Journal of Scientific Computing (2023) 94:54
https://doi.org/10.1007/s10915-023-02106-8

The Cartesian Grid Active Flux Method with Adaptive Mesh Refinement

Donna Calhoun[1] · Erik Chudzik[2] · Christiane Helzel[2]

Received: 2 April 2022 / Revised: 29 November 2022 / Accepted: 27 December 2022 /
Published online: 27 January 2023

Abstract
We present the first implementation of the Active Flux method on adaptively refined Cartesian grids. The Active Flux method is a third order accurate finite volume method for hyperbolic conservation laws, which is based on the use of point values as well as cell average values of the conserved quantities. The resulting method has a compact stencil in space and time and good stability properties. The method is implemented as a new solver in ForestClaw, a software for parallel adaptive mesh refinement of patch-based solvers. On each Cartesian grid patch the single grid Active Flux method can be applied. The exchange of data between grid patches is organised via ghost cells. The local stencil in space and time and the availability of the point values that are used for the reconstruction, leads to an efficient implementation. The resulting method is third order accurate, conservative and allows the use of subcycling in time.

Keywords Cartesian grid active flux method · Hyperbolic conservation laws · Adaptive mesh refinement

Mathematics subject classification 65M08, 65M25, 65M50

1 Introduction

The Active Flux method is a finite volume method for hyperbolic conservation laws that was previously introduced by Eymann & Roe and Roe and coauthors [15–17, 23, 24, 26] . In its original form the method is third order accurate. This is achieved by using a continuous,

✉ Christiane Helzel
Christiane.Helzel@hhu.de

Donna Calhoun
donnacalhoun@boisestate.edu

Erik Chudzik
Erik.Chudzik@hhu.de

[1] Department of Mathematics, Boise State University, Boise, ID, USA

[2] Institute of Mathematics, Heinrich-Heine-University Düsseldorf, Düsseldorf, Germany

piecewise quadratic reconstruction and a sufficiently accurate quadrature rule to compute numerical fluxes. The quadrature method, i.e. typically Simpson's rule, requires point values of the conserved quantities at grid cell interfaces at the current time as well as at later time levels. These point values together with the cell average value are also used to compute the reconstruction. While classical finite volume methods only use cell average values of the conserved quantities as degrees of freedom, the Active Flux method involves both point values and cell average values degrees of freedom. This adds flexibility to the numerical method. For linear advection and the acoustic equations in one, two or three spatial dimensions, these point values can be updated using exact evolution formulas, making the method truly multidimensional.

Originally, Roe and Eymann [17] used unstructured triangular grids for their two-dimensional Active Flux method. In [5] and [19], two-dimensional Cartesian grid versions of the Active Flux method were introduced. The Cartesian grid method will also be used in this paper and briefly reviewed in the next section.

While the order of convergence of a numerical scheme is a property that can be shown in the limit when the mesh width and the time step goes to zero, for practical computations it is desirable to obtain accurate results on relatively coarse grids. Roe [24, 25] argues that the accurate approximation on coarse grids is strongly influenced by the computational stencil and that exact evolution operators perform well in this respect. Barsukow showed that the two-dimensional Cartesian grid Active Flux method for the acoustic equations with an exact evolution operator as described in [2, 18] is stationary preserving. This means that the numerical scheme does not add dissipation to discrete representations of all stationary states of the acoustic equations. As a consequence such states can be computed with very high accuracy on coarse grids, while other methods would require a higher resolution.

In earlier related work, Lukáčová-Medvid'ová et al. [21, 22], used multidimensional evolution operators as building blocks of finite volume methods of various order. Those methods differ from the Active Flux method mainly in the choice of the degrees of freedom. There is also recent work in progress on applying the Active Flux method to nonlinear hyperbolic problems [3, 19] and to hyperbolic problems with source term [4].

Although the Active Flux method performs well on coarse grids, for practical applications it might still be desirable to vary the size of the grid cells adaptively in order to allow a higher resolution in parts of the computational domain as needed. A possible application that could benefit from local refinement is the propagation of a high frequency acoustic wave.

In this paper we show how the Active Flux method can be applied on adaptively refined Cartesian grids. The local stencil of the method allows an efficient transfer of data between the different grid patches. Our numerical results confirm third order accuracy of the resulting method.The method is implemented as a new solver in ForestClaw [9, 11, 12].

This paper is organised as follows. In Sect. 2 we briefly review the Active Flux method for two-dimensional Cartesian grids. Section 3 describes the extension to adaptively refined grids. In Sect. 4 we introduce new Active Flux methods for advective transport and illustrate the performance of the adaptively refined Active Flux method for a variety of test problems.

2 The Cartesian Grid Active Flux Method

In this section we provide a brief review of the Active Flux method on a single two-dimensional Cartesian grid. More details can be found in Barsukow et al. [5] and Helzel et al. [19].

We consider hyperbolic conservation laws in divergence form

$$\partial_t q + \partial_x f(q) + \partial_y g(q) = 0,$$

where $q : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}^s$ is a vector of conserved quantities and $f, g : \mathbb{R}^s \to \mathbb{R}^s$ are vector valued flux functions. On a single patch we use a two-dimensional Cartesian grid with equidistant mesh sizes Δx and Δy. The grid cell (i, j) is described by $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \subset \mathbb{R}^2,\ i, j, \in \mathbb{Z}$. As a finite volume method, the Active Flux method computes cell averaged values of the conserved quantities via an update of the form

$$Q^{n+1}_{i,j} = Q^n_{i,j} - \frac{\Delta t}{\Delta x}\left(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}\right) - \frac{\Delta t}{\Delta y}\left(G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}\right), \tag{1}$$

where $Q^n_{i,j}$ is an approximation of the cell average values of the conserved quantities in grid cell (i, j) at time t_n and $F_{i\pm\frac{1}{2},j}$, $G_{i,j\pm\frac{1}{2}}$ are numerical fluxes at the grid cell interfaces given by

$$F_{i+\frac{1}{2},j} \approx \frac{1}{\Delta t \Delta y}\int_{t_n}^{t_{n+1}}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(q(x_{i+\frac{1}{2}}, y, t))dy\,dt$$

$$G_{i,j+\frac{1}{2}} \approx \frac{1}{\Delta t \Delta x}\int_{t_n}^{t_{n+1}}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(q(x, y_{j+\frac{1}{2}}, t))dx\,dt.$$

As suggested by Eymann and Roe, we use Simpson's rule to compute the numerical fluxes. For fluxes $F_{i+\frac{1}{2},j}$, this leads to the formula

$$\begin{aligned} F_{i+\frac{1}{2},j} := \frac{1}{36}\Big(& f(Q^n_{i+\frac{1}{2},j-\frac{1}{2}}) + 4f(Q^n_{i+\frac{1}{2},j}) + f(Q^n_{i+\frac{1}{2},j+\frac{1}{2}}) \\ & + 4f(Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}}) + 16f(Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j}) + 4f(Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}) \\ & + f(Q^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}}) + 4f(Q^{n+1}_{i+\frac{1}{2},j}) + f(Q^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}})\Big). \end{aligned} \tag{2}$$

We use an analogous formula for the flux $G_{i,j+\frac{1}{2}}$. The Q values in the right hand side of (2) are approximations to point values of the conserved quantities at the grid cell interface at times t_n, $t_{n+\frac{1}{2}}$ and t_{n+1}. To compute these point values, we assume that at time t_n the average values $Q^n_{i,j}$ and point values at cell corners $Q^n_{i\pm\frac{1}{2},j\pm\frac{1}{2}}$ and edge midpoints $Q^n_{i\pm\frac{1}{2},j}$ and $Q^n_{i,j\pm\frac{1}{2}}$ are known. The locations of these point values are shown in Fig. 1.

