

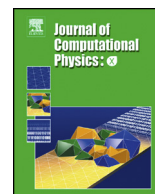


ELSEVIER

Contents lists available at ScienceDirect

Journal of Computational Physics: X

www.elsevier.com/locate/jcpX



An adaptive local time-stepping scheme for multiresolution simulations of hyperbolic conservation laws

Jakob W.J. Kaiser*, Nils Hoppe, Stefan Adami, Nikolaus A. Adams

Technical University of Munich, Department of Mechanical Engineering, Chair of Aerodynamics and Fluid Mechanics, Boltzmannstraße 15, 85748 Garching, Germany



ARTICLE INFO

Article history:

Received 3 December 2018

Received in revised form 9 July 2019

Accepted 20 August 2019

Available online 14 October 2019

Keywords:

Adaptive time stepping

Local time stepping

Multiresolution simulations

Shock-bubble interaction

ABSTRACT

We present an adaptive local time-stepping (ALTS) scheme for a block-structured multiresolution scheme of hyperbolic conservation laws for fluid flow. The stability of standard local time-stepping (LTS) schemes with level-dependent time-step sizes is improved by local time-step size adaptation when progressing through the underlying multi-stage time integration scheme. The novelty of the approach is that it merges flux computation and time integration of the state vector with projection and prediction operations of the multiresolution scheme [15]. This enables consistent time integration of subdomains with different refinement levels without the need for intermediate time synchronization which can be prohibitively expensive in parallel computations. Consequently, coarser subdomains are advanced in time only once finer subdomains have advanced to the same time instant. Full spatial resolution adaptivity for integrated regions after each substep is maintained.

The new scheme exhibits significantly improved numerical stability as compared to previous LTS schemes due to the local time-step size adaptation at each substep. The computational overhead of the incurred additional operations is small. In applications, the ALTS scheme demonstrates the same computational efficiency as standard LTS schemes.

The new scheme can be applied to any explicit single-step time-integration scheme and is independent of the employed spatial discretization scheme. The improved stability is demonstrated with several one- and two-dimensional examples of flows with one and two phases, applying second- and third-order Runge-Kutta time integration schemes.

© 2019 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Computational simulations of complex flows remain a challenge even today. The large range of temporal and spatial scales in combustive or turbulent flows or flow cavitation challenges the performance of even the largest parallel computers [26]. While a wide range of spatial scales causes high memory consumption, small temporal scales lead to small time-step sizes that inversely proportionally increase the number of required computational operations. Although the smallest spatial and temporal scales predominantly govern the flow evolution, in particular interfacial flows or flows with reaction fronts require the highest spatial resolution levels on co-dimension one-manifolds. Therefore, spatial and temporal resolution adaptation schemes have been and are being developed as a remedy to reduce computational cost.

* Corresponding author.

E-mail addresses: jakob.kaiser@tum.de (J.W.J. Kaiser), nils.hoppe@tum.de (N. Hoppe), stefan.adami@tum.de (S. Adami), nikolaus.adams@tum.de (N.A. Adams).

<https://doi.org/10.1016/j.jcpX.2019.100038>

2590-0552/© 2019 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

As an approach for spatial adaptivity, Harten [15,16] proposed a multiresolution (MR) framework for the spatial discretization of hyperbolic conservation laws to improve the efficiency of existing finite-volume solvers. The concept, which has been further developed to provide full spatial adaptivity [4], relies on a wavelet-based representation of the flow fields. This allows for compressing data to the minimum necessary amount for representing a field function at desired accuracy. Starting from a solution on a coarse mesh, higher-order wavelet coefficients are added successively to achieve the required accuracy using an increasingly finer mesh. In smooth flow regions only few wavelet coefficients may suffice to approximate the solution with the required accuracy on a coarse grid. In regions where the solution has strong local gradients, higher-wavenumber wavelets need to be used to represent the solution which results in a locally refined grid. The underlying refinement strategy is based on dyadic refinement, where the cell spacing is halved in each direction for every additional set of wavelet coefficients. Compared to other spatial adaption methods, such as adaptive mesh refinement (AMR), MR techniques result in improved CPU and memory compression rates for the same truncation-error order discretizations of hyperbolic conservation laws [6].

Spatially varying grid resolution affects the temporal integration of the governing equations. The choice of a stable time-step size Δt from a stability criterion, such as the CFL condition, depends on the local cell sizes. Constant time-stepping (CTS) schemes choose the minimum Δt based on the globally maximum wave speed $|u \pm c|$ together with the smallest occurring cell size to ensure stability in the entire domain [10,29]. Although formally and algorithmically simple, this approach is not very efficient as larger cells could use larger time-step sizes.

As a remedy, Osher and Sanders [25] introduced a local time-stepping (LTS) scheme for solving one-dimensional conservation laws on grids with varying spatial cell size. Each level of refinement uses its specific cell size to scale the time-step size Δt . For dyadic refinement in one dimension, the time step at each successively coarser level l is twice the time-step size of the finer level $l + 1$, thus the required number of time integration steps are strongly reduced compared to the CTS scheme. The LTS scheme has been combined with the MR approach by Domingues et al. [7–9] and Lopes et al. [22]. In their approach, the time-step size is fixed during each full cycle on the coarsest resolution level. At intermediate stages, interpolation is required in cells at resolution jumps. Müller and Stiriba [24] introduced transition zones between coarser and finer cells. This enables updating the time-step size after each full cycle on the finest levels and propagating this change upwards to coarser grid regions. Their scheme has been successfully employed for multi-fluid simulations [5]. However, their LTS algorithm is limited to stepwise grid refinements and cell-flux reconstruction schemes which require two adjacent cells only. Han et al. [12,13] applied the local time-stepping approach on a block-structured multiresolution scheme, with a fixed time-step size during each full cycle on the coarsest resolution level.

Several works [2,18,28] have demonstrated that the LTS approach with MR outperforms the computational efficiency of CTS. The minimum time-step size, however, can be determined only after each full cycle on the coarsest mesh, regardless of the interim local flow evolution on the finest mesh. Hence, the method does not deliver instantaneous time-step size adaptivity [8], and eventually may violate the stability criterion. Flows with strong temporal gradients, i.e. fast physical processes such as collapsing bubbles during cavitation [1] or chemical reactions [2] reveal such a deficiency of the LTS scheme. In the standard LTS approach, stability for such cases can be achieved by choosing a smaller CFL number, thus decreasing overall efficiency. Alternatively, the number of refinement levels can be reduced to allow more frequent time-step size adjustments, which decreases memory compression.

In this work, we present a revised LTS scheme without limitations on the spatial and temporal adaptivity for a block-structured MR scheme. Projection and prediction operations of Harten [15] are applied on the numerical divergence of the flux vector rather than the state vector of the finite-volume discretization of the governing equations, similar to the approach of Bihari and Harten [3]. By advancing also halo cells of cell blocks in time, time-step size adaptation for different resolutions in different domain regions without additional synchronization on coarser resolution levels is obtained. The time-step size is adapted after each full cycle on the finest level, thus recovering the stability properties of the CTS approach. This is not limited to single-stage time-integration schemes, but also applies to multi-stage time integration schemes, e.g., the two-stage second order Runge-Kutta (RK) Total Variation Diminishing (TVD) scheme [11,14]. In addition, the scheme does not constrain spatial discretization. Hence, it can be used with high-order spatial schemes, such as WENO, and does not limit the resolution jumps between neighboring domain regions.

The organization of the paper is the following: in section 2, the adaptive MR scheme for the finite volume (FV) method is reviewed. In section 3, we describe the new adaptive local time-stepping scheme. In section 4, we present simulation results of our fully adaptive scheme for one- and two-dimensional problems. We compare our simulation results with exact solutions and other numerical schemes to discuss the accuracy and stability of the new scheme. In section 5, we conclude our results.

