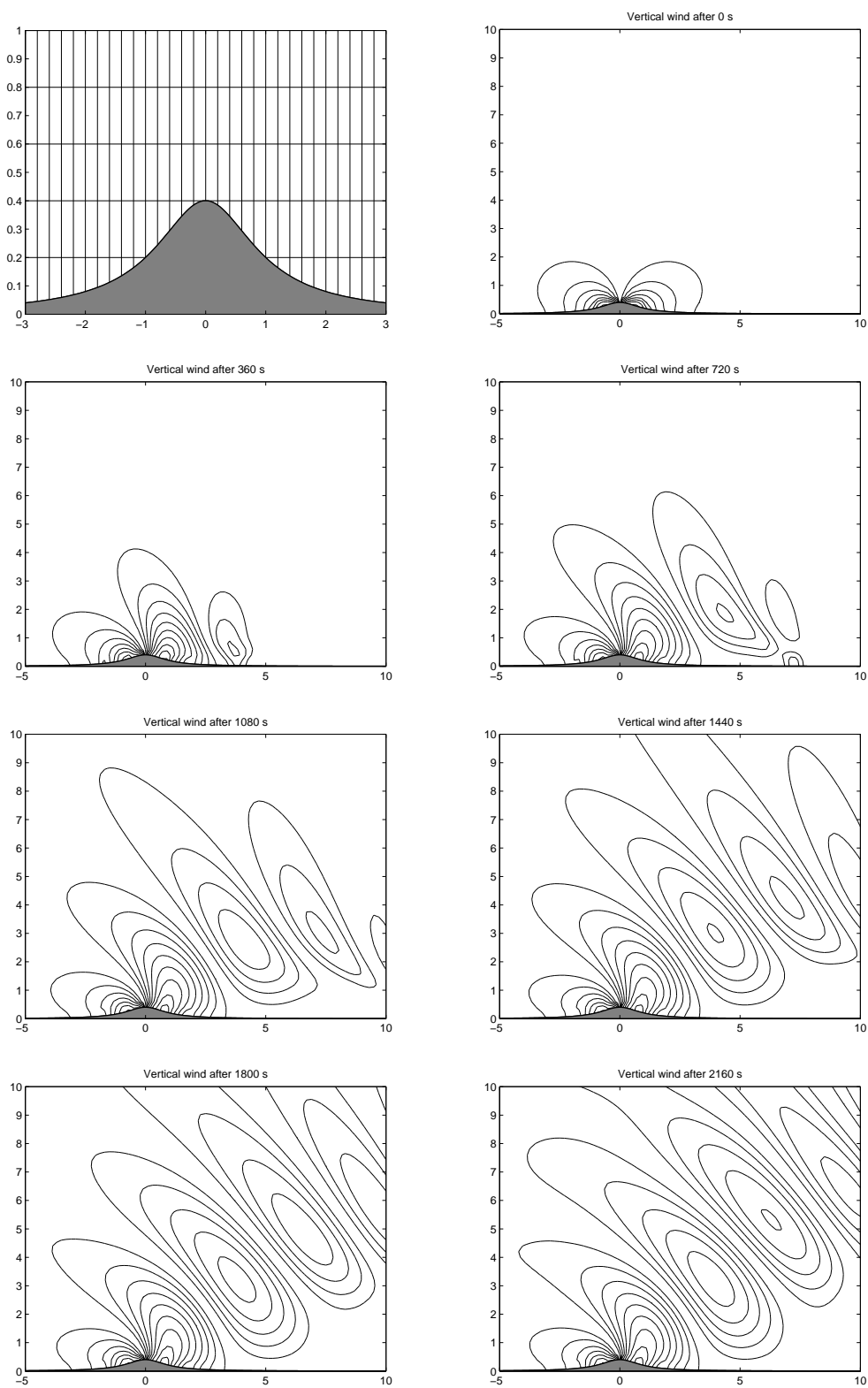


Figure 17: Witch of Agnesi mountain with cut cells and vertical wind after 0 s, 360 s, 720 s, 1080 s, 1440 s, 1800 s and 2160 s for the flow over a mountain test case. Contour interval is 0.25 m s^{-1} .