Based on these known values, a two-dimensional quadratic polynomial can be reconstructed in each grid cell. For simplicity, all Cartesian grid cells are mapped to a reference cell $[-1, 1] \times [-1, 1]$. In each cell, the reconstruction has the form

$$\begin{aligned} q^{rec}(\xi, \eta) = {} & c_{00} + c_{10}\xi + c_{01}\eta + c_{20}\xi^2 + c_{11}\xi\eta + c_{02}\eta^2 \\ & + c_{21}\xi^2\eta + c_{12}\xi\eta^2 + c_{22}\xi^2\eta^2 \end{aligned} \tag{3}$$

with $c_{ij} \in \mathbb{R},\ i, j = 0, 1, 2$ and $(\xi, \eta) \in [-1, 1] \times [-1, 1]$. The precise form of the reconstructed function is obtained by interpolating the eight known point values along the grid cell boundary and by requiring that the average of the reconstructed function agrees in each grid cell with the known cell average. This can conveniently be obtained by using appropriate basis functions as introduced in [19].

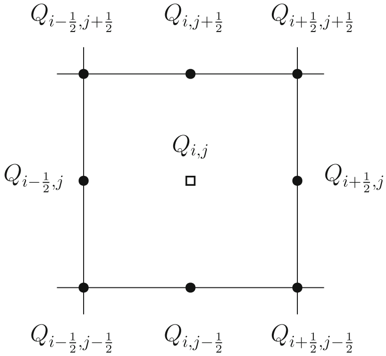
$Q_{i-\frac{1}{2},j+\frac{1}{2}}$
$Q_{i,j+\frac{1}{2}}$
$Q_{i+\frac{1}{2},j+\frac{1}{2}}$
$Q_{i,j}$
$Q_{i-\frac{1}{2},j}$
$Q_{i+\frac{1}{2},j}$
$Q_{i-\frac{1}{2},j-\frac{1}{2}}$
$Q_{i,j-\frac{1}{2}}$
$Q_{i+\frac{1}{2},j-\frac{1}{2}}$

Fig. 1 Degrees of freedom used to reconstruct the piecewise quadratic function in grid cell (i, j). Point values are marked by a dot and the cell average is marked by a square

We denote the reconstructed, continuous, piecewise polynomial function at time t_n by $q^n(x, y)$. A reconstructed polynomial in grid cell (i, j) is denoted $q_{ij}(x, y)$.

For special linear partial differential equations, the required point values of the conserved quantities can be calculated using an exact evolution formula applied to the piecewise quadratic data. This is in particular the case for linear advection and acoustics. The use of the exact evolution formula leads to a truly multidimensional numerical method. While the point values at time $t_{n+\frac{1}{2}}$ are only used in order to compute the numerical fluxes, the point values at time t_{n+1} are also used to compute the continuous, piecewise quadratic reconstruction $q^{n+1}(x, y)$, which provides the initial data for the next time step.

Since the point values are located along the grid cell boundaries, they are used for the reconstruction in two or four grid cells. Thus, the Active Flux method for two-dimensional Cartesian grids has four degrees of freedom per grid cell: the cell average plus three point values along the grid cell boundary. In our implementation, we associate point values $Q_{i-\frac{1}{2},j}$, $Q_{i-\frac{1}{2},j-\frac{1}{2}}$, $Q_{i,j-\frac{1}{2}}$ and the average value $Q_{i,j}$ with grid cell (i, j).

2.1 Advection Equation

For the advection equation

$$\partial_t q + a\partial_x q + b\partial_y q = 0, \tag{4}$$

with $q : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$, $a, b \in \mathbb{R}$, the exact evolution formula has the form

$$q(x, y, t) = q(x - a\,t, y - b\,t, 0).$$

Thus, the required point values at time $t_{n+\frac{1}{2}}$ and t_{n+1} can easily be computed by evaluating the continuous, piecewise quadratic function q^n at the appropriate upwind points, for example

$$\begin{aligned} Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}} &= q^n\left(x_{i+\frac{1}{2}} - a\frac{\Delta t}{2}, y_{j-\frac{1}{2}} - b\frac{\Delta t}{2}\right) \\ Q^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} &= q^n\left(x_{i+\frac{1}{2}} - a\Delta t, y_{j-\frac{1}{2}} - b\Delta t\right). \end{aligned} \tag{5}$$

In the numerical method, we want to restrict the time steps, so that waves propagate at most one grid cell per time step. This can be expressed in the form

$$\text{CFL} := \max\left(\frac{|a|\Delta t}{\Delta x}, \frac{|b|\Delta t}{\Delta y}\right) \le 1. \tag{6}$$

We have shown in [13] that linear stability requires a more severe time step restriction. In practical computations we therefore use time steps which satisfy CFL ≤ 0.75.

In Sect. 4 the approach is extended to advective transport in a spatially and temporally varying velocity field.

2.2 Burgers' Equation

For the two-dimensional Burgers' equation

$$q_t + \left(\frac{1}{2}q^2\right)_x + \left(\frac{1}{2}q^2\right)_y = 0 \tag{7}$$

with $q : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$, we do not have an exact evolution formula. Instead we approximate the point values using an approximative approach described in [13]. See also [3], where a similar approach was proposed for general scalar nonlinear hyperbolic problems.

For smooth solutions, equation (7) can equivalently be written in the advective form

$$q_t + qq_x + qq_y = 0, \tag{8}$$

which suggests the implicitly defined evolution formula

$$q(x, y, t) = q(x - q(x, y, t)t, y - q(x, y, t)t, 0). \tag{9}$$

Starting with an initial guess

$$\left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right)^0, \quad \left(Q^{n+1}_{i-\frac{1}{2},j}\right)^0,$$

we iteratively compute

$$\begin{aligned} \left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right)^\ell &= q^n\left(x_{i-\frac{1}{2}} - \left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right)^{\ell-1}\frac{\Delta t}{2}, y_j - \left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right)^{\ell-1}\frac{\Delta t}{2}\right) \\ \left(Q^{n+1}_{i-\frac{1}{2},j}\right)^\ell &= q^n\left(x_{i-\frac{1}{2}} - \left(Q^{n+1}_{i-\frac{1}{2},j}\right)^{\ell-1}\Delta t, y_j - \left(Q^{n+1}_{i-\frac{1}{2},j}\right)^{\ell-1}\Delta t\right) \quad \ell = 1, 2, \ldots. \end{aligned} \tag{10}$$

Wave speeds at other positions are computed analogously. Each iteration improves the accuracy by one order. This was shown in [13] using Taylor series expansion. We start the iteration with a first order accurate approximation. The piecewise quadratic reconstruction limits the achievable accuracy to third order. Thus, a third order accurate approximation can be achieved after two iterations.

An obvious initial guess would be to use the point values at the respective location, i.e.

$$\left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right)^0 = \left(Q^{n+1}_{i-\frac{1}{2},j}\right)^0 = Q^n_{i-\frac{1}{2},j}$$

and analogously for all other point values along the grid cell boundaries. These wave speeds are third order accurate in space and first order accurate in time. However, this choice suffers from an instability, if the characteristic speed changes sign as explained in detail in [19]. The instability can be mitigated if data from all adjacent grid cells are used to compute the initial guess. This leads to a stronger coupling of wave speeds and cell average values. Here, we compute the initial guess of the wave speed based on the neighbouring cell average values, i.e. we use

$$\left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right)^0 = \left(Q^{n+1}_{i-\frac{1}{2},j}\right)^0 = \frac{1}{2}\left(Q^n_{i-1,j} + Q^n_{i,j}\right)$$

$$\left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j-\frac{1}{2}}\right)^0 = \left(Q^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}}\right)^0 = \frac{1}{4}\left(Q^n_{i-1,j} + Q^n_{i,j} + Q^n_{i-1,j-1} + Q^n_{i,j-1}\right)$$

$$\left(Q^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\right)^0 = \left(Q^{n+1}_{i,j-\frac{1}{2}}\right)^0 = \frac{1}{2}\left(Q^n_{i,j} + Q^n_{i,j-1}\right).$$

The slight increase of the stencil, as introduced by this initial guess of the wave speeds, leads to a stable approximation.