2. Multiresolution scheme

2.1. Finite volume representation

We consider a finite-volume representation of hyperbolic conservation laws

$$\frac{\partial \bar{u}}{\partial t} + \nabla \cdot f(\bar{u}) = 0 \quad (1)$$

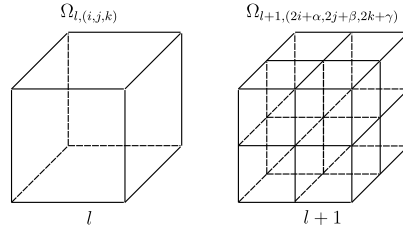


Fig. 1. Dyadic refinement scheme in 3D, following Domingues et al. [8].

on a Cartesian grid $(x, y, z, t) \in \Omega \times [0, \infty)$, $\Omega \subset \mathbb{R}^3$ to evolve the state vector \bar{u} . The governing equations (1) constitute an initial-boundary value problem with $\bar{u}(x, y, z, 0) = \bar{u}_0(x, y, z)$ and appropriate boundary conditions on the boundary $\Gamma = \partial\Omega$. In the finite volume approach, the cell-averaged numerical solution \bar{U} can be advanced in time by solving the ordinary differential equation (ODE) system

$$\frac{d\bar{U}}{dt} + \mathcal{D}(\bar{U}) = 0, \tag{2}$$

where $\mathcal{D}(\bar{U})$ represents the divergence of the evaluated numerical flux function. High-order shock capturing is achieved in this paper with a 5th-order WENO shock-capturing scheme for the reconstruction of the numerical flux function [19]. The ALTS scheme, however, can be applied with every other linear or nonlinear finite-volume or finite-difference scheme.

The semi-discrete ODE system (2) is integrated in time using the 2nd-order RK-TVD scheme

$$\bar{U}^{(n+1/2)} = \bar{U}^{(n)} + \Delta t^{(n)} \mathcal{D}(\bar{U}^{(n)}) \tag{3}$$

$$\bar{U}^{(n+1)} = \bar{U}^{(n)} + \Delta t^{(n)} \frac{1}{2} \left[\mathcal{D}(\bar{U}^{(n)}) + \mathcal{D}(\bar{U}^{(n+1/2)}) \right] \tag{4}$$

of Gottlieb and Shu [11]. The solution at time $t^{(n)}$ and $t^{(n+1)} = t^{(n)} + \Delta t^{(n)}$ is denoted as $\bar{U}^{(n)}$ and $\bar{U}^{(n+1)}$, respectively. The intermediate stage of the RK scheme is denoted as $\bar{U}^{(n+1/2)}$. The time-step size Δt is chosen according to a Courant-Friedrichs-Lewy (CFL) stability criterion

$$\Delta t \leq \text{CFL} \left(\sum_i \frac{\|u_i \pm c\|_\infty}{\Delta x_i} \right)^{-1} \tag{5}$$

with $\text{CFL} \leq 1$, where c , u_i and Δx_i are the local speed of sound, the velocity component in i -direction, and the cell size in i -direction of the Cartesian mesh, respectively [11]. We use the discrete evolution operator

$$\mathcal{E} = \mathcal{E}(\bar{U}^{(n)}, \mathcal{D}(\bar{U}_1), \mathcal{D}(\bar{U}_2), \Delta t^{(n)}) \tag{6}$$

as short notation for a single RK step [29]. For the first stage, it is $\mathcal{D}(\bar{U}_1) = \mathcal{D}(\bar{U}_2) = \mathcal{D}(\bar{U}^{(n)})$, for the second stage $\mathcal{D}(\bar{U}_1) = \mathcal{D}(\bar{U}^{(n)})$ and $\mathcal{D}(\bar{U}_2) = \mathcal{D}(\bar{U}^{(n+1/2)})$.

2.2. Multiresolution representation

Following the original concept of Harten [15], the cell average solution \bar{U} is represented by the combination of data on a coarse grid and a series of values on successively finer grids. We use a Cartesian grid with global indices i, j , and k , where each cell can be refined dyadically into 2^D children cells, with D being the number of dimensions. Starting from the parent cell states $\bar{U}_{l,(i,j,k)}$ at level l , the children at the next-finer level $l + 1$ are denoted as $\bar{U}_{l+1,(2i+\alpha,2j+\beta,2k+\gamma)}$, $\alpha, \beta, \gamma \in \{0, 1\}$. A sketch of this dyadic refinement strategy is shown in Fig. 1.

Two basic operations are introduced to exchange data between consecutive levels l and $l + 1$ [15]. Cell averages at level l are estimated from a finer level $l + 1$ by the projection operation $\mathcal{Q}(l + 1 \rightarrow l)$

$$\bar{U}_{l,(i,j,k)} = \mathcal{Q}\bar{U}_{l+1} = \frac{1}{2^D} \sum_{\alpha=0}^1 \sum_{\beta=0}^1 \sum_{\gamma=0}^1 \bar{U}_{l+1,(2i+\alpha,2j+\beta,2k+\gamma)}. \tag{7}$$

Note that we formulate the projection operator for 3D data sets. For lower dimensions, the indices k (2D) or j and k (1D) as well as γ and β are omitted. This operation is exact for cell-averaged data.

The prediction operation $\mathcal{P}(l \rightarrow l+1)$ is used to estimate cell averages at level $l+1$ from a coarser level l and is given by

$$\begin{aligned} \hat{U}_{l+1,(2i+i_0,2j+j_0,2k+k_0)} &= \mathcal{P}\bar{U}_l \\ &= \bar{U}_{l,(i,j,k)} + (-1)^{i_0} Q_x^s + (-1)^{j_0} Q_y^s + (-1)^{k_0} Q_z^s + \\ &\quad (-1)^{(i_0+j_0)} Q_{xy}^s + (-1)^{(i_0+k_0)} Q_{xz}^s + \\ &\quad (-1)^{(j_0+k_0)} Q_{yz}^s + (-1)^{(i_0+j_0+k_0)} Q_{xyz}^s \end{aligned} \quad (8)$$

with

$$\begin{aligned} Q_x^s &= \sum_{m=1}^s \gamma_m (\bar{U}_{l,(i+m,j,k)} - \bar{U}_{l,(i-m,j,k)}), \\ Q_y^s &= \sum_{n=1}^s \gamma_n (\bar{U}_{l,(i,j+n,k)} - \bar{U}_{l,(i,j-n,k)}), \\ Q_z^s &= \sum_{o=1}^s \gamma_o (\bar{U}_{l,(i,j,k+o)} - \bar{U}_{l,(i,j,k-o)}), \\ Q_{xy}^s &= \sum_{m=1}^s \gamma_m \sum_{n=1}^s \gamma_n (\bar{U}_{l,i+m,j+n,k} - \bar{U}_{l,i+m,j-n,k} - \bar{U}_{l,i-m,j+n,k} + \bar{U}_{l,i-m,j-n,k}), \\ Q_{xz}^s &= \sum_{m=1}^s \gamma_m \sum_{o=1}^s \gamma_o (\bar{U}_{l,i+m,j,k+o} - \bar{U}_{l,i+m,j,k-o} - \bar{U}_{l,i-m,j,k+o} + \bar{U}_{l,i-m,j,k-o}), \\ Q_{yz}^s &= \sum_{n=1}^s \gamma_n \sum_{o=1}^s \gamma_o (\bar{U}_{l,i,j+n,k+o} - \bar{U}_{l,i,j+n,k-o} - \bar{U}_{l,i,j-n,k+o} + \bar{U}_{l,i,j-n,k-o}), \\ Q_{xyz}^s &= \sum_{m=1}^s \gamma_m \sum_{n=1}^s \gamma_n \sum_{o=1}^s \gamma_o (\bar{U}_{l,i+m,j+n,k+o} - \bar{U}_{l,i+m,j+n,k-o} - \bar{U}_{l,i+m,j-n,k+o} \\ &\quad - \bar{U}_{l,i-m,j+n,k+o} + \bar{U}_{l,i+m,j-n,k-o} + \bar{U}_{l,i-m,j+n,k-o} + \bar{U}_{l,i-m,j-n,k+o} \\ &\quad - \bar{U}_{l,i-m,j-n,k-o}) \end{aligned}$$

and $i_0, j_0, k_0 \in \{0, 1\}$ depending on the location of the finer cell. The order of this central interpolation scheme is $r = 2s + 1$, with the stencil half-width s . The interpolation coefficients are $\gamma_1 = \{1/8\}$ for a 3rd order scheme ($s = 1$) and $\gamma_1 = \{-22/128, 3/128\}$ for a 5th order scheme ($s = 2$). For lower dimensions, the prediction operation is given by eq. (8) without higher-dimensional terms (2D: $Q_{xyz}^s = Q_{xz}^s = Q_{yz}^s = Q_z^s = 0$, 1D: additionally $Q_{xy}^s = Q_y^s = 0$). These operations can be applied on both the cell-averaged state vector and the numerical divergence of the flux functions [3].