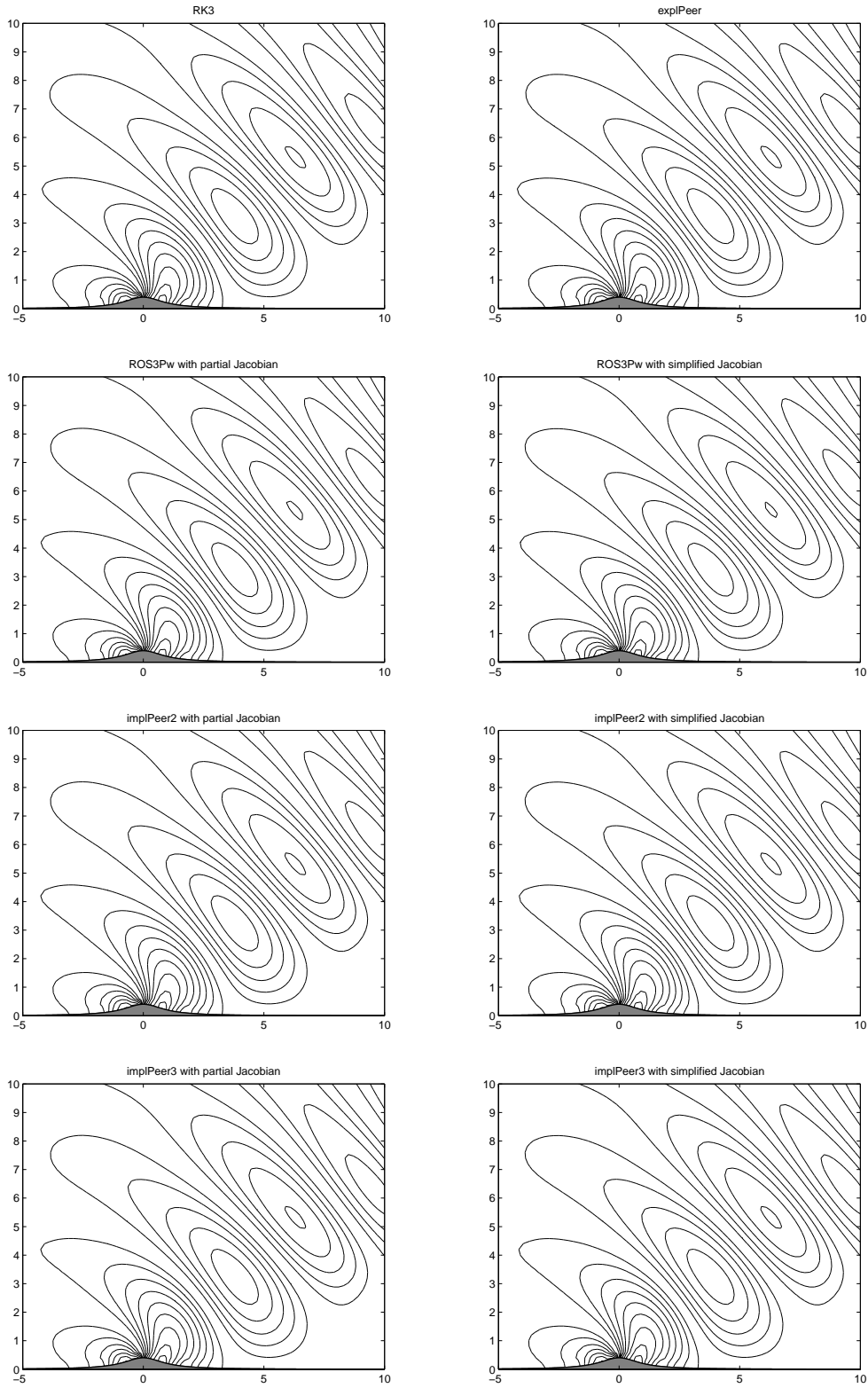


Figure 18: Vertical wind after 2160 s for the flow over a mountain test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with simplified Jacobian in cut cells only (left) and everywhere (right). Contour interval is 0.25 m s^{-1} .

6.7 Test case 4: Zeppelin test

The fourth test is a more fanciful 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 rising bubble 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 simplified Jacobian while other parts are updated with the partial Jacobian. A detailed description of the initial conditions can be found in [17] but our setup differs a little bit from that one. There is the same warm bubble as in the rising bubble test with the same grid but the bubble is located 1 km to 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 Figure 19 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 flow over a mountain test we close tiny cells so that the smallest cells have a minimum size of 1% of the cells in the free atmosphere. The bubble will crash against the obstacle and will asymmetrically divide into two separate parts as demonstrated by Figure 19.

ROS3Pw with simplified Jacobian and the linearly implicit peer methods compute with $\Delta t = 10$ s and the maximum wind speed is approximately 17.5 m s^{-1} , which 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 before we use the half time step size for ROS3Pw with simplified Jacobian in cut cells only and a 100 times smaller time step size for the split-explicit methods. They use FB with $n_s = 40$ for the integration of the fast part. As in the density current and flow over a mountain tests there is no qualitative difference between the different methods as shown by Figure 20. So again it is sufficient to use the full but simplified Jacobian in cut cells only.