2.3 Acoustics

The acoustic equations are given by

$$\begin{aligned} \partial_t p + c\nabla \cdot \boldsymbol{u} &= 0 \\ \partial_t \boldsymbol{u} + c\nabla p &= 0, \end{aligned} \tag{11}$$

where $\boldsymbol{u} : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}^2$ is the velocity vector, $p : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$ is the pressure and $c \in \mathbb{R}^+$ is the speed of sound.

The evolution formula for the two-dimensional acoustic equations, used for most problems in this paper, can be found in [17]. It is based on the observation that (11) can be rewritten as

$$\begin{aligned} \partial_{tt} p - c^2 \Delta p &= 0 \\ \partial_{tt} \boldsymbol{u} - c^2 \Delta \boldsymbol{u} &= c^2 \nabla \times \boldsymbol{w}, \end{aligned} \tag{12}$$

where $\boldsymbol{w} = \nabla \times \boldsymbol{u}$ is the vorticity and $\triangle$ is the Laplacian operator. For the two-dimensional cases considered in this paper, vorticity will have the form $(0, 0, v_x - u_y)^T$. Thus, in a flow with constant vorticity (i.e. irrotational flow) both pressure and velocity satisfy a wave equation. Furthermore, it is easy to verify that the vorticity is stationary, i.e. in the two-dimensional case the relation $\partial_t (v_x - u_y) = 0$ holds.

Using an exact representation of the solution of the wave equation $\partial_{tt}\phi = c^2 \Delta \phi$ in the form

$$\phi(\mathbf{x}, t) = t M_{ct}\{\partial_t \phi(\mathbf{x}, 0)\} + \partial_t \left(t M_{ct}\{\phi(\mathbf{x}, 0)\}\right), \tag{13}$$

that can be found for example in Courant and Hilbert [14], Eymann and Roe [17] proposed the evolution formulas

$$\begin{aligned} p(\mathbf{x}, t) &= M_R\{p\} + R\left(\partial_R M_R\{p\} - M_R\{\nabla \cdot \boldsymbol{u}\}\right) \\ \boldsymbol{u}(\mathbf{x}, t) &= M_R\{\boldsymbol{u}\} + R\left(\partial_R M_R\{\boldsymbol{u}\} - M_R\{\nabla p\}\right), \end{aligned} \tag{14}$$

where $R = c \cdot t$ and $M_R\{f\}$ is the spherical mean. The values of pressure and velocity at the right hand side of the evolution equation (14) are given initial values at time $t = 0$. For a scalar function $f : \mathbb{R}^2 \to \mathbb{R}$, the spherical mean over a disc with radius R, centred at (x, y) is defined by

$$M_R\{f\}(x, y) := \frac{1}{2\pi R} \int_0^{2\pi} \int_0^R f(x + s\cos\phi, y + s\sin\phi) \frac{s}{\sqrt{R^2 - s^2}} ds d\phi. \tag{15}$$

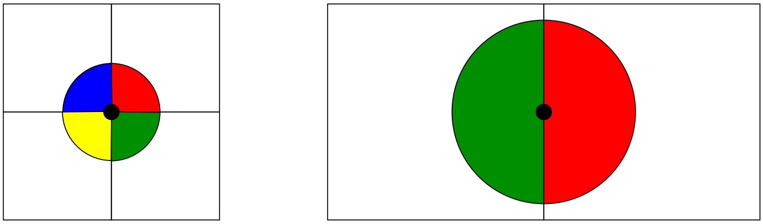

Fig. 2 Illustration of the update of point values for the acoustic equations on a Cartesian mesh

In the vector valued case, the formula is applied component wise. The solution formula can be evaluated exactly, if during each time step the previous values of pressure and velocity are replaced by the corresponding components of the reconstructed continuous, piecewise quadratic function q^n. In [17], this is explained in detail for triangular grids. In the above equations, we have replaced the time derivatives in (13) by derivatives with respect to R to obtain (14). For the Active Flux reconstruction on Cartesian grids those derivatives can be computed in the classical sense. Figure 2 illustrates the circular areas that contribute to the update of the point values at a corner of a grid cell as well as at an edge. By derivation, (14) is exact for initial values with constant vorticity.

Fan and Roe [18] derived an exact evolution formula for general initial values based on the observation that (13) can be rewritten as

$$\phi(\mathbf{x}, t) = \phi(\mathbf{x}, 0) + tM_{ct}\{\partial_t \phi(\mathbf{x}, 0)\} + \int_0^{ct} tM_{ct}\{\Delta\phi(\mathbf{x}, 0)\}dt. \tag{16}$$

Applying (16) to the acoustic equations (11) leads to

$$\begin{aligned} p(\mathbf{x}, t) &= p(\mathbf{x}, 0) - ctM_{ct}\{\nabla \cdot \mathbf{u}\} + \int_0^{ct} tM_{ct}\{\Delta p\}dt \\ \mathbf{u}(\mathbf{x}, t) &= \mathbf{u}(\mathbf{x}, 0) - ctM_{ct}\{\nabla p\} + \int_0^{ct} tM_{ct}\{\Delta \mathbf{u}\}dt. \end{aligned} \tag{17}$$

Using Helmholtz decomposition, Fan and Roe argued that this is indeed an exact evolution formula for the acoustic equations. The pressure and the curl free component of the velocity satisfy a wave equation and thus (17). The divergence free component is constant in time and therefore correctly contained in the term $\mathbf{u}(\mathbf{x}, 0)$.

While the derivation of Fan and Roe assumed sufficiently smooth data such that all derivatives can be computed in the classical sense, Barsukow [1] showed that the evolution formula is also valid more generally if interpreted in the distributional sense. If the solution operator is applied to the continuous, piecewise quadratic Active Flux reconstruction, the evaluation of the Laplacian introduces Dirac delta functions along grid cell interfaces. Barsukow proposes integration in the radial direction, thus avoiding these delta function singularities at grid cell interfaces. The resulting formulas are given in equations (2.16) and (2.17) of [5]. If applied to the Active Flux reconstruction, differentiation across grid cell interfaces is avoided and all terms can be evaluated in the classical sense.

Each time step of the explicit Active Flux method is restricted so that the circle around the edge midpoint over which the integration takes place remains inside the two adjacent grid cells, as shown in Fig. 2. We want to restrict the time step so that waves propagate at most

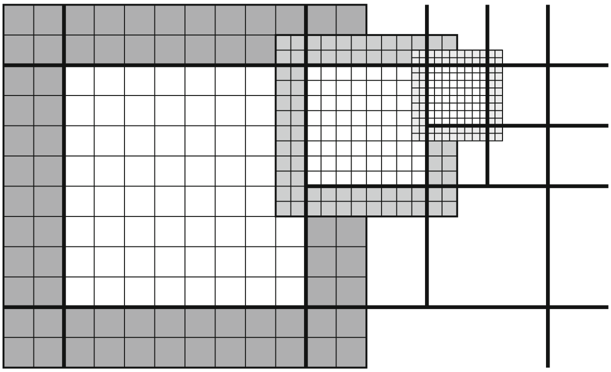

Fig. 3 Three refinement levels of quadrants of a ForestClaw mesh are shown. Each quadrant is occupied by a logically Cartesian grid of fixed resolution (e.g. 8 × 8 in the figure). Each grid also contains a layer of ghost cells (shaded region) which are used to facilitate the exchange of data between adjacent grid patches and between local and remote processors

half a grid cell per time step. This condition will be met if

$$\max\left(\frac{c\Delta t}{\Delta x}, \frac{c\Delta t}{\Delta y}\right) \leq \frac{1}{2}. \tag{18}$$

Larger time steps would make the implementation of the update of the point value indicated in the right of Fig. 2 more complicated and have therefore not been considered. In [13] we showed that this necessary condition is sufficient for linear stability of the Cartesian grid Active Flux method with exact evolution.