Projection and prediction operations are *local* and *consistent* ($\mathcal{Q} \circ \mathcal{P} = Id$), but not commutative ($\mathcal{P} \circ \mathcal{Q} \neq Id$). This motivates the definition of the so-called *details* d as the difference between the exact solution and the predicted solution

$$d_{l+1,(2i,2j,2k)} = \bar{U}_{l+1,(2i,2j,2k)} - \hat{U}_{l+1,(2i,2j,2k)}. \quad (9)$$

This definition shows that knowing the cell average of the coarsest cell and details of all successively finer cells is equivalent to knowing the cell-averaged values of all finer cells. Implicit mesh adaption is achieved by considering details larger than a pre-defined level-dependent threshold ε , only. As details become negligible, i.e. $\|d\| < \varepsilon$, further grid refinement is not necessary, as a finer mesh does not lead to significantly better results than interpolation from a coarser grid.

For efficiency reasons, we follow the approach of Rossinelli et al. [27] and apply \mathcal{P} , \mathcal{Q} , and all other operations, e.g. the numerical flux estimation, always on a set of multiple cells, which is called a *block*. The refinement of cells occurs block-wise, too, i.e., all cells of a block are refined as soon as one single cell in the block triggers refinement. A two-dimensional example of the resulting tree structure for a maximum refinement level of $l_{max} = 3$ is shown in Fig. 2. In this example, we start with one single block on the coarsest level, which is successively refined three times to give the maximum resolution in the region of interest.

The following definitions are introduced for the tree-structure to reflect the relationship between different blocks. Taking the single block at the coarsest level $l = 0$ from Fig. 2, the four blocks which are at the next finer level $l = 1$ are called *children*, the blocks at the next level $l = 2$ are correspondingly the *grandchildren*. Vice versa, we speak of *parent* and *grandparent* blocks. When a block does not have any children, it is called a *leaf*. The governing equations are evolved in time only on leaves, all other blocks receive their values from their children by the projection operation eq. (7).

When performing \mathcal{E} (or other operations) close to the edge of a block, data from neighboring blocks are required. We apply the approach of Han et al. [12] who use halo cells for each block to facilitate these operations without requiring

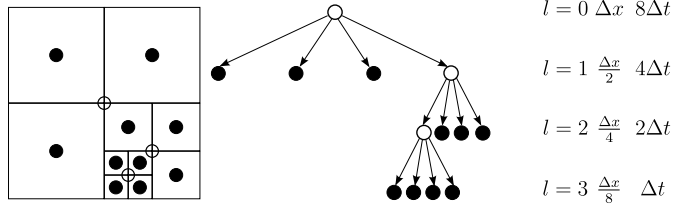


Fig. 2. Sketch of the multiresolution discretization: Grid representation with locally refined regions (left) with $l_{max} = 3$ in two dimensions. The graph shows the tree-structure with corresponding level labels, cell sizes, and time-step sizes for a local time-stepping algorithm. In the grid representation and the tree, existing blocks are shown as circles, where empty circles denote further refined blocks and filled circles mark leaves.

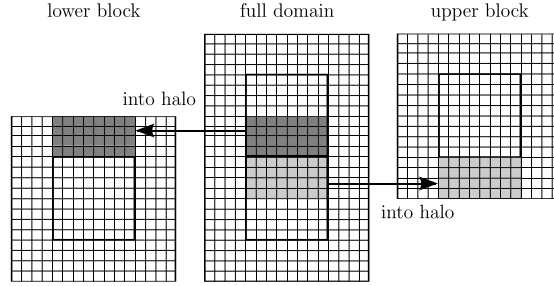


Fig. 3. Relationship of internal cells and halo cells for two neighboring blocks on the same level. In the center, the actual domain of two blocks with surrounding halo cells for external boundary conditions is shown. Each block is displayed separately to the left and right with its own halo cells. The internal cells which are halo cells for these blocks are marked grey.

frequent data exchange with neighboring blocks. Halo cells overlap with internal cells from neighboring blocks or external boundary conditions and contain copies of their states without being evolved themselves. These halo cells are then used to update the solution of internal cells in time. Halo cells need to be updated after each stage to be consistent. A 2D example for a domain with two blocks is shown in Fig. 3.

When evaluating the divergence of the numerical flux function $\mathcal{D}(\bar{U})$ at resolution jumps, conservation may be violated. For two neighboring leaves at levels l and $l + m$, with the outgoing flux being computed at level l and the incoming flux at level $l + m$ at their block boundaries, the flux-reconstruction operation performed on different levels may result in a mismatch of the incoming and outgoing fluxes. This mismatch is corrected by overwriting the fluxes of the coarser cell at the resolution jump with the fluxes computed on the finer mesh [29]. Note that fluxes of successive iterations need to be scaled with the different time-step sizes to ensure conservation. This way, we maintain strict conservation at cell patches between leaves on different resolution levels.

3. Adaptive local time-stepping scheme

The fixed cell-size ratio of successive refinement levels allows for the efficient formulation of a local time-stepping (LTS) scheme [8]. The level-dependent increment Δt_l is scaled from the finest admissible stepsize at the maximum refinement level $\Delta t_{l_{max}}$

$$\Delta t_l = \Delta t_{l_{max}} \cdot 2^{l_{max}-l} \tag{10}$$

for an arbitrary level l , where l_{max} is the maximum number of refinement levels, and $\Delta t_{l_{max}}$ the time-step size computed with the cell size of the finest level. This leads to an improved efficiency of the scheme as compared to the CTS approach, where all cells are integrated in time with $\Delta t_{l_{max}}$ regardless of their resolution. Drawbacks of the LTS scheme are the necessary time synchronization at resolution jumps and the constant time-step size Δt_l within each full cycle of the coarsest level [8]. Therefore, the time-step size adaption loses efficiency with increasing number of refinement levels. For example, for a setup with $l_{max} = 4$, $\Delta t_{l_{max}}$ is updated only after $2^4 = 16$ full cycles on l_{max} . For a setup with $l_{max} = 8$, the earliest possible adjustment of $\Delta t_{l_{max}}$ occurs after $2^8 = 256$ full cycles on l_{max} . In case of strong non-linear effects such as wave-steepening or bubble collapse events, this procedure may result in severe stability issues.

In this work, we propose a fully adaptive local time-stepping (ALTS) algorithm for multiresolution simulations to solve the aforementioned limitations. We split the evolution operation \mathcal{E} into the evaluation of the divergence of the numerical flux functions $\mathcal{F} = \mathcal{F}(\bar{U})$, and the time integration $\mathcal{T} = \mathcal{T}(\bar{U}^{(n)}, \mathcal{D}(\bar{U}_1), \mathcal{D}(\bar{U}_2), \Delta t^{(n)})$, so that

$$\mathcal{T} \circ \mathcal{F} = \mathcal{E}. \tag{11}$$

The key step to allow for instantaneous time-step size adaptivity is to apply the projection and prediction operations to the numerical divergence of the flux functions in halo cells at resolution jumps. The halo cells are integrated in time together

with the internal cells according to their level-dependent time-step size. This is mathematically equivalent to performing first the time integration of the internal cells followed by the projection and prediction to update the halo cells, which is shown in the following derivation.