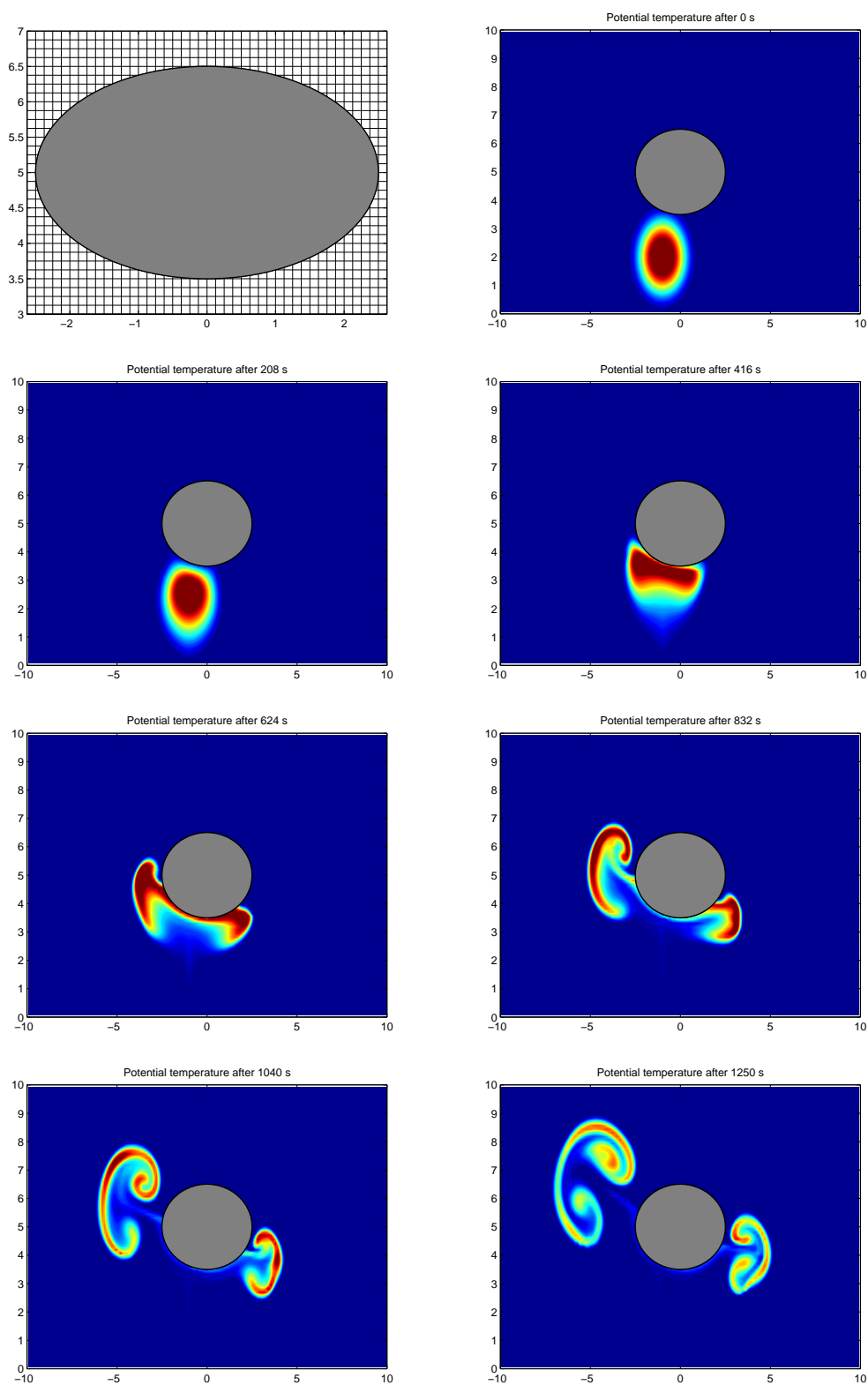


Figure 19: Zeppelin obstacle with cut cells and potential temperature after 0 s, 208 s, 416 s, 624 s, 832 s, 1040 s and 1250 s for the zeppelin test case.

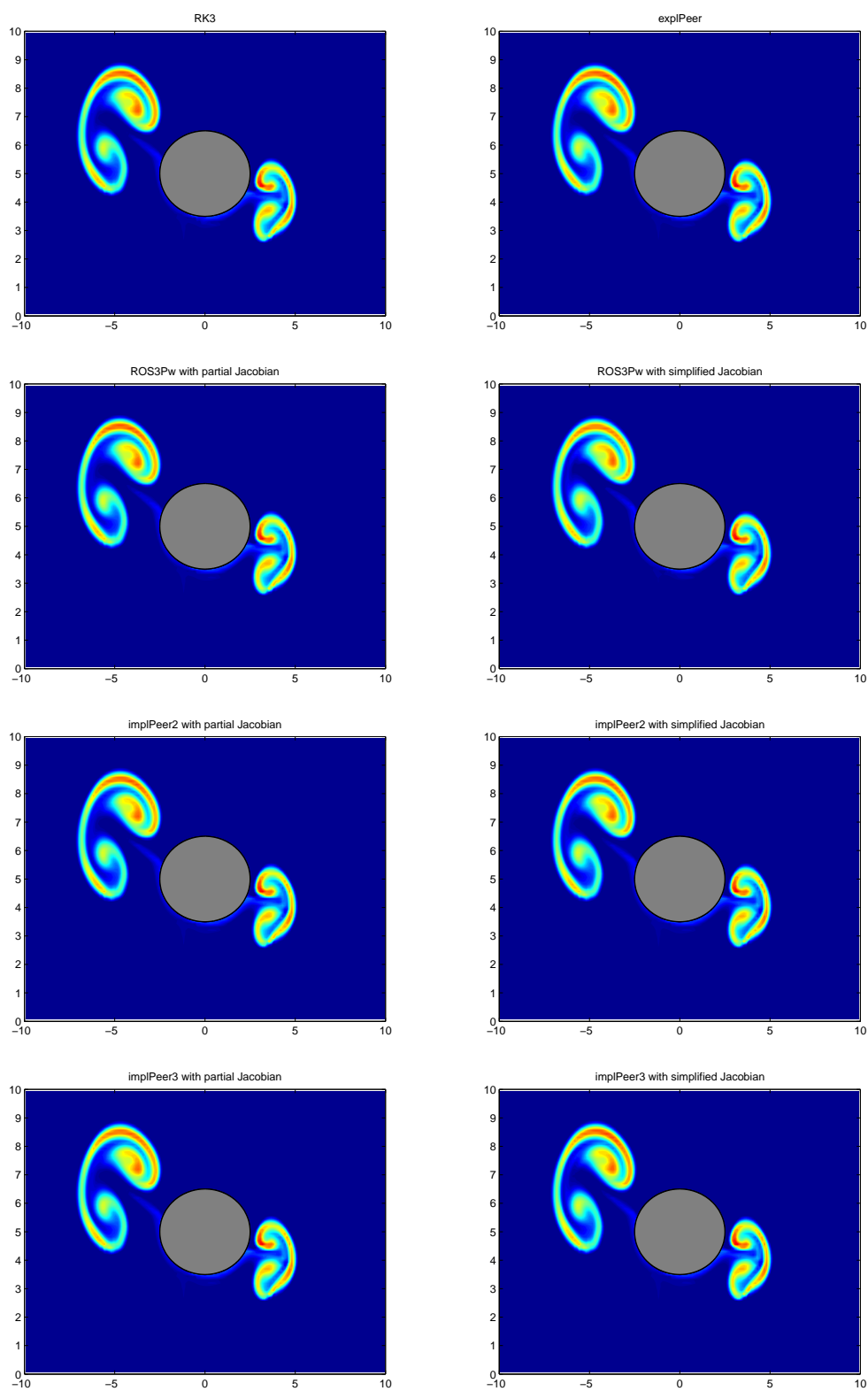


Figure 20: Potential temperature after 1250 s for the zeppelin test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with simplified Jacobian in cut cells only (left) and everywhere (right).