3 Adaptive Mesh Refinement for the Active Flux Method

We now describe the implementation of the adaptive Active Flux method as a new solver in ForestClaw [9], a software for parallel adaptive mesh refinement based on a quadtree approach. In ForestClaw, Cartesian grid patches occupy quadrants in a quadtree, or multiblock forest of quadtrees. ForestClaw was developed by Calhoun and Burstedde based on the p4est software [10].

Mesh refinement is realised by a bisection of grid patches so that a quadrant of resolution level ℓ is replaced by four quadrants of resolution level $\ell + 1$. A patch of level zero would correspond to a single Cartesian grid discretizing a single, square domain. The number of grid cells on a single quadrant is constant for all levels, resulting in a 2:1 refinement ratio between resolution levels. Typically 8 × 8, 16 × 16 or 32 × 32 grid cells are used on a single patch, which allows a flexible change of the resolution. Furthermore, single grid patches can efficiently be handled by separate processors in a parallel computation. Figure 3 shows a typical situation.

A ForestClaw mesh inherits many properties from the underlying p4est mesh [10]. In particular, a ForestClaw mesh is *well-balanced* so that adjacent quadrants never differ by more than one level. This is a necessary condition for the definition of data transfer between different patches, which is organized using ghost cells and described in more detail below. Another key feature of the ForestClaw mesh is that it is dynamically adapted to follow solution features of interest. At each time step, we apply coarsening and refinement criteria to the solution on each quadrant. If the coarsening criteria is satisfied by the solution in each quadrant in a family of four quadrants, the four quadrants will be replaced by a single quadrant, and the solution will be averaged from the finer grids to the new coarser solution. Otherwise, if the solution on a quadrant satisfies the refinement criteria, the quadrant will be subdivided into four quadrants, and the solution will be interpolated from the coarse parent to the new finer solution on each of the child quadrants. The details of the interpolation and averaging are provided next.

3.1 Spatial Transfer of Grid Cell Information

Solution data in a composite ForestClaw mesh needs to be communicated between adjacent grids sharing quadrant boundaries, and when dynamically coarsening and refining the mesh. The Active Flux method allows a very efficient transfer of both pointwise and cell-average information. In the following, a "grid patch" is both the p4est quadrant and the solution data in the quadrant. The following three situations need to be considered:

1. A **transfer from a fine grid to a coarse grid** is needed if four grid patches at level $\ell + 1$ are coarsened to a single patch at level ℓ. The same approach is used for the computation of ghost cell values on a patch of level ℓ from a neighbouring patch of level $\ell + 1$.
2. A **transfer from a coarse grid to a fine grid** is needed if a patch of level ℓ is marked for refinement and four patches at level $\ell + 1$ need to be reconstructed. The same approach is used for the computation of ghost cell values for a patch at level $\ell + 1$ from a neighbouring patch of level ℓ.
3. For neighbouring grid patches of the same level the ghost cell information is simply copied from the neighbouring grid patches.

We will now discuss the first two approaches in more detail.

Transfer from Fine to Coarse Grids

The degrees of freedom of a coarse grid cell are computed from the degrees of freedom of four grid cells on the finer level as illustrated in Fig. 4. The cell average of the coarse cell is the average of the four cell average values on the fine grid. The point values are copied from the point values at the vertices of the fine grid cells.

Transfer from Coarse to Fine Grids

We use the degrees of freedom of the coarse grid cell to reconstruct a quadratic polynomial as described in Sect. 2. This polynomial can be evaluated at all the required point values along the edges of the fine grid cells. The cell average values of the four fine grid cells are computed using Simpson's rule. This requires the additional computation of four point values on the coarse grid cell at the positions of the centers of the fine grid cells. An illustration is shown in Fig. 5. Since Simpson's rule is exact for our reconstructed function the sum of the cell average values on the four fine grid cells agrees with the cell average on the coarse grid cell exactly. This is a necessary property in order to maintain conservation.

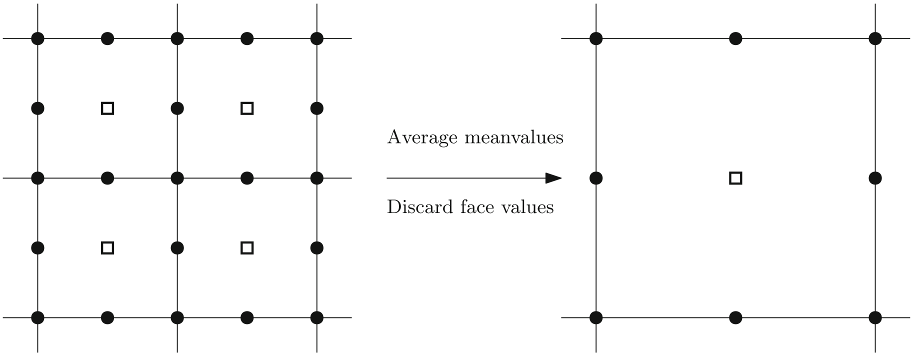

Fig. 4 Illustration of coarsening: The cell average on the coarse grid is the average of the four average values on the fine grid. The point values are copied from vertices of the fine grid cells

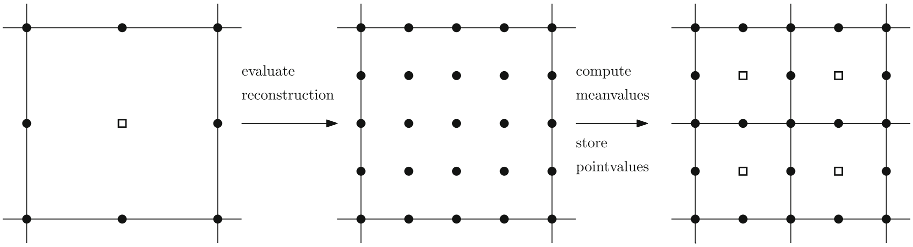

Fig. 5 Illustration of refinement: Point values and cell average values on the four fine grid cells are computed from the reconstruction of the conserved quantities at the coarse grid cell

3.2 Subcycling for a Local Refinement in Space and Time

In explicit finite volume methods for hyperbolic conservation laws, the time step is necessarily restricted by a CFL condition, which requires the numerical domain of dependence to contain the true domain of dependence of the partial differential equations [20].

The Active Flux method has a very compact stencil as explained in Sect. 2. Thus, for stability it is necessary to restrict the time step in such a way that information travels at most through one grid cell. Our results from [13] show that the time step should be restricted by CFL ≤ 0.75 for two-dimensional advection problems and by CFL ≤ 0.5 for the two-dimensional acoustic equations.

On an adaptively refined mesh, the smallest grid cells would typically dictate the time step restriction for the whole domain. To increase the efficiency of the computation, local time stepping (or "subcycling") can be used. In a subcycled computation, several time steps on more refined patches are taken for one time step on the coarsest grid. Subcycling was included in the original AMR algorithm by Berger and Oliger [8] and Berger and Colella [6] and is a standard feature of many AMR codes, including AMRClaw [7], AMReX [27] and many others. The local stencil of the Active Flux method allows for efficient implementation of subcycling, which will now be described in more detail.

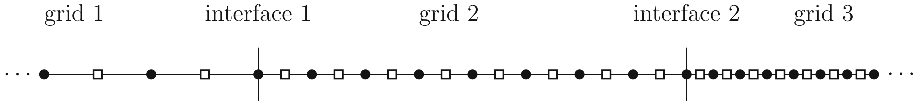

Fig. 6 Illustration of one-dimensional grid interfaces with different resolution

Subcycling

In order to use subcycling, we reconstruct the solution in two layers of ghost cells surrounding each patch.