The projection operation (7) with an Euler-forward time update can be written as

$$\begin{aligned}
\bar{U}_{l,(i,j,k)}^{(n+1)} &= \mathcal{Q}\bar{U}_{l+1,(2i,2j,2k)}^{(n+1)} \\
&= \frac{1}{2^D} \sum_{\alpha=0}^{i_{\max}} \sum_{\beta=0}^{j_{\max}} \sum_{\gamma=0}^{k_{\max}} \bar{U}_{l+1,(2i+\alpha,2j+\beta,2k+\gamma)}^{(n+1)} \\
&= \frac{1}{2^D} \sum_{\alpha=0}^{i_{\max}} \sum_{\beta=0}^{j_{\max}} \sum_{\gamma=0}^{k_{\max}} \left[\bar{U}_{l+1,(2i+\alpha,2j+\beta,2k+\gamma)}^{(n)} + \Delta t_{l+1} \mathcal{D} \left(\bar{U}_{l+1,(2i+\alpha,2j+\beta,2k+\gamma)}^{(n)} \right) \right] \\
&= \frac{1}{2^D} \sum_{\alpha=0}^{i_{\max}} \sum_{\beta=0}^{j_{\max}} \sum_{\gamma=0}^{k_{\max}} \bar{U}_{l+1,(2i+\alpha,2j+\beta,2k+\gamma)}^{(n)} \\
&\quad + \Delta t_{l+1} \frac{1}{2^D} \sum_{\alpha=0}^{i_{\max}} \sum_{\beta=0}^{j_{\max}} \sum_{\gamma=0}^{k_{\max}} \mathcal{D} \left(\bar{U}_{l+1,(2i+\alpha,2j+\beta,2k+\gamma)}^{(n)} \right) \\
&= \bar{U}_{l,(i,j,k)}^{(n)} + \Delta t_{l+1} \mathcal{Q} \mathcal{D} \left(\bar{U}_{l+1,(2i,2j,2k)}^{(n)} \right) \\
&= \bar{U}_{l,(i,j,k)}^{(n)} + \Delta t_{l+1} (\mathcal{Q} \circ \mathcal{F}) \bar{U}_{l+1,(2i,2j,2k)}^{(n)}.
\end{aligned}$$

The prediction operation (8) yields

$$\begin{aligned}
\hat{U}_{l+1,(2i+i_0,2j+j_0,2k+k_0)}^{(n+1)} &= \mathcal{P}\bar{U}_{l,(i,j,k)}^{(n+1)} \\
&= \bar{U}_{l,(i,j,k)}^{(n+1)} + \\
&\quad (-1)^{i_0} Q_x^{s,(n+1)} + (-1)^{j_0} Q_y^{s,(n+1)} + (-1)^{k_0} Q_z^{s,(n+1)} + \\
&\quad (-1)^{(i_0+j_0)} Q_{xy}^{s,(n+1)} + (-1)^{(i_0+k_0)} Q_{xz}^{s,(n+1)} + \\
&\quad (-1)^{(j_0+k_0)} Q_{yz}^{s,(n+1)} + (-1)^{(i_0+j_0+k_0)} Q_{xyz}^{s,(n+1)} \\
&= \bar{U}_{l,(i,j,k)}^{(n)} + \Delta t_l \mathcal{D} \left(\bar{U}_{l,(i,j,k)}^{(n)} \right) + \\
&\quad (-1)^{i_0} Q_x^{s,(n+1)} + (-1)^{j_0} Q_y^{s,(n+1)} + (-1)^{k_0} Q_z^{s,(n+1)} + \\
&\quad (-1)^{(i_0+j_0)} Q_{xy}^{s,(n+1)} + (-1)^{(i_0+k_0)} Q_{xz}^{s,(n+1)} + \\
&\quad (-1)^{(j_0+k_0)} Q_{yz}^{s,(n+1)} + (-1)^{(i_0+j_0+k_0)} Q_{xyz}^{s,(n+1)}.
\end{aligned}$$

The $Q_\alpha^{s,(n+1)}$ terms change accordingly, for any component α . For example, $\alpha = x$ can be written as

$$\begin{aligned}
Q_x^{s,(n+1)} &= \sum_{m=1}^s \gamma_m \left(\bar{U}_{l,(i+m,j,k)}^{(n+1)} - \bar{U}_{l,(i-m,j,k)}^{(n+1)} \right) \\
&= \sum_{m=1}^s \gamma_m \left(\bar{U}_{l,(i+m,j,k)}^{(n)} + \Delta t_l \mathcal{D} \left(\bar{U}_{l,(i+m,j,k)}^{(n)} \right) - \bar{U}_{l,(i-m,j,k)}^{(n)} - \Delta t_l \mathcal{D} \left(\bar{U}_{l,(i-m,j,k)}^{(n)} \right) \right) \\
&= \sum_{m=1}^s \gamma_m \left(\bar{U}_{l,(i+m,j,k)}^{(n)} - \bar{U}_{l,(i-m,j,k)}^{(n)} \right) + \\
&\quad \Delta t_l \sum_{m=1}^s \gamma_m \left(\mathcal{D} \left(\bar{U}_{l,(i+m,j,k)}^{(n)} \right) - \mathcal{D} \left(\bar{U}_{l,(i-m,j,k)}^{(n)} \right) \right) \\
&= Q_x^{s,(n)} + \Delta t_l \sum_{m=1}^s \gamma_m \left(\mathcal{D} \left(\bar{U}_{l,(i+m,j,k)}^{(n)} \right) - \mathcal{D} \left(\bar{U}_{l,(i-m,j,k)}^{(n)} \right) \right) \\
&= Q_x^{s,(n)} + \Delta t_l Q_{\mathcal{D},x}^{s,(n)}.
\end{aligned}$$

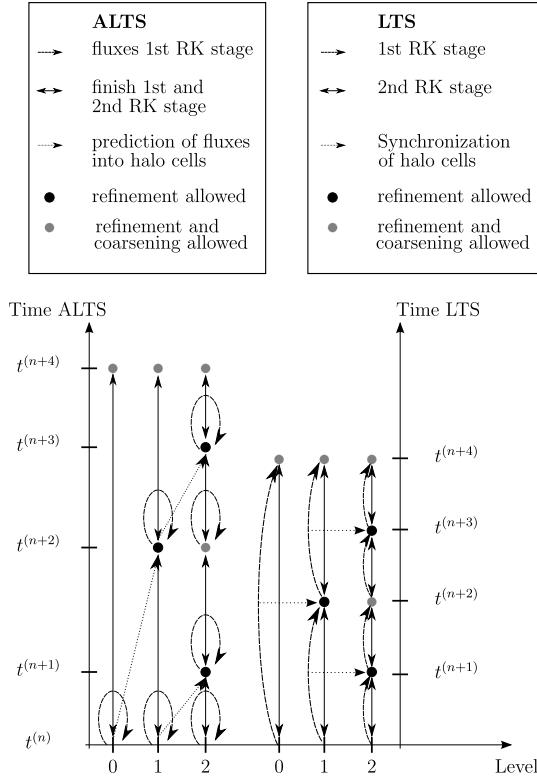


Fig. 4. Sketch of the time integration scheme: ALTS with $\Delta t^{(n+i)} \neq \Delta t^{(n+j)}$, $i \neq j$ (left) and classical LTS (right). See also Domingues et al. [8] for a similar presentation of LTS.

Combining these operations for all terms $Q_\alpha^{s,(n+1)}$ demonstrates the equivalence of the new approach with the standard time integration and projection operations of cell-averaged values (here for $i_0 = j_0 = k_0 = 0$)

$$\begin{aligned}
 \hat{U}_{l+1,(2i,2j,2k)}^{(n+1)} &= \bar{U}_{l,(i,j,k)}^{(n)} + Q_x^{s,(n)} + Q_y^{s,(n)} + Q_z^{s,(n)} + \\
 &\quad Q_{xy}^{s,(n)} + Q_{xz}^{s,(n)} + Q_{yz}^{s,(n)} + Q_{xyz}^{s,(n)} + \\
 &\quad \Delta t_l \left[\mathcal{D} \left(\bar{U}_{l,(i,j,k)}^{(n)} \right) + Q_{\mathcal{D},x}^{s,(n)} + Q_{\mathcal{D},y}^{s,(n)} + Q_{\mathcal{D},z}^{s,(n)} + \right. \\
 &\quad \left. Q_{\mathcal{D},xy}^{s,(n)} + Q_{\mathcal{D},xz}^{s,(n)} + Q_{\mathcal{D},yz}^{s,(n)} + Q_{\mathcal{D},xyz}^{s,(n)} \right] \\
 &= \hat{U}_{l+1,(2i,2j,2k)}^{(n)} + \Delta t_l (\mathcal{P} \circ \mathcal{F}) \bar{U}_{l,(i,j,k)}^{(n)}.
 \end{aligned}$$

In other words, instead of integrating cells close to resolution jumps in time, we only compute the divergence of cell-face fluxes for both coarser and finer cells at the resolution jump. Then, we exchange them in the halo cells and integrate them in time for the finer block with its level-dependent time-step size. Halo cells at resolution jumps are now at the same time instant as the internal cells, hence we avoid costly synchronization in time which is necessary for the standard LTS scheme. In fact, the temporal integration of the coarser level is postponed until all finer levels reach the same time. This allows for adapting the time-step size after every full cycle on the finest level, and ensures strongly improved time-step size adaptivity.