6.8 Test case 5: Dam-break test

For the fifth test we solve the inviscid shallow water equations. These can be derived from the Euler equations by setting $\theta = 1$ as described in Section 5.1. We present the results for the two-dimensional dam-break described in [37]. The domain is 1400 m long in both directions with a dam in the center of the domain. It has broken from 560 m to 840 m, i.e. the water from the side where it has a depth of 10 m flows into the other half of the domain where the water is 9.5 m deep. Contrariwise to [37] our dam has a width of the spatial step size $\Delta x = \Delta z = 7$ m, i.e. there are uncut cells and cells which are completely cut out but no cut cells. Therefore the explicit methods can compute with the same time step sizes which the linearly implicit schemes use. To make the test more stringent for linearly implicit methods we broaden the dam-break by 7 cm so that cut cells appear at the edges of the break which have 1% of the size of the uncut cells. This tiny difference in the setup has no visible influence on the solutions computed by the linearly implicit methods (not shown). Therefore we compare the results of both kinds of methods despite the fact that they use a slightly different setup from this test case. Figure 21 shows the water depth during the first 50 s of the test computed by RK3 with $\Delta t = 0.1$ s.

The results after 50 s are presented in Figure 22. The maximum occurring speed was 2 m s^{-1} . All methods computed with $\Delta t = 1$ s, i.e. with a CFL number of 0.3 in the uncut cells. Because of the slightly different setup for the linearly implicit methods their time step size corresponds to a CFL number of 30 in the cut cells at the edges of the dam-break. The split-explicit methods use the forward-backward Euler scheme with $n_s = 3$ small time steps per large time step. When comparing the results we find that RK3 produces the most accurate solution. It is in good agreement with the solutions presented in [37] but there a 500 times smaller time step size was used. The discontinuity in the initial conditions results in a solution which is not very smooth. This behaviour was also documented in [37] for the inviscid case. In contrast to the RK3 solution the solutions produced by the other methods are much more damped, especially the solutions computed with the linearly implicit methods are not as accurate as the solution of RK3. This fact results from the accuracy of the considered methods, all methods produce as good solutions as RK3 if the small time step size from [37] is used (not shown). No difference is visible between the solutions for which the simplified Jacobian was used everywhere and the solutions of the methods which used the simplified Jacobian only in the two cut cells at the edges of the dam-break. The linearly implicit methods can stably compute with the simplified Jacobian in cut cells only, as was the case for the other tests which incorporate cut cells, too.

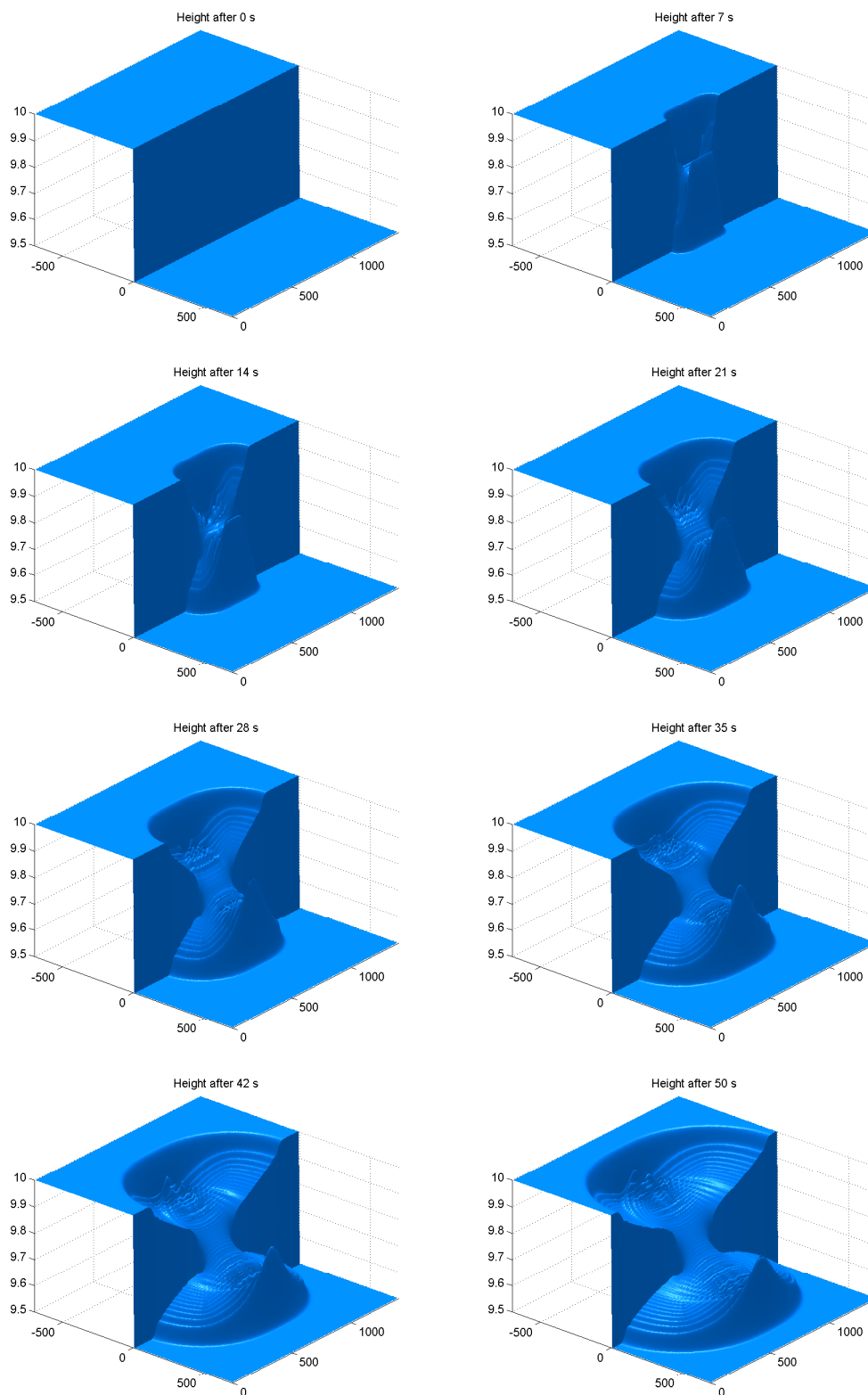


Figure 21: Depth after 0 s, 7 s, 14 s, 21 s, 28s, 35 s, 42 s and 50 s for the dam-break test case.