We illustrate the idea for the one-dimensional case but an extension to the two-dimensional situation is straightforward. Assume a situation with three different grid patches as illustrated in Fig. 6. The coarsest grid, denoted as grid 1, has refinement level ℓ. The grid cells on this part of the domain have length Δx_ℓ. Our grid 2 has refinement level $\ell + 1$ and the grid cell length satisfies $\Delta x_{\ell+1} = \Delta x_\ell/2$. On the finest mesh, i.e. grid 3, we have $\Delta x_{\ell+2} = \Delta x_{\ell+1}/2$. For efficiency, we will use a time step Δt_ℓ on grid 1, two time steps $\Delta t_{\ell+1} = \Delta t_\ell/2$ on grid 2 and four time steps $\Delta t_{\ell+2} = \Delta t_{\ell+1}/2 = \Delta t_\ell/4$ on grid 3.

In the ForestClaw implementation, we start the time stepping on the finest level grids. Referring to grids 1,2 and 3 described above, the algorithm proceeds as follows for this three-level mesh configuration.

1. Advance the solution one step on grid 3 using time step $\Delta t_{\ell+2}$.
2. Recursively advance the solution one step on grid 2 (using time step $\Delta t_{\ell+1}$) and on grid 1 (using time step Δt_ℓ).
3. Advance the solution a second step on grid 3.
4. Grids 2 and 3 are now time synchronized and ghost cell data is exchanged between these levels.
5. Advance the solution a third step on grid 3.
6. Recursively update the solution a second step on level 2.
7. Advance the solution a fourth step on grid 3.

In each grid advance, the first layer of ghost cells is updated along with all interior cells. These ghost cell values are needed to update interior cells at the fine grid intermediate time level with no corresponding coarse grid time level. To simplify the subcycling algorithm, we update this first layer of ghost cells at both fine grid time steps, even though the ghost cell values from the second update are never used and will be replaced by data averaged or interpolated from the neighboring finer or coarser grids when coarse and fine grids are time synchronized. For those grids at the physical boundary, physical boundary conditions are used at all time levels. The time step on each grid uses a stable time step appropriate for that grid. These steps are illustrated in Fig. 7.

The key difference between the subcycling approach described above and the subcycling described in the original second order finite volume schemes described by Berger, Oliger and Colella is that in the Active Flux method, ghost cells at the intermediate time level are not filled using time interpolation from neighboring coarser grid cells. Instead we obtain all the required data directly from previously updated degrees of freedom of the Active Flux method.

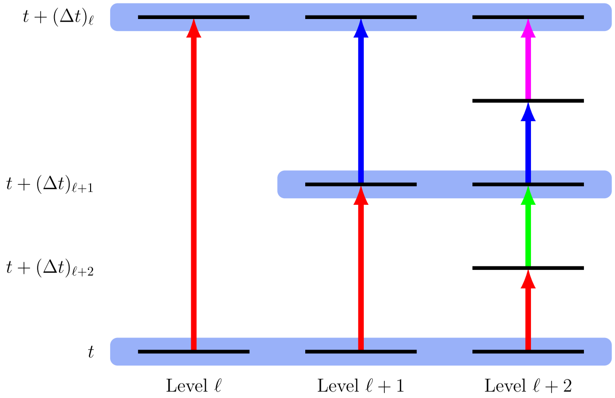

Fig. 7 Subcycling algorithm for advancing three levels (with grids 1, 2 and 3) a single coarse grid time step of size Δt_ℓ. The algorithm starts at the lower right by advancing one step on the finest level. All arrows of the same color are advanced recursively. The blue shaded boxes indicate time synchronized levels where ghost exchanges between neighboring grids takes place. Grids without time synchronized coarse grid neighbors advance by using updated values in an extra layer of ghost cells

Global Conservation Property

To ensure the global conservation property of the method, the fluxes at grid cell interfaces need to be defined in a unique way. At regular grid cell interfaces the Active Flux method automatically satisfies this conservation property. At interfaces of patches with different refinement level, conservation needs to be enforced by some kind of "conservative fix". We use the classical approach described by Berger and Colella [6], and update both the coarse and the fine grid cells using the fluxes that have been computed for the more resolved grid.

Exactness for Advective Transport

Remark 3.1 If the Cartesian grid Active Flux method with adaptive mesh refinement is used to solve the two-dimensional advection equation (4) with polynomial initial values of the form $q_0 \in \mathbb{P}_p$, $p \in \mathbb{N}$, then the numerical solution agrees with the exact solution for $p \leq 2$.

Proof As a consequence of the interpolation at the point value degrees of freedom, the continuous, piecewise quadratic reconstruction of the Active Flux method agrees with the initial values. The coarsening and the refinement steps, described in Sect. 3.1, are exact for all polynomials of degree two or less. Thus, the update of the point values using the exact evolution formula described in Sect. 2.1 is exact. The numerical fluxes agree with the exact fluxes because Simpson's rule is even exact for all polynomial of degree three. This leads to an exact computation of the new cell average values. □

Efficiency

In ForestClaw, the overhead costs associated with patch communication (ghost cell exchanges as well as parallel communication) and dynamic regridding is typically less than 10%, even for problems with inexpensive solvers (e.g. second order scalar advection update). Overhead costs are dominated by ghost cells exchanges and communication, which will increase

Table 1 Error at time $t = 1$ measured in the 1-norm and convergence rates for the advection problem with constant refinement along the diagonal (left table) and refinement that resolves the relevant solution structure (right table)

Level	Error	EOC	Level	Error	EOC
4	6.46×10^{-5}	–	5	8.45×10^{-6}	–
5	8.45×10^{-6}	2.9347	6	1.07×10^{-6}	2.9865
4-5	5.08×10^{-5}	–	3-5	8.47×10^{-6}	–
5-6	6.61×10^{-6}	2.9414	3-6	1.07×10^{-6}	2.9849

slightly in the Active flux method, since point values, in addition to cell averages must be communicated. However, the arithmetic intensity of the scheme also increases, and so we expect that AMR overhead with the Active Flux method will still be only a small fraction of the total costs of the simulation. Detailed performance characteristics will be investigated in future work.

4 Numerical Results

In this section we show numerical results for advection, advective transport with spatially and temporally varying velocity field, Burgers' equation and acoustics. We introduce new Active Flux methods for advective transport problems and discuss the preservation of constant states.

For all computations we use subcycling and the conservative fix described above unless otherwise noted. Each grid patch uses 16 × 16 Cartesian grid cells plus ghost cells. Our results confirm third order accuracy of the Active Flux method on adaptively refined grids.

4.1 Convergence Study for Advection

We consider the advection equation (4) on the domain $[0, 1] \times [0, 1]$ with initial condition

$$q(x, y, 0) = H(r(x, y) + r_0) - H(r(x, y) - r_0), \tag{19}$$

where $r(x, y) := \sqrt{(x - x_0)^2 + (y - y_0)^2}$ and $H(r) := (\tanh(r/0.02) + 1)/2$ with $r_0 = 0.15$ and $x_0 = y_0 = 0.5$. We use the advection speeds $a = 1$ and $b = 0.5$ and time steps which satisfy CFL $= 0.6$.

Two different adaptively refined grids are considered. In the first case refinement is allowed only along the diagonal of the domain as shown in Fig. 8 (left plot). In the second case a patch is refined, if $q_{max} - q_{min} > 0.001$. In this case the refined grid will follow the solution structure as shown in Fig. 8 (right plot). By comparing the numerical solution with the exact solution we can measure the error and compute the experimental order of convergence (EOC). The results are shown in Table 1 for refinement along the diagonal and for dynamic refinement which follows the solution structure.

For the academic test case with refinement along the diagonal, we see that the changes in the grid structure did not introduce any grid-induced artefacts. The accuracy observed on the adaptively refined grid is comparable with the accuracy on a regular Cartesian grid on the coarser level.

If the adaptive mesh follows the solution structure, then the accuracy obtained on the adaptively refined grid compares well with the accuracy obtained on a regular Cartesian grid that uses the highest level of refinement in the full domain. These results are shown in Table 1.