Our scheme imposes no limits on the number of resolution jumps between two neighboring leaves. Assume two neighboring blocks, with block A being a leaf at level l and block B a leaf at level $l + m$ with $m = 2$. First, halo values are exchanged at level l , thus the grandparent block of block B has updated halo values. Second, the halo cells are predicted twice, first to the parent block of block B, and from there to block B itself. Third, the predictions in the halo cells at level $l + 2$ are evaluated 2^2 times until the neighboring levels are synchronous again. This scheme works accordingly for $m > 2$.

Fig. 4 illustrates our new algorithm in comparison to the classical LTS. The basic algorithm works as follows: the divergence of cell-face fluxes on all parent and child levels are computed. They are projected to update divergences of parent blocks in the full tree, exchanged with neighboring blocks in their halo cells, and then predicted to update halo cells at resolution jumps. The evolution is finished by finally performing the integration in time on the target level. In short notation, the full step is abbreviated with the updated evolution operator

$$\tilde{\mathcal{E}} = \mathcal{T} \circ \mathcal{P} \circ \mathcal{Q} \circ \mathcal{F}. \tag{12}$$

Following Domingues et al. [8], the three basic steps to evolve the numerical solution from $\bar{U}^{(n)}$ to $\bar{U}^{(n+1)}$ can be expressed as

1. **Refinement** : $\mathcal{R}\bar{U}^{(n)} \rightarrow \bar{U}^{(n+)}$
2. **Evolution** : $\bar{\mathcal{E}}\bar{U}^{(n+)} \rightarrow \tilde{\bar{U}}^{(n+1)}$
3. **Coarsening** : $\mathcal{C}(\varepsilon)\tilde{\bar{U}}^{(n+1)} \rightarrow \bar{U}^{(n+1)}$.

Here \mathcal{R} is the refinement operator and $\mathcal{C}(\varepsilon)$ the coarsening operator with the coarsening threshold ε .

The main aspects of one time-step iteration in this scheme are:

1. Each macro cycle extends from $t^{(n)}$ to $t^{(n+2^{l_{max}})}$.
2. Each sub-cycle can be described by the tuple (τ, ζ) . $0 \leq \tau < 2^{l_{max}}$ describes the number of full time-step iterations on the finest level (micro time step) per iteration on the coarsest level (macro time step). $0 \leq \zeta < \zeta_{max}$ describes the number of intermediate steps for the explicit time integration scheme. E.g., for the 2nd-order RK-TVD scheme employed here, $\zeta_{max} = 2$.
3. All sub-cycles τ with $\zeta = 0$ start with an update of the current time-step size at the finest level $\Delta t_{l_{max}}^{\tau}$. Time-step sizes on all coarser levels accumulate the evolved steps and yield

$$\Delta t_l^{\tau} = \sum_{i=0}^{2^{(l_{max}-l)}-1} \Delta t_{l_{max}}^{\tau+i}. \quad (13)$$

4. In the first cycle of each macro time step, i.e. cycle $(0, 0)$, we compute the divergence of the fluxes (\mathcal{F}) for the first RK stage for all levels. We then perform the projection and prediction of the divergence to update halo cells at resolution jumps (\mathcal{Q}, \mathcal{P}).
5. In each following sub-cycle (τ, ζ) , leaves at all levels l which fulfill the constraint

$$(\tau + \zeta) \bmod 2^{l_{max}-l} = 0$$

are integrated in time (\mathcal{T}). Afterwards, cells at the parent levels are updated through cell average projection (\mathcal{Q}). Then, flux divergences are computed (\mathcal{F}), projection is performed, and halo cells at jump conditions are filled by prediction (\mathcal{Q}, \mathcal{P}).

6. In each sub-cycle, refinement (\mathcal{R}) and coarsening (\mathcal{C}) can be performed after time integration (\mathcal{T}). The condition for refining level l again is

$$(\tau + \zeta) \bmod 2^{l_{max}-l} = 0,$$

i.e. this level was integrated in this sub-cycle. For coarsening, the more restrictive condition

$$(\tau + \zeta) \bmod 2^{l_{max}-(l-1)} = 0$$

is applied. Note that a leaf at level l can only be removed when also the parent level was updated in time during this iteration. This is required to allow for a consistent computation of the details as refinement criterion.

7. The last sub-cycle $(2^{l_{max}} - 1, \zeta_{max} - 1)$ ends with the time integration of the second RK stage for all levels.

The scheme allows an update of $\Delta t_{l_{max}}^{\tau}$ during each full cycle at the finest level. The update ensures that the CFL-condition is fulfilled at every level at all times τ , which is demonstrated in the following, for simplicity, for the two finest levels l_{max} and $l_{max} - 1$. The maximum wave-speed $\|u_i \pm c\|_{\infty}^{\tau}$ is computed at all levels from the last completed full RK step and is used to determine the time-step size $\Delta t_{l_{max}}^{\tau}$ from the CFL-condition (5), giving $\Delta t_{l_{max}}^{\tau}$. The timestep size $\Delta t_{l_{max}}^{\tau+1}$ is computed analogously from the maximum wave-speed $\|u_i \pm c\|_{\infty}^{\tau+1}$. The time-step size at level $l_{max} - 1$ is $\Delta t_{l_{max}-1}^{\tau} = \Delta t_{l_{max}}^{\tau} + \Delta t_{l_{max}}^{\tau+1}$. We analyze at which level the maximum wave-speed $\|u_i \pm c\|_{\infty}^{\tau}$ occurs during two time steps on the finest level to estimate whether the time-step size at levels l_{max} and $l_{max} - 1$ is within the bounds of the CFL-criterion:

- $\|u_i \pm c\|_{\infty}^{\tau} = \|u_i \pm c\|_{\infty, l_{max}}^{\tau} > \|u_i \pm c\|_{\infty, l_{max}-1}^{\tau}$ and $\|u_i \pm c\|_{\infty}^{\tau+1} = \|u_i \pm c\|_{\infty, l_{max}}^{\tau+1} > \|u_i \pm c\|_{\infty, l_{max}-1}^{\tau}$: the maximum wave-speed in the domain occurs for both time steps at level l_{max} , the maximum admitted time-step size at level $l_{max} - 1$ therefore is larger than $\Delta t_{l_{max}-1}^{\tau} = \Delta t_{l_{max}}^{\tau} + \Delta t_{l_{max}}^{\tau+1}$. Hence, the CFL-condition is fulfilled at $l_{max} - 1$ and l_{max} .
- $\|u_i \pm c\|_{\infty}^{\tau} = \|u_i \pm c\|_{\infty, l_{max}-1}^{\tau} > \|u_i \pm c\|_{\infty, l_{max}}^{\tau}$ and $\|u_i \pm c\|_{\infty}^{\tau+1} = \|u_i \pm c\|_{\infty, l_{max}-1}^{\tau+1} > \|u_i \pm c\|_{\infty, l_{max}}^{\tau+1}$: the maximum wave-speed in the domain occurs for both time steps at level $l_{max} - 1$, therefore $\Delta t_{l_{max}-1}^{\tau} = 2 \cdot \Delta t_{l_{max}}^{\tau} = 2 \cdot \Delta t_{l_{max}}^{\tau+1}$. Hence, the CFL-condition is fulfilled at $l_{max} - 1$ and l_{max} .

- $\|u_i \pm c\|_\infty^\tau = \|u_i \pm c\|_{\infty, l_{max-1}}^\tau > \|u_i \pm c\|_{\infty, l_{max}}^\tau$ and $\|u_i \pm c\|_\infty^{\tau+1} = \|u_i \pm c\|_{\infty, l_{max}}^{\tau+1} > \|u_i \pm c\|_{\infty, l_{max-1}}^\tau$: the level of the maximum wave-speed changes after the first time step. For level $l_{max} - 1$, $\Delta t_{l_{max}-1}^\tau = 2 \cdot \Delta t_{l_{max}}^\tau$ satisfies the CFL-condition exactly, $2 \cdot \Delta t_{l_{max}}^{\tau+1} < 2 \cdot \Delta t_{l_{max}}^\tau$ is too restrictive, hence $\Delta t_{l_{max}-1}^\tau = \Delta t_{l_{max}}^\tau + \Delta t_{l_{max}}^{\tau+1}$ also satisfies the CFL-condition at both levels $l_{max} - 1$ and l_{max} .

Thus, the ALTS scheme provides a CFL-number which globally is always within the bounds of the prescribed stability criterion for every RK cycle.