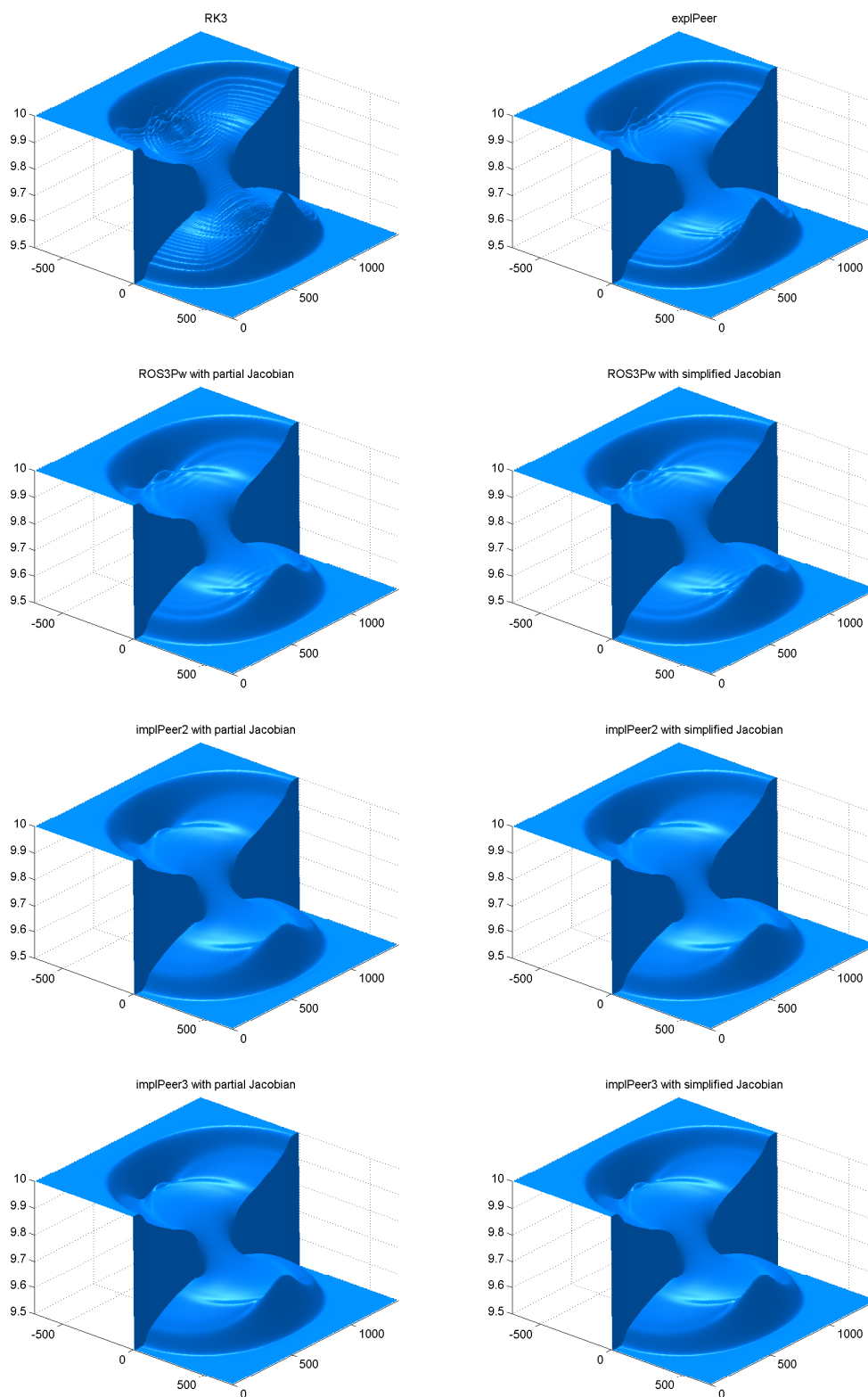


Figure 22: Depth after 50 s for the dam-break test computed with (from top to bottom) RK3 (left) and explPeer (right) with forward-backward Euler scheme, ROS3Pw, implPeer2 and implPeer3 with simplified Jacobian in cut cells only (left) and everywhere (right).

7 Conclusions

In this thesis we presented a new methodology to describe time-splitting methods for the solution of the compressible Euler equations. The basic principle of the presented approach is that we split the Euler equations into the advection part and the acoustics and make the assumption that one part, the acoustics, of the split differential equation can be solved analytically so that stability and order investigations can be made for the underlying method which solves the advection part. With this methodology we considered split-explicit Runge-Kutta methods and reproduced the known stability results.

We introduced the class of explicit peer methods and used them as the underlying method for the solution of split differential equations. We presented order conditions and stability results and finally derived the split-explicit peer method `explPeer` which is as accurate and efficient as the common split-explicit Runge-Kutta method RK3. Particularly linear stability analysis showed that `explPeer`, even without any artificial damping, is as stable as RK3 with divergence damping. The derived split-explicit peer method `explPeer` also remains stable when using the trapezoidal rule as integrator for the acoustics in contrast to RK3, which is not sufficiently stable and therefore needs off-centering which results in a reduction of order down to 1.

The computational effort of `explPeer` is comparable to RK3: Both have $s = 3$ stages and therefore need three evaluations of the expensive, slow part of the right-hand side per large time step. Because the sum of the fast integration intervals of `explPeer` $\|\alpha\|_1 = 1.44$ is about 20% smaller than the sum of the nodes of RK3 ($1/3+1/2+1=11/6$) the expense for the evaluation of the fast part is a slightly lower for the peer method. In general peer methods have a higher overhead than Runge-Kutta methods because they use more linear combinations of the numerical solution and its function evaluations but for partial differential equations this fact is a negligible disadvantage because of the very expensive right-hand side.