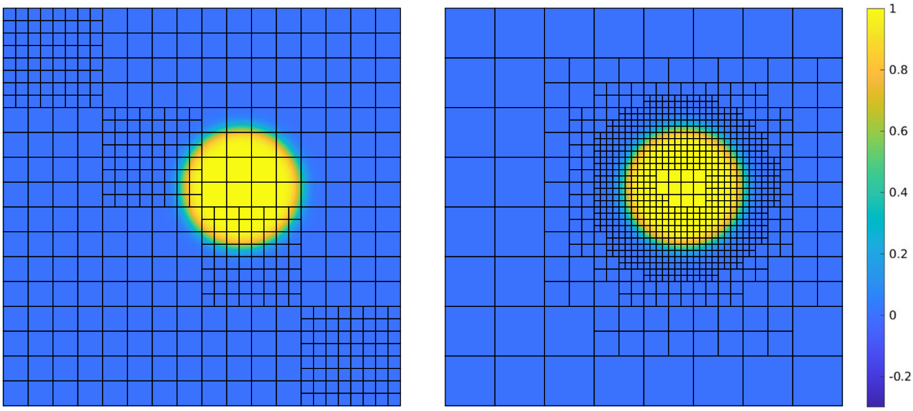

Fig. 8 Solutions to (19) with constant refinement along the diagonal (left) and with refinement level 3-6 (right)at time $t = 0.1$

The accuracy obtained on the adaptive mesh with levels $3-5$ or $3-6$ compares well with the accuracy obtained on the grids that are refined uniformly to levels 5 or 6.

4.2 Solid-Body Rotation

Next we consider the linear advection equation with variable coefficients

$$\partial_t q + \partial_x(a(x, y)q) + \partial_y(b(x, y)q) = 0 \tag{20}$$

on $[-1, 1] \times [-1, 1]$. We define the advection field for solid body rotation using the stream-function $\Psi(x, y) = \frac{\pi}{2}(x^2 + y^2)$ and obtain

$$\begin{aligned} a(x, y) &= -\partial_y \Psi(x, y) = -\pi y, \\ b(x, y) &= \partial_x \Psi(x, y) = \pi x. \end{aligned} \tag{21}$$

The velocity field is divergence free, thus Eq. (20) is equivalent to

$$\partial_t q + a(x, y)\partial_x q + b(x, y)\partial_y q = 0. \tag{22}$$

Furthermore, the solution at time $t = 2n$, $n \in \mathbb{N}$ agrees with the initial values since the flow has simply made n complete rotations. The characteristics satisfy the ordinary differential equation

$$\begin{aligned} x'(t) &= a(x(t), y(t)) \\ y'(t) &= b(x(t), y(t)) \end{aligned} \tag{23}$$

with initial values $x(0) = x_0$, $y(0) = y_0$. We are interested in $(x(-\tau), y(-\tau))$ with $\tau = \Delta t/2, \Delta t$. Those values can be computed using the analytical solution

$$\begin{aligned} x(t) &= -y_0 \sin(\pi t) + x_0 \cos(\pi t) \\ y(t) &= y_0 \cos(\pi t) + x_0 \sin(\pi t). \end{aligned} \tag{24}$$

This allows us to evaluate the conserved quantities at all required nodes of Simpson's quadrature formula by tracing back the characteristics. For the flux $F_{i+\frac{1}{2},j}$, we use Eq. (2) with, for

Table 2 Error at time $t = 2$ measured in the 1-norm and EOC for solid-body rotation using the approach of Sect. 4.2

Level	Error	EOC
6	4.60×10^{-6}	–
7	5.83×10^{-7}	2.9807
3-6	4.67×10^{-6}	–
3-7	5.92×10^{-7}	2.9801

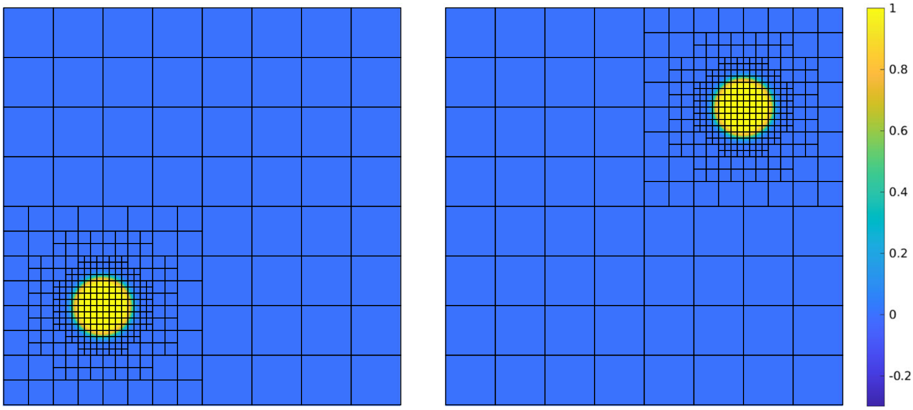

Fig. 9 Solution to (22) with initial condition (19) after a half (left) and a full rotation (right)

example,

$$
\begin{aligned}
f(Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j}) &= a(x_{i+\frac{1}{2}}, y_j)Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = -\pi y_j Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j} \\
g(Q^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}) &= b(x_i, y_{j+\frac{1}{2}})Q^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = \pi x_i Q^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}
\end{aligned}
\tag{25}
$$

and

$$
\begin{aligned}
Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j} &= q^n \left(-y_j \sin\left(-\pi \frac{\Delta t}{2}\right) + x_{i+\frac{1}{2}} \cos\left(-\pi \frac{\Delta t}{2}\right), y_j \cos\left(-\pi \frac{\Delta t}{2}\right)\right. \\
&\qquad \left. + x_{i+\frac{1}{2}} \sin\left(-\pi \frac{\Delta t}{2}\right)\right) \\
Q^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} &= q^n \left(-y_{j+\frac{1}{2}} \sin\left(-\pi \frac{\Delta t}{2}\right) + x_i \cos\left(-\pi \frac{\Delta t}{2}\right), y_{j+\frac{1}{2}} \cos\left(-\pi \frac{\Delta t}{2}\right)\right. \\
&\qquad \left. + x_i \sin\left(-\pi \frac{\Delta t}{2}\right)\right)
\end{aligned}
\tag{26}
$$

and analogously for all the other nodes.

We compare numerical solutions of (22) with initial condition (19) using refinement levels 3-6 and 3-7 after a half rotation with the exact solution and compute the error as well as the EOC. Results are shown in Table 2.

Again, the accuracy of the computations on the adaptively refined grids compares well with the accuracy obtained on the equidistant grids with highest resolution. Figure 9 shows the numerical solution with refinement level 3-6 after a half and a full rotation.

Remark 4.1 The Active Flux method for solid body rotation described in Sect. 4.2 preserves constant states on regular Cartesian grids.

Proof We consider constant data at time t_n, i.e. $q(x, y, t_n) = C \in \mathbb{R}$. Then all the point values at time t_n, $t_{n+\frac{1}{2}}$ and t_{n+1} are equal to C and the numerical fluxes have the form

$$\begin{aligned} F_{i+\frac{1}{2},j} &= -\frac{C\pi}{6}\left(y_{j-\frac{1}{2}} + 4y_j + y_{j+\frac{1}{2}}\right) \\ G_{i,j+\frac{1}{2}} &= \frac{C\pi}{6}\left(x_{i-\frac{1}{2}} + 4x_i + x_{i+\frac{1}{2}}\right). \end{aligned}$$

Thus, the fluxes in the x-direction only depend on y and the fluxes in the y-direction only depend on x. Consequently, all the flux differences cancel and the cell average values remain constant. □

4.3 Advective Transport in a Spatially and Temporally Varying Divergence Free Velocity Field Defined by a Stream Function

Now we consider advective transport in a velocity field that depends on space and time, i.e. we study an equation of the form

$$\partial_t q + \partial_x \left(a(x, y, t)q\right) + \partial_y \left(b(x, y, t)q\right) = 0. \tag{27}$$