4. Examples

The following examples illustrate the accuracy, improved stability, and efficiency of the proposed scheme. Grid adaptation is employed using the MR analysis with level- and dimension-dependent error levels

$$\varepsilon_l = \varepsilon_{\text{ref}} \cdot e^{(-D \cdot (l - l_{max}))}, \tag{14}$$

with the number of dimensions D . Unless stated otherwise, we use $\varepsilon_{\text{ref}} = 0.01$ for the refinement from $l_{max} - 1$ to l_{max} , and each simulation is performed with one block at level 0 using 16 internal cells in each spatial direction.

4.1. Accuracy-order analysis

Two one-dimensional test cases are presented to demonstrate the accuracy of the underlying methods of our adaptive local time-stepping scheme. First, we solve the linear scalar advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \tag{15}$$

for a sinusoidal profile $u_0(x) = 1 + 0.25 \sin(2\pi x)$ on the one-dimensional periodic domain $x \in [0, 1]$. The problem is advanced to a final time of $t = 1$, when the wave is again at its initial position. The second test case is the non-linear steepening of a sinusoidal pressure wave $p_0(x) = 1 + 0.25 \sin(2\pi x)$ on the one-dimensional periodic domain $x \in [0, 1]$. The one-dimensional Euler equations

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial u \rho}{\partial x} &= 0, \\ \frac{\partial \rho u}{\partial t} + \frac{\partial u \rho u}{\partial x} + \frac{\partial p}{\partial x} &= 0, \\ \frac{\partial \rho e}{\partial t} + \frac{\partial u(\rho e + p)}{\partial x} &= 0, \end{aligned} \tag{16}$$

govern the problem, where ρ is the density, u the velocity, p the pressure, and e the specific energy. The system is closed by the stiffened-gas equation-of-state

$$p = (\gamma - 1)\rho e - \gamma p_\infty, \tag{17}$$

with the material parameters γ for the ratio of specific heats and p_∞ for the background pressure. Here, we use $\gamma = 1.4$ and $p_\infty = 0$. Initial conditions for velocity and density fields are obtained from the Rankine-Hugoniot conditions for an ideal gas ($\gamma = 1.4$, $p_\infty = 0$). The problem is advanced to a final time of $t = 0.2$, when the wave has steepened already but still is continuous. For both cases, we present convergence rates for the spatial scheme (WENO 5), the standard RK2-TVD time-integration scheme, and the RK3-TVD time-integration scheme to demonstrate the applicability to higher-order explicit time-integration schemes. The simulations for the spatial error are run with $\text{CFL} = 10^{-3}$ to minimize the temporal error, and $\varepsilon_{\text{ref}} = 0.001$ for a refinement from level 0 to level 1. The simulations for the temporal error are computed on a mesh with maximum level $l_{max} = 4$.

Error plots of the L_1 norm of the density field

$$L_1 = \frac{1}{V} \sum_n \frac{|\rho_{n, \text{exact}} - \rho_{n, \text{sim}}|}{\rho_{n, \text{exact}}} dV_n, \tag{18}$$

are given in Fig. 5 for the two cases, numeric values are given in Table 1. The convergence rates of both the linear and the non-linear examples agree well with the values expected from theory. Only close to $\text{CFL} = 1.0$, the convergence rate of RK2 deviates from the expected value. Note that for RK3 the second-order treatment at resolution jumps [7,22] reduces the global order of the convergence to slightly less than three.

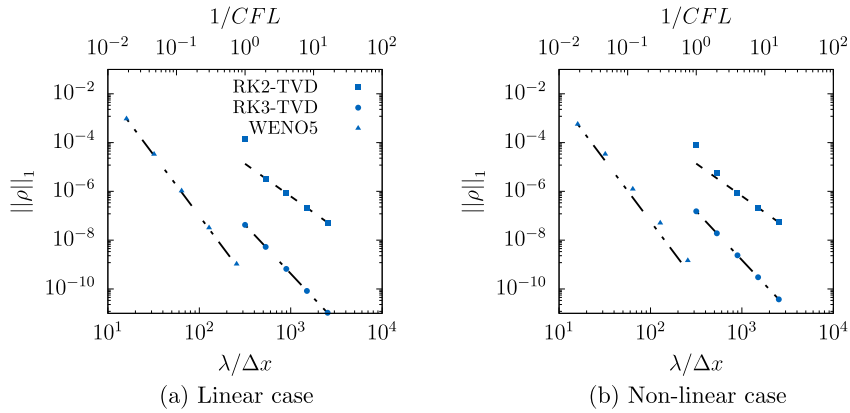


Fig. 5. Accuracy-order analysis for the linear (a) and non-linear (b) advection cases: temporal errors for RK2-TVD (■) and RK3-TVD (●), and spatial error for WENO5 (▲). For comparison, convergence plots for second (— —), third (— ·), and fifth (— · ·) order are given.

Table 1

The spatial and temporal accuracy of the linear sine-wave advection and non-linear sine-wave steepening for RK2-TVD, RK3-TVD, and WENO5 schemes.

Type	resolution	L ₁ error linear	L ₁ accuracy linear	L ₁ error non-linear	L ₁ accuracy non-linear
temporal (RK2-TVD)	1	1.3450E-4		7.7143E-5	
	0.5	3.3719E-6	6.32	5.4441E-6	3.76
	0.25	8.4330E-7	2.00	8.8377E-7	2.48
	0.125	2.1083E-7	2.00	2.1691E-7	2.02
	0.0625	5.2706E-8	2.00	5.3878E-8	2.01
temporal (RK3-TVD)	1	4.2420E-8		1.5239E-7	
	0.5	5.3045E-9	2.83	1.9113E-8	2.82
	0.25	6.6339E-10	2.83	2.3964E-9	2.82
	0.125	8.3011E-11	2.83	3.0006E-10	2.83
	0.0625	1.0551E-11	2.81	3.7514E-11	2.83
spatial (WENO5)	16	9.4091E-4		5.6337E-4	
	32	3.3506E-5	5.30	3.3520E-5	4.10
	64	1.0500E-6	5.65	1.2419E-6	5.20
	128	3.2178E-8	5.71	5.0419E-8	4.96
	256	1.0517E-9	5.53	1.4527E-9	5.89

4.2. One-dimensional examples

Two one-dimensional testcases with strong discontinuities and therefore strong temporal gradients in the wave-speeds are considered. The simulations are again governed by the compressible Euler equations (16).

The first setup is the inviscid shock-tube problem of Sod [30]. The initial conditions for a domain of length 1.0 are

$$(\rho, u, p) = \begin{cases} (1.0, 0.0, 1.0) & , 0.0 \leq x < 0.5 \\ (0.125, 0.0, 0.1) & , 0.5 \leq x \leq 1.0 \end{cases} \quad (19)$$

The domain is refined to $l_{max} = 7$ and the final simulation time is $t = 0.2$.

The second one-dimensional case is the two blast-waves problem of Woodward and Colella [31]. The initial conditions are

$$(\rho, u, p) = \begin{cases} (1.0, 0.0, 1000.0) & , 0.0 \leq x < 0.1 \\ (1.0, 0.0, 0.01) & , 0.1 \leq x < 0.9 \\ (1.0, 0.0, 100.0) & , 0.9 \leq x \leq 1.0 \end{cases} \quad (20)$$

The domain is refined to $l_{max} = 7$ and the final simulation time is $t = 0.038$.

Results for both 1D test cases are shown in Fig. 6. The applied CFL-number is $CFL = 1.0$. The standard local time stepping cannot resolve the wave dynamics for this high CFL-number and becomes unstable within the first macro time step in both cases. The ALTS scheme allows for stable simulations in both cases. The results show good agreement with the ones from an equally-spaced grid without adaptive mesh. This underlines that the ALTS scheme shows the same temporal stability behavior as an homogeneous grid without local spatial adaptation, which is not the case for the classical LTS scheme.