One disadvantage of the peer method might be the memory requirement. Because only the secondary diagonal of the Butcher tableau of RK3 has non-zero entries RK3 needs the memory capacity for three numerical solutions and function evaluations: One for the initial value, another for the function evaluation and the third for the updated fast part. In contrast the peer method needs twice of this memory: $s = 3$ for the linear combinations $e_i^T BY_{m-1} + e_i^T SY_m$ which are successively updated to Y_{mi} , i.e. in the i th stage the integration of the fast part starts with the initial value $e_i^T BY_{m-1} + e_i^T SY_m$ and every update is stored at the same place so that in the end $e_i^T BY_{m-1} + e_i^T SY_m$ is overwritten by Y_{mi} . The same holds for the function evaluations so that the memory for six solutions is needed altogether. Because in practical applications problems are solved in a massively parallel environment which usually uses domain decomposition techniques every CPU has to know only the values on its small part of the whole domain and therefore the requirement for twice of the memory capacity is no relevant disadvantage, there are enough reserves.

The split-explicit peer method `explPeer` with the trapezoidal rule can be interpreted as a linearly implicit peer method whose Jacobian incorporates acoustics only. Unfortunately this linearly implicit peer method is not much more stable when it uses the simplified Jacobian which additionally includes advection and diffusion. This fact moti-

vated us to consider linearly implicit peer methods which have the same good stability properties of explPeer when they use the partial Jacobian which only incorporates acoustics, but furthermore they should be stable for arbitrary large CFL numbers if they use the full but simplified Jacobian. This property is necessary because we use cut cells for the representation of orography which can result in arbitrary small cells.

We derived order conditions for the class of linearly implicit peer methods which allow the construction of methods that retain the full order independently of what is used as Jacobian. Furthermore we presented a condition which is sufficient for superconvergence. With this order theory we constructed two linearly implicit methods with three stages and order of consistency 2. They are A-stable in the common sense (not shown). The method implPeer2 was derived from the coefficients of explPeer, while implPeer3 was constructed from new coefficients in order to obtain a superconvergent third-order method. For wind speeds smaller than $c_s/6$ they are stable for arbitrary large advection and acoustic CFL numbers if the simplified Jacobian is used. They are as stable as explPeer, 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 these methods with a genetic algorithm, where we optimized the degrees of freedom with respect to small amplitude and phase errors with the desired order conditions and stability criteria as side conditions.

The simplified Jacobian corresponds to the advection form of the Euler equation while the conservative form was used as the right-hand side to conserve mass, momentum and entropy. It uses a lower-order spatial discretization. Furthermore it is used in cut cells only while in the remaining domain the partial Jacobian, which only incorporates the acoustic part of the Euler equations, is used. Table 1 on page 55 shows the amount of memory which is needed for the Jacobian per grid cell. The partial Jacobian needs less memory than the simplified Jacobian and much less than the exact Jacobian. Because in numerical weather prediction models orography appears at ground level, 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 memory point of view. Therefore the amount of memory for the Jacobian in applications where the full but simplified Jacobian is used in cut cells only is nearly the same as for the partial Jacobian. 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 more entries have to be computed and the systems might be worse conditioned if the simplified Jacobian is used. In comparison to the exact Jacobian the speed-up would even be 5.7. The computing time for the zeppelin test with the simplified Jacobian for implPeer2 (without the time needed for computing the initial values) is 608 s with MATLAB on an Intel Core 2 Quad Q9550 @ 2833 Mhz with 3 GB RAM. When using the full but simplified Jacobian in cut cells only the computation takes 405 s, i.e. 33% of the computing time is saved, which corresponds to a speed-up of 1.5. The times needed for setting up and solving 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 explPeer 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 implPeer2 with

the same time step size are 13251 s (3975 s) when the simplified Jacobian is used in cut cells only and 20011 s (10560 s) when it is used everywhere. So the speed-up when using the simplified 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 into what computational time can be saved when computing with the full but simplified Jacobian in cut cells only. Furthermore the partial and simplified Jacobians are easier to implement than the exact Jacobian due to the smaller stencils, which 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 application of the split-explicit methods to the compressible Euler equations in conservative flux-form with a finite volume spatial discretization confirmed the linear stability results: Both methods stably integrate the test problems. So we found a split-explicit method for the compressible equations which is as accurate, stable and efficient as RK3 but without the need for damping. The split-explicit peer method is stable even for a very high number of small time steps and therefore it is appropriate for solving low Mach number problems. The use of a time step size which nearly corresponds to the Courant number from linear stability theory still produces an acceptable solution.

The tests which incorporate orography, i.e. the flow around a mountain test, the zeppelin test and the dam-break test, all reveal the same results: No differences between the solutions computed with the simplified Jacobian in cut cells only and the simplified Jacobian everywhere are visible. Despite 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 the stability maximum of 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 methods with the partial and the full but simplified Jacobian harmonize 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 simplified 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 and diffusion only where it is necessary for stability. There are two reasonable applications 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 m s^{-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 which

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 cannot be violated, i.e. this ansatz not only results in computational efficiency but also in reliability of the model.