The divergence free velocity field is again defined via a stream function which may now also vary in time. We use either the exact derivatives

$$\begin{aligned} a(x, y, t) &= -\frac{\partial}{\partial y}\Psi(x, y, t), \\ b(x, y, t) &= \frac{\partial}{\partial x}\Psi(x, y, t) \end{aligned} \tag{28}$$

or centered finite difference formulas

$$\begin{aligned} a(x, y, t) &= -\frac{\Psi(x, y + \Delta y/2, t) - \Psi(x, y - \Delta y/2, t)}{\Delta y}, \\ b(x, y, t) &= \frac{\Psi(x + \Delta x/2, y, t) - \Psi(x - \Delta x/2, y, t)}{\Delta x}. \end{aligned} \tag{29}$$

The point values of the conserved quantity q are computed at the intermediate and final time by tracing the characteristics back in time using the classical fourth order accurate Runge-Kutta method. The numerical fluxes are computed using Simpson's rule, i.e., to compute $F_{i+\frac{1}{2},j}$ we use (2) with for example $f(Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j}) = a(x_{i+\frac{1}{2}}, y_j, t_{n+\frac{1}{2}})Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$, where $Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$ is the point value of the conserved quantity, computed using characteristics and evaluating the reconstructed function at the tracing point, and $a(x_{i+\frac{1}{2}}, y_j, t_{n+\frac{1}{2}})$ is obtained by evaluating either (28) or (29).

4.3.1 The Swirl Flow Problem

For the swirl flow problem we consider the stream function

$$\Psi(x, y, t) = \frac{1}{\pi}\sin^2(\pi x) \cdot \sin^2(\pi y) \cdot \cos(\pi t).$$

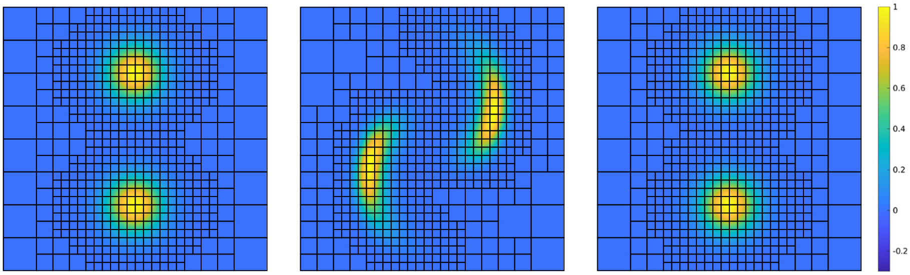

Fig. 10 Solution of the swirl problem with initial data of the form (30) at times $t = 0$ (left), $t = 0.5$ (middle) and $t = 1$ (right) for the swirl problem with smooth initial condition

Table 3 Error at time $t = 1$ measured in the 1-norm and EOC for the smooth swirl problem using subcycling (left) with finite differences, (right) with exact derivatives

Level	Error	EOC	Level	Error	EOC
4	5.5904×10^{-6}	–	4	5.5907×10^{-6}	–
5	7.0812×10^{-7}	2.9809	5	7.0813×10^{-7}	2.9809
3-4	5.5914×10^{-6}	–	3-4	5.5914×10^{-6}	–
3-5	7.0948×10^{-7}	2.9784	3-5	7.0920×10^{-7}	2.9789

To perform numerical convergence studies, we consider smooth initial values of the form

$$\begin{aligned} q(x, y, 0) = \exp(-100((x-0.5)^2 + (y-0.25)^2)) \\ + \exp(-100((x-0.5)^2 + (y-0.75)^2)) \end{aligned} \tag{30}$$

on the domain $[0, 1] \times [0, 1]$ with periodic boundary conditions. The solution at time $t = n$, $n = 1, 2, \ldots$ agrees with the initial data. In our computations, the velocity field was computed using either (28) or (29). For both cases we perform numerical convergence studies on uniform Cartesian grids as well as on adaptively refined grids with subcycling. Numerical results are shown in Fig. 10 and the results of a convergence study with and without subcycling are shown in Table 3. The results again confirm third order convergence. The use of subcycling did not significantly influence the accuracy nor does the use of the finite difference formula. Both approaches provide third order accurate approximations and comparable values of the error.

4.3.2 Exact Preservation of Constant States

In Remark 3.1 we have seen that for advection in a constant velocity field, constant, linear and quadratic functions are approximated exactly with the Active Flux method. This is no longer the case if we consider advective transport in a more general velocity field. We will now introduce a numerical flux which does preserve constant states. This new flux computation for $F_{i+\frac{1}{2},j}$ uses the approximation

$$
\begin{aligned}
&\frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} a(x_{i+\frac{1}{2}}, y, t) q(x_{i+\frac{1}{2}}, y, t) dy dt \\
&\quad \approx \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \left(a(x_{i+\frac{1}{2}}, y_j, t) \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} q(x_{i+\frac{1}{2}}, y, t) dy \right) dt
\end{aligned}
\tag{31}
$$

and analogously for the flux $G_{i,j+\frac{1}{2}}$. Using again Simpson's rule we obtain

$$
\begin{aligned}
F_{i+\frac{1}{2},j} = \frac{1}{6}\Bigg(&\frac{1}{6} a(x_{i+\frac{1}{2}}, y_j, t_n) \cdot \left(Q^n_{i+\frac{1}{2},j-\frac{1}{2}} + 4 Q^n_{i+\frac{1}{2},j} + Q^n_{i+\frac{1}{2},j+\frac{1}{2}} \right) \\
&+ \frac{4}{6} a(x_{i+\frac{1}{2}}, y_j, t_{n+\frac{1}{2}}) \cdot \left(Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}} + 4 Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j} + Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} \right) \\
&+ \frac{1}{6} a(x_{i+\frac{1}{2}}, y_j, t_{n+1}) \cdot \left(Q^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} + 4 Q^{n+1}_{i+\frac{1}{2},j} + Q^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} \right) \Bigg)
\end{aligned}
\tag{32}
$$

and

$$
\begin{aligned}
G_{i,j+\frac{1}{2}} = \frac{1}{6}\Bigg(&\frac{1}{6} b(x_i, y_{j+\frac{1}{2}}, t_n) \cdot \left(Q^n_{i-\frac{1}{2},j+\frac{1}{2}} + 4 Q^n_{i,j+\frac{1}{2}} + Q^n_{i+\frac{1}{2},j+\frac{1}{2}} \right) \\
&+ \frac{4}{6} b(x_i, y_{j+\frac{1}{2}}, t_{n+\frac{1}{2}}) \cdot \left(Q^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2}} + 4 Q^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} + Q^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} \right) \\
&+ \frac{1}{6} b(x_i, y_{j+\frac{1}{2}}, t_{n+1}) \cdot \left(Q^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}} + 4 Q^{n+1}_{i,j+\frac{1}{2}} + Q^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} \right) \Bigg).
\end{aligned}
\tag{33}
$$

In Appendix A we show that the use of the numerical fluxes (32), (33) leads to a second order accurate approximation of smooth solutions on Cartesian grids, if the divergence free velocity field is defined via a stream function and approximated using finite differences. A method which exactly preserves constant states might be of interest for the advective transport of piecewise constant data. In this case we can not expect high order accuracy even with a third order accurate method, since the data is not sufficiently smooth.