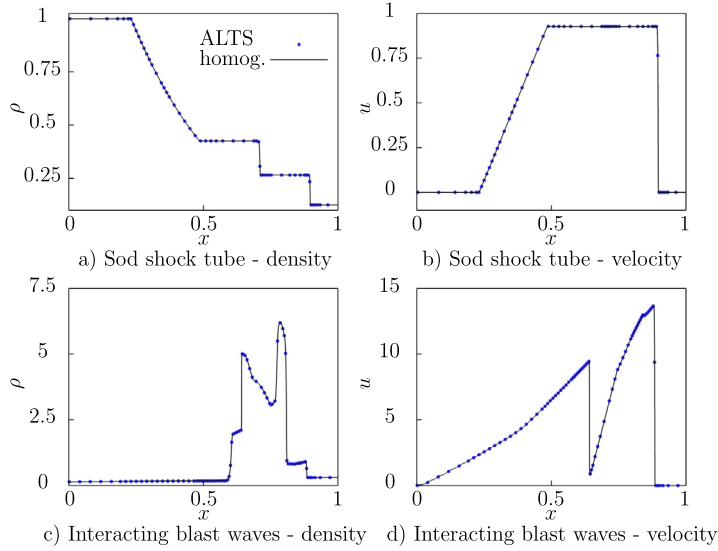


Fig. 6. One-dimensional examples. Upper half: Sod shock tube, density a) and velocity b) profiles at $t = 0.2$. Lower half: interacting blast waves, density c) and velocity d) at $t = 0.038$. Shown are results for a homogeneous grid (solid line) and an adaptive grid ($l_{max} = 7$) applying ALTS (dots) with CFL = 1.0. The results of every tenth cell are shown for ALTS.

4.3. Two-dimensional example

As two-dimensional test case we present the non-symmetric shock-induced collapse of a bubble near a rigid wall [20]. This problem is of practical relevance in a range of applications, e.g. shock-induced lithotripsy [21]. The wave-speed increases strongly during the bubble collapse, therefore instant adaption of the time-step size is required. Initial conditions for an axisymmetric domain $[4.0 \times 8.0] \mu\text{m}$, discretized with $l_{max} = 7$, two blocks in y-direction and one block in x-direction on $l = 0$ (effective resolution 2048×4096 cells), are

$$(\rho, u, v, p) = \begin{cases} (1011.05, 0.0, 21.36, 351 \times 10^5) & \text{post-shock water,} \\ (998.0, 0.0, 0.0, 10^5) & \text{pre-shock water,} \\ (1.2, 0.0, 0.0, 10^5) & \text{air bubble} \end{cases} \quad (21)$$

for a 35 MPa shock, which hits the bubble at $t = 0.0$. To model the second phase, we introduce a level-set function. The initial level-set field for the bubble is

$$\phi = -R_0 + \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (22)$$

with the bubble radius $R_0 = 50 \mu\text{m}$, $x_0 = 0$, and $y_0 = 0.7 \text{ mm}$. This leads to a stand-off distance of $2R_0$ of the bubble center from the wall. The bubble is resolved with 256 cells per initial radius on l_{max} , and we use CFL = 0.9 for the simulation. A sketch of this setup, which was investigated by Johnsen and Colonius [20], is shown in Fig. 7. We consider the compressible Euler equations in axisymmetric formulation and employ a level-set approach for multi-phase flows. The system is closed by the complete stiffened-gas equation-of-state

$$p + \gamma p \infty = (\gamma - 1)\rho(e + E_\infty), \quad (23)$$

where E_∞ is the energy translation factor. We use $\gamma = 4.4$, $p_\infty = 6.0 \times 10^8$, $E_\infty = 7.456 \times 10^6$ for water, and $\gamma = 1.4$, $p_\infty = 0$, $E_\infty = 0$ for air [17,23].

The pressure and axial velocity fields are shown in Fig. 8 at $t = 2.47 \text{ ms}$ and $t = 2.48 \text{ ms}$ for both schemes. At $t = 2.47 \text{ ms}$, the bubble is already strongly deformed. The maximum pressure regions are slightly shifted from the center axis, and coincide with the location where the bubble is thinnest (a). The high-speed water-jet has formed in the inner part of the bubble (b). For this instant, both schemes exhibit the same results. At $t = 2.48 \text{ ms}$, the jet strikes the downstream bubble wall, and the bubble breaks up at the location of the maximum pressure in the previous instant. The jet accelerates even further, and a strong shock wave can be observed propagating away from the bubble (c, d). This leads to a strong increase in the wave speed, so the time-step size needs to be adapted immediately to satisfy the stability criterion. The LTS scheme cannot provide this immediate adaption, and spurious oscillations occur at the wave front of the water hammer. These oscillations indicate the onset of nonlinear numerical instability near the wave front. The ALTS provides immediate time-step size adaption, and does not exhibit such spurious oscillations.

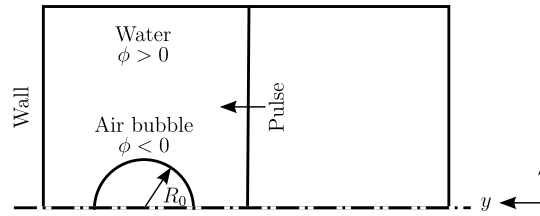


Fig. 7. Initial conditions for the shock-induced bubble collapse near a rigid wall in two dimensions.

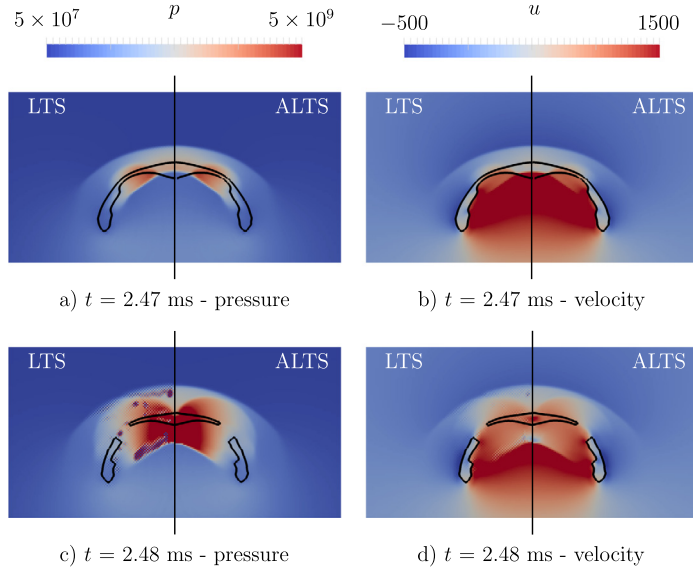


Fig. 8. Contour plots for LTS (left) and ALTS (right) at two different time instants: at $t = 2.47$ ms pressure (a) and streamwise velocity (b), and at $t = 2.48$ ms pressure (c) and streamwise velocity (d). The interface is denoted by the black line.

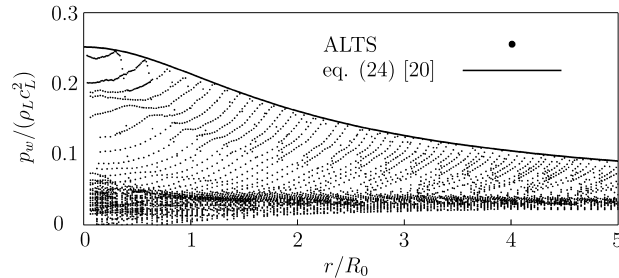


Fig. 9. Pressure profile along the wall at different instants with the analytical approximation of [20] (eq. (24)).

Johnsen and Colonius [20] have proposed an empirical analytical profile for the localized maximum wall pressure as function of the circular off-center distance along the wall

$$p_w \left(\frac{r}{R_0} \right) = \frac{c_1}{\sqrt{H_c^2 + (r/R_0)^2}} + c_2. \quad (24)$$

Here, c_1 and c_2 are two constants, which are determined from two arbitrary pressure profiles of the temporal evolution, and H_c is the stand-off distance at collapse time. For the given case, $H_c = 1.5$, $c_1 = 0.34$, and $c_2 = 0.025$. In Fig. 9, a series of radial wall-pressure profiles at several time instants is shown together with the asymptotical maximum wall-pressure function eq. (24). The time difference between two subsequent radial profiles is $20 \mu\text{s}$, and the first profile shows the pressure profile at $t = 2.80$ ms. Due to the radial spreading, the maximum pressure decreases in time. The overall maximum pressure at the wall infact occurs not at the centerline, but is shifted slightly outwards as the collapsing bubble shields the wall from the incoming pressure wave. This maximum pressure is slightly larger than the prediction of Johnsen and Colonius [20], which was already found earlier for this model configuration [1].