References

- [1] K. Biermann, *Explizite Zweischrittmethoden mit Peer-Variablen*, diploma thesis (2005), Institute of Mathematics, University of Halle, 55 pp.
- [2] G.H. Bryan and J.M. Fritsch, *A Benchmark Simulation for Moist Nonhydrostatic Numerical Models*, Mon. Wea. Rev. 130 (2002), 2917-2928.
- [3] K. Burrage and J.C. Butcher, *Non-linear stability of a general class of differential equation methods*, BIT 20 (1980), 185-203.
- [4] Deutscher Wetterdienst, Offenbach (DWD), www.dwd.de/forschung.
- [5] G. Fischer, *Comments on 'Some Problems Involved in the Numerical Solutions of Tidal Hydraulics Equations'*, Mon. Wea. Rev. 93 (1965), 110-111.
- [6] W. Gallus and J.B. Klemp, *Behaviour of flow over step orography*, Mon. Wea. Rev. 128 (2000), 1153-1164.
- [7] C. Gatti-Bono and P. Colella, *An anelastic allspeed projection method for gravitationally stratified flows*, J. Comput. Phys. 216 (2006), 589-615.
- [8] A. Gerisch, J. Lang, H. Podhaisky and R. Weiner, *High-order linearly implicit two-step peer-finite element methods for time-dependent PDEs*, Appl. Numer. Math. 59 (2009), 624-638.
- [9] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I*, 2nd ed., Springer-Verlag (1993), 528 pp.
- [10] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, 2nd ed., Springer-Verlag (1996), 614 pp.
- [11] D. Hinneburg and O. Knoth, *Non-dissipative cloud transport in Eulerian grid models by the volume-of-fluid (VOF) method*, Atmos. Environ. 39 (2005), 4321-4330.
- [12] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*, John Wiley & Sons (2009), 482 pp.
- [13] S. Jebens, O. Knoth and R. Weiner, *Explicit Two-Step Peer Methods for the Compressible Euler Equations*, Mon. Wea. Rev. 137 (2009), 2380-2392.
- [14] S. Jebens, O. Knoth and R. Weiner, *Linearly Implicit Peer Methods for the Compressible Euler Equations*, to appear in Appl. Numer. Math.
- [15] S. Jebens, O. Knoth and R. Weiner, *Partially Implicit Peer Methods for the Compressible Euler Equations*, J. Comput. Phys. 230 (2011), 4955-4974.
- [16] S.K. Kar, *A Semi-Implicit Runge-Kutta Time-Difference Scheme for the Two-Dimensional Shallow-Water Equations*, Mon. Wea. Rev. 134 (2006), 2916-2926.

- [17] R. Klein, K.R. Bates and N. Nikiforakis, *Well-balanced compressible cut-cell simulation of atmospheric flow*, Phil. Trans. R. Soc. 367 (2009), 4559-4575.
- [18] J.B. Klemp and R.B. Wilhelmson, *The Simulation of Three-Dimensional Convective Storm Dynamics*, J. Atmos. Sci. 35 (1978), 1070-1096.
- [19] O. Knöth and R. Wolke, *Implicit-explicit Runge-Kutta methods for computing atmospheric reactive flows*, Appl. Numer. Math. 28 (1998), 327-341.
- [20] J. Lang and J.G. Verwer, *ROS3P - An Accurate Third-Order Rosenbrock Solver Designed for Parabolic Problems*, BIT 41 (2001), 730-737.
- [21] F.M. Mesinger, *Forward-backward scheme, and its use in a limited area model*, Contrib. Atmos. Phys. 50 (1977), 200-210.
- [22] R.A. Pielke and coauthors, *A Comprehensive Meteorological Modeling System - RAMS*, Meteor. Atmos. Phys. 49 (1992), 69-91.
- [23] H. Podhaisky, R. Weiner and B.A. Schmitt, *Rosenbrock-type 'Peer' two-step methods*, Appl. Numer. Math. 53 (2005), 409-420.
- [24] R.J. Purser, *Accuracy Considerations of Time-Splitting Methods for Models Using Two-Time-Level Schemes*, Mon. Wea. Rev. 135 (2007), 1158-1164.
- [25] J. Rang and L. Angermann, *New Rosenbrock W-methods of Order 3 for PDAEs of Index 1*, BIT 45 (2005), 761-787.
- [26] B.A. Schmitt, R. Weiner and S. Jebens, *Parameter optimization for explicit parallel peer two-step methods*, Appl. Numer. Math. 59 (2009), 769-782.
- [27] A.F. Shchepetkin and J.C. McWilliams, *The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model*, Ocean Modelling 9 (2005), 347-404.
- [28] W.C. Skamarock, J. Dudhia, D. Gill, D. Barker, W. Wei and J. Powers, *A description of the advanced research WRF version 2*, NCAR Tech. Note (2008), NCAR/TN-468+STR, 88 pp.
- [29] W.C. Skamarock and J.B. Klemp, *The Stability of Time-Split Numerical Methods for the Hydrostatic and the Nonhydrostatic Elastic Equations*, Mon. Wea. Rev. 120 (1992), 2109-2127.
- [30] W.C. Skamarock and J.B. Klemp, *Efficiency and Accuracy of the Klemp-Wilhelmson Time-Splitting Technique*, Mon. Wea. Rev. 122 (1994), 2623-2630.
- [31] W.C. Skamarock and J.B. Klemp, *A time-split nonhydrostatic atmospheric model for weather research and forecasting applications*, J. Comput. Phys. 227 (2008), 3465-3485.
- [32] R. Skeel, *Analysis of Fixed-Stepsize Methods*, SIAM J. Numer. Anal. 13 (1976), 664-685.

- [33] G. Steinebach, *Order-reduction of ROW-methods for DAEs and Method of Lines Applications*, Preprint 1741 (1995), Technische Universität Darmstadt, Darmstadt, 21 pp.
- [34] J. Steppeler, H.W. Bitzer, M. Minotte and L. Bonaventura, *Nonhydrostatic Atmospheric Modeling using a z-Coordinate Representation*, Mon. Wea. Rev. 130 (2002), 2143-2149.
- [35] J. Straka, R.B. Wilhelmson, L.J. Wicker, J.R. Anderson and K. Droegemeier, *Numerical Solutions of a Nonlinear Density Current: A Benchmark Solution and Comparisons*, Int. J. Numer. Methods Fluids 17 (1993), 1-22.
- [36] H. Sundqvist, *On Vertical Interpolation and Truncation in Connexion with Use of Sigma System Models*, Atmosphere 14 (1976), 37-52.
- [37] E. Tadmor and W. Zhong, *Energy-Preserving and Stable Approximations for the Two-Dimensional Shallow Water Equations*, Mathematics and Computation, a Contemporary View, Springer-Verlag (2008), 67-94.
- [38] R.L. Walko and 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.
- [39] R. Weiner, K. Biermann, B.A. Schmitt and H. Podhaisky, *Explicit two-step peer methods*, Comput. Math. Appl. 55 (2008), 609-619.
- [40] R. Weiner, S. Jebens, B.A. Schmitt and H. Podhaisky, *Explicit parallel two-step peer methods*, Reports of the Institute of Numerical Mathematics 10 (2006), Institute of Mathematics, University of Halle, 18 pp.
- [41] R. Weiner, B.A. Schmitt and H. Podhaisky, *Parallel 'peer' two-step W-methods and their application to MOL-systems*, Appl. Numer. Math. 48 (2004), 425-439.
- [42] R. Weiner, B.A. Schmitt, H. Podhaisky and S. Jebens, *Superconvergent explicit two-step peer methods*, J. Comput. Appl. Math. 223 (2009), 753-764.
- [43] J. Wensch, O. Knoth and A. Galant, *Multirate infinitesimal step methods for atmospheric flow simulation*, BIT 49 (2009), 449-473.
- [44] L.J. Wicker and 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.
- [45] L.J. Wicker and W.C. Skamarock, *Time-Splitting Methods for Elastic Models Using Forward Time Schemes*, Mon. Wea. Rev. 130 (2002), 2088-2097.

Erklärung

Hiermit erkläre ich, Stefan Jebens, dass ich die vorliegende Dissertation selbstständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die aus anderen Werken wörtlich oder inhaltlich entnommenen Daten, Fakten und Konzepte sind unter Angabe der entsprechenden Quellen als solche gekennzeichnet. Diese Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem anderen Prüfungsverfahren vorgelegt.

Köln, den 21. April 2011

A handwritten signature in black ink, appearing to read 'Stefan Jebens', written over a horizontal line.

Stefan Jebens



Lebenslauf

Persönliche Daten

Stefan Jebens
am 17. März 1984 in
Hamburg geboren

Ausbildung

August 1990 – Juli 1992

Grundschule Moorflagen
(3. Klasse übersprungen)

August 1992 – Juli 1993

Katholische Schule Hamburg

August 1993 – Januar 1994

Johanneum
(humanistisches Gymnasium)

Februar 1994 – Januar 1997

Wilhelm-Gymnasium
(humanistisches Gymnasium)

Februar 1997 – Juli 1999

Albert-Schweitzer-Gymnasium
(allgemeines Gymnasium)

August 1999 – Juni 2002

Wirtschaftsgymnasium H7 City Nord
Abschluss mit Abitur, Note 1,1

- LK Mathematik: 1+
- LK Wirtschaft: 1
- GK Deutsch: 3+
- GK Physik: 1+

Oktober 2002 – Juni 2007

Mathematik-Studium an der Universität Halle
Abschluss als Diplom-Mathematiker, Note 1,4

- Vertiefungsrichtung: Numerische Mathematik
- Praktikumsarbeit: Parallele explizite Zweischritt-Peer-Methoden
- Diplomarbeit: Explizite Mehrschritt-Peer-Methoden für spezielle Systeme 2. Ordnung

seit Juli 2007

Promotion an der Universität Halle

Berufspraxis

August 1998 – September 1998

Praktikum bei der Lufthansa Technik AG

April 2004 – Oktober 2004

Callcenter-Agent für die Tele2 GmbH

November 2004 – Juni 2007

Nachhilfelehrer beim Lernstudio Barbarossa

Oktober 2005 – März 2006

Forschungspraktikum an der Universität Halle

Juli 2007 – August 2010

Wissenschaftlicher Mitarbeiter am Leibniz-Institut für
Troposphärenforschung

seit August 2010

Lehrer am Schiller-Gymnasium Köln

Stipendien

August 2001 – Juni 2002	Stipendiat der J. und E. Frauendorfer-Förderstiftung
seit Januar 2003	Stipendiat bei der e-fellows.net GmbH & Co. KG

Auszeichnungen

1994	3. Preis bei der Hamburger Landesrunde der 33. Mathematik-Olympiade
1995	Anerkennungsurkunde bei der Hamburger Landesrunde der 34. Mathematik-Olympiade
1996	Anerkennungsurkunde bei der Hamburger Landesrunde der 35. Mathematik-Olympiade
1999	3. Preis bei der Hamburger Landesrunde der 38. Mathematik-Olympiade
2008	Förderpreis der Georg-Cantor-Vereinigung
2009	Preis für die beste Publikation des Leibniz-Instituts für Troposphärenforschung

Publikationen

Applied Mathematics and Computations	Explicit multi-step peer methods for special second-order differential equations, Ausgabe 202 (2008), Seiten 803-813
Journal of Computational and Applied Mathematics	Superconvergent explicit two-step peer methods, Ausgabe 223 (2009), Seiten 753-764
Applied Numerical Mathematics	Parameter optimization for explicit parallel peer two-step methods, Ausgabe 59 (2009), Seiten 769-782
Monthly Weather Review	Explicit Two-Step Peer Methods for the Compressible Euler Equations, Ausgabe 137 (2009), Seiten 2380-2392
Applied Numerical Mathematics	Linearly Implicit Peer Methods for the Compressible Euler Equations, im Druck
Journal of Computational Physics	Partially Implicit Peer Methods for the Compressible Euler Equations, Ausgabe 230 (2011), Seiten 4955-4974

Interessen

Hobbys	Rock- und Metalkonzerte, Fahrrad fahren, joggen, lesen, Computerspiele, rudern, historisches Fechten
Soziales Engagement	Mitgliedschaft im World Wide Fund For Nature (WWF), Patenschaft für Pinguin am Zoo Leipzig, regelmäßiges Kochen in Volxküchen