Table 2

Percentage of different operations on computational cost during each full Runge-Kutta cycle. We separate between the update of the time-step size, functions which are introduced for ALTS (*ALTS related*), and other functions which appear both during LTS and ALTS (*common operations*).

simulation	update Δt	ALTS related	common operations
1D (single fluid)	4.9 %	10.7 %	84.4 %
2D (multi-fluid)	2.1 %	5.2 %	92.7 %

4.4. Performance estimation

Finally, we analyze the additional cost due to computational operations which were introduced for the ALTS scheme: the projection and prediction of flux functions and the temporal integration of halo cells. We analyze the results for the one-dimensional single-fluid shock-tube simulation ($l_{max} = 7$) and the two-dimensional multi-fluid shock-bubble interaction ($l_{max} = 8$) presented in sections 4.2 and 4.3. We have performed our numerical tests on a homogeneous cluster of Intel Xeon E5-2697 v3 “Haswell” CPUs with 28 cores at 2.7 GHz and 2.3 GB memory. One-dimensional single-fluid simulations are executed sequentially, on one core only. Two-dimensional multi-fluid results are run in parallel on all 28 cores using 28 MPI ranks. Operations of a full Runge-Kutta cycle are split into three main categories: the update of the time-step size, additional operations that were introduced with the ALTS algorithm (*ALTS related*), and operations that are required for both LTS and ALTS schemes (*common operations*). We compute for each category its share of the compute time over several macro time steps. Results are given in Table 2.

The share of each category varies with the simulation type: the time step size update and ALTS-related operations account for approximately 16% of computational cost in the one-dimensional single-phase simulation, and for approximately 7% in the two-dimensional multi-phase simulation. We conclude that the additional cost has no significant adverse impact on the overall computing efficiency.

5. Conclusion and outlook

We have presented an adaptive local time-stepping scheme for a block-based multiresolution algorithm. The key improvement of the new approach is the projection and prediction of the right-hand-side terms of the governing equations and additional halo-cell time integration. The benefit of the improved time-stepping scheme is full adaptivity of the numerical time-step size down to blocks with highest resolution and, thus, improved stability for larger CFL numbers.

The ALTS scheme is applied to various one- and two-dimensional examples with strong velocity gradients to demonstrate its advantages. Even for large CFL numbers up to $CFL = 1.0$, our approach allows for stable and accurate simulations. The scheme is not limited to single-phase simulations, but can also be used for multiple phases without constraints on, e.g., the variation of the wave speed across the material interface.

Despite the fact that additional operations are required during each iteration (projection and prediction of the numerical divergence of the flux function, time integration of halo cells), the additional work of the ALTS scheme in comparison to the LTS scheme is negligible for multi-dimensional realistic flow problems. The numerical stability is improved as compared to the classical LTS scheme. Overall, the stability behavior of the constant time-stepping scheme is recovered, which was lost with the introduction of the LTS scheme.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The lead author has received funding from German Research Foundation (Deutsche Forschungsgesellschaft, project No. 277161739). The third and fourth author have received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 667483). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. for funding this project by providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre.

References

- [1] S. Adami, J. Kaiser, N.A. Adams, I. Bermejo-Moreno, Numerical modeling of shock waves in biomedicine, in: Center for Turbulence Research Proceedings of the Summer Program, 2016.
- [2] M. Bendahmane, R. Bürger, R.R. Baier, K. Schneider, Adaptive multiresolution schemes with local time stepping for two-dimensional degenerate reaction-diffusion systems, *Appl. Numer. Math.* 57 (7) (2009) 1668–1692.
- [3] B.L. Bihari, A. Harten, Multiresolution schemes for the numerical solution of 2-D conservation laws I, *J. Sci. Comput.* 18 (2) (1997) 315–354.

- [4] A. Cohen, S. Kaber, S. Müller, M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws, *Math. Comput.* 72 (241) (2001) 183–225.
- [5] F. Coquel, Q.L. Nguyen, M. Postel, Q.H. Tran, Local time stepping applied to implicit-explicit methods for hyperbolic systems, *Multiscale Model. Simul.* 8 (2) (2010) 540–570.
- [6] R. Deiterding, M.O. Domingues, S.M. Gomes, K. Schneider, Comparison of adaptive multiresolution and adaptive mesh refinement applied to simulations of the compressible Euler equations, *J. Sci. Comput.* 38 (5) (2016) 173–193.
- [7] M.O. Domingues, S.M. Gomes, O. Roussel, K. Schneider, An adaptive multiresolution scheme with local time stepping for evolutionary PDEs, *J. Comput. Phys.* 227 (8) (2008) 3758–3780.
- [8] M.O. Domingues, S.M. Gomes, O. Roussel, K. Schneider, Space-time adaptive multiresolution methods for hyperbolic conservation laws: applications to compressible Euler equations, *Appl. Numer. Math.* 59 (9) (2009) 2303–2321.
- [9] M.O. Domingues, O. Roussel, K. Schneider, An adaptive multiresolution method for parabolic PDEs with time-step control, *Int. J. Numer. Methods Eng.* 78 (2009) 652–670.
- [10] A.K.F. Gomes, M.O. Domingues, K. Schneider, O. Mendes, R. Deiterding, An adaptive multiresolution method for ideal magnetohydrodynamics using divergence cleaning with parabolic-hyperbolic correction, *Appl. Numer. Math.* 95 (2015) 199–213.
- [11] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, *Math. Comput.* 67 (221) (1998) 73–85.
- [12] L.H. Han, X.Y. Hu, N.A. Adams, Adaptive multi-resolution method for compressible multi-phase flows with sharp interface model and pyramid data structure, *J. Comput. Phys.* 262 (2014) 131–152.
- [13] L.H. Han, T. Indinger, X.Y. Hu, N.A. Adams, Wavelet-based adaptive multi-resolution solver on heterogeneous parallel architecture for computational fluid dynamics, *Comput. Sci. Res. Dev.* 26 (2011) 197–203.
- [14] A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* 135 (1983) 260–278.
- [15] A. Harten, Adaptive multiresolution schemes for shock computations, *J. Comput. Phys.* 115 (2) (1994) 319–338.
- [16] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, *Commun. Pure Appl. Math.* 48 (1995) 1305–1342.
- [17] N.A. Hawker, Y. Ventikos, Interaction of a strong shockwave with a gas bubble in a liquid medium: a numerical study, *J. Fluid Mech.* 701 (2012) 59–97.
- [18] B. Hejazialhosseini, D. Rossinelli, M. Bergdorf, P. Koumoutsakos, High order finite volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions, *J. Comput. Phys.* 229 (22) (2010) 8364–8383.
- [19] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202–228.
- [20] E. Johnsen, T. Colonius, Numerical simulations of non-spherical bubble collapse, *J. Fluid Mech.* 629 (2009) 231.
- [21] K. Kobayashi, T. Kodama, H. Takahira, Shock wave–bubble interaction near soft and rigid boundaries during lithotripsy: numerical analysis by the improved ghost fluid method, *Phys. Med. Biol.* 56 (19) (2011) 6421–6440.
- [22] M.M. Lopes, M.O. Domingues, K. Schneider, O. Mendes, Local time-stepping for adaptive multiresolution using natural extension of Runge–Kutta methods, *J. Comput. Phys.* 382 (2019) 291–318.
- [23] R. Menikoff, B.J. Plohr, The Riemann problem for fluid flows of real materials, *Rev. Mod. Phys.* 61 (1) (1989) 75–130.
- [24] S. Müller, Y. Stiriba, Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping, *J. Sci. Comput.* 30 (3) (2007) 493–531.
- [25] S. Osher, R. Sanders, Numerical approximations to nonlinear conservation laws with locally varying time and space grids, *Math. Comput.* 41 (164) (1983) 321–336.
- [26] D. Rossinelli, B. Hejazialhosseini, P. Hadjidoukas, C. Bekas, A. Curioni, A. Bertsch, S. Futral, S.J. Schmidt, N.A. Adams, P. Koumoutsakos, 11 pflop/s simulations of cloud cavitation collapse, in: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC '13 3*, ACM, New York, NY, USA, 2013, pp. 1–13.
- [27] D. Rossinelli, B. Hejazialhosseini, D.G. Spampinato, P. Koumoutsakos, Multicore/multi-gpu accelerated simulations of multiphase compressible flows using wavelet adapted grids, *J. Sci. Comput.* 33 (2) (2011) 512–540.
- [28] O. Roussel, K. Schneider, Adaptive multiresolution computations applied to detonations, *Z. Phys. Chem.* 229 (6) (2015) 931–953.
- [29] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic PDEs, *J. Comput. Phys.* 188 (2003) 493–523.
- [30] G.A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys.* 27 (1) (1978) 1–31.
- [31] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1) (1984) 115–173.