Theorem 4.2 *The Active Flux method with fluxes of the form* (32)*,* (33) *and* a, b *as defined in* (29)*, provides a method for* (27) *that preserves constant states on the uniform Cartesian mesh.*

Proof We consider constant data at time t_n, i.e. $q(x, y, t_n) = C \in \mathbb{R}$. Then all the point values at time t_n, $t_{n+\frac{1}{2}}$ and t_{n+1}, which are obtained by tracing back the characteristics, are also equal to C. The finite volume update now reduces to

$$
\begin{aligned}
Q^{n+1}_{i,j} = Q^n_{i,j} - \frac{\Delta t}{\Delta x} \frac{C}{6} \Big(&a(x_{i+\frac{1}{2}}, y_j, t_n) + 4a(x_{i+\frac{1}{2}}, y_j, t_{n+\frac{1}{2}}) + a(x_{i+\frac{1}{2}}, y_j, t_{n+1}) \\
&- a(x_{i-\frac{1}{2}}, y_j, t_n) - 4a(x_{i-\frac{1}{2}}, y_j, t_{n+\frac{1}{2}}) - a(x_{i-\frac{1}{2}}, y_j, t_{n+1}) \Big) \\
- \frac{\Delta t}{\Delta y} \frac{C}{6} \Big(&b(x_i, y_{j+\frac{1}{2}}, t_n) + 4b(x_i, y_{j+\frac{1}{2}}, t_{n+\frac{1}{2}}) + b(x_i, y_{j+\frac{1}{2}} t_{n+1}) \\
&- b(x_i, y_{j-\frac{1}{2}}, t_n) - 4b(x_i, y_{j-\frac{1}{2}}, t_{n+\frac{1}{2}}) - b(x_i, y_{j-\frac{1}{2}}, t_{n+1}) \Big).
\end{aligned}
$$

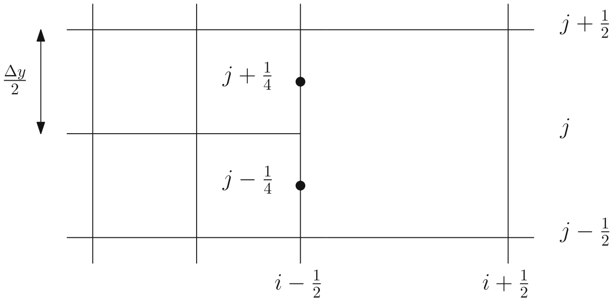

Fig. 11 Coarse grid cell with neighboring fine grid cells

For the terms at time level t_n we obtain

$$
\begin{aligned}
&-\frac{C\Delta t}{6\Delta x}\Big(a(x_{i+\frac{1}{2}}, y_j, t_n) - a(x_{i-\frac{1}{2}}, y_j, t_n)\Big) - \frac{C\Delta t}{6\Delta y}\Big(b(x_i, y_{j+\frac{1}{2}}, t_n) - b(x_i, y_{j-\frac{1}{2}}, t_n)\Big)\\
&\quad = -\frac{C\Delta t}{6\Delta x\Delta y}\Big(- \Psi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, t_n) + \Psi(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}, t_n)\\
&\qquad + \Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}, t_n) - \Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, t_n)\Big)\\
&\qquad - \frac{C\Delta t}{6\Delta y\Delta x}\Big(\Psi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, t_n) - \Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}, t_n)\\
&\qquad - \Psi(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}, t_n) + \Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}, t_n)\Big) = 0
\end{aligned}
$$

In the same way the terms at time $t_{n+\frac{1}{2}}$ and t_{n+1} cancel and we obtain $Q_{i,j}^{n+1} = Q_{i,j}^{n} = C$ for all i, j. □

Now we consider Cartesian grids with adaptive mesh refinement. Without loss of generality we restrict considerations to the situation shown in Fig. 11. We again consider constant data at time t_n, i.e. $q(x, y, t_n) = C$. By Theorem 4.2, then, constant states are preserved for both fine grid cell and coarse grid cell updates. For the coarse cell update, the left flux $F_{i-\frac{1}{2},j}$ is the sum of two fluxes used to update the two small cells, i.e.

$$
\begin{aligned}
F_{i-\frac{1}{2},j} &= \frac{1}{2}\Big(F_{i-\frac{1}{2},j-\frac{1}{4}} + F_{i-\frac{1}{2},j+\frac{1}{4}}\Big)\\
&= \frac{1}{2}C\Big[\frac{1}{6}\Big(a(x_{i-\frac{1}{2}}, y_{j-\frac{1}{4}}, t_n) + 4a(x_{i-\frac{1}{2}}, y_{j-\frac{1}{4}}, t_{n+\frac{1}{2}}) + a(x_{i-\frac{1}{2}}, y_{j-\frac{1}{4}}, t_{n+1})\Big)\\
&\qquad + \frac{1}{6}\Big(a(x_{i-\frac{1}{2}}, y_{j+\frac{1}{4}}, t_n) + 4a(x_{i-\frac{1}{2}}, y_{j+\frac{1}{4}}, t_{n+\frac{1}{2}}) + a(x_{i-\frac{1}{2}}, y_{j+\frac{1}{4}}, t_{n+1})\Big)\Big]\\
&= \frac{1}{2}C\frac{1}{6}\Big[-\frac{\Psi(x_{i-\frac{1}{2}}, y_j, t_n) - \Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, t_n)}{\Delta y/2}\\
&\qquad -4\frac{\Psi(x_{i-\frac{1}{2}}, y_j, t_{n+\frac{1}{2}}) - \Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, t_{n+\frac{1}{2}})}{\Delta y/2}\\
&\qquad -\frac{\Psi(x_{i-\frac{1}{2}}, y_j, t_{n+1}) - \Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, t_{n+1})}{\Delta y/2}\\
&\qquad -\frac{\Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}, t_n) - \Psi(x_{i-\frac{1}{2}}, y_j, t_n)}{\Delta y/2}
\end{aligned}
$$

$$
\begin{aligned}
&-4\frac{\Psi(x_{i-\frac{1}{2}},y_{j+\frac{1}{2}},t_{n+\frac{1}{2}})-\Psi(x_{i-\frac{1}{2}},y_j,t_{n+\frac{1}{2}})}{\Delta y/2}\\
&-\frac{\Psi(x_{i-\frac{1}{2}},y_{j+\frac{1}{2}},t_{n+1})-\Psi(x_{i-\frac{1}{2}},y_j,t_{n+1})}{\Delta y/2}\Bigg]\\
&=\frac{1}{6}C\Bigg[-\frac{\Psi(x_{i-\frac{1}{2}},y_{j+\frac{1}{2}},t_n)-\Psi(x_{i-\frac{1}{2}},y_{j-\frac{1}{2}},t_n)}{\Delta y}\\
&-4\frac{\Psi(x_{i-\frac{1}{2}},y_{j+\frac{1}{2}},t_{n+\frac{1}{2}})-\Psi(x_{i-\frac{1}{2}},y_{j-\frac{1}{2}},t_{n+\frac{1}{2}})}{\Delta y}\\
&-\frac{\Psi(x_{i-\frac{1}{2}},y_{j+\frac{1}{2}},t_{n+1})-\Psi(x_{i-\frac{1}{2}},y_{j-\frac{1}{2}},t_{n+1})}{\Delta y}\Bigg]\\
&=C\frac{1}{6}\left(a(x_{i-\frac{1}{2}},y_j,t_n)+4a(x_{i-\frac{1}{2}},y_j,t_{n+\frac{1}{2}})+a(x_{i-\frac{1}{2}},y_j,t_{n+1})\right).
\end{aligned}
$$

This is the same flux that is used on the uniform Cartesian grid. By Theorem 4.2, then, constant states are preserved for both fine grid cell and coarse grid cell updates. We summarize our result.

Corollary 4.3 *The Active Flux method described above preserves constant states on Cartesian grids with adaptive mesh refinement when used with global time stepping (e.g. no subcycling).*

With subcycling, the sum of the fluxes from the fine grid cells and that is used to update the coarse grid cell would contain components at the intermediate time that are in general not balanced by the remaining coarse grid fluxes.

To measure the error in the approximation of constant states by using subcycling, we compute the solution at time $t = 1$ using the velocity field of the swirl flow problem but with constant initial values that are equal to one in the whole domain. Refinement with levels $3-4$ is used along the diagonal as shown in Fig. 8 (left). With subcycling we observe an error of size 10^{-13}. Without subcycling the error is about 10^{-16}, i.e. agrees with the expected roundoff error. If we instead use the third order accurate Active Flux method, which does not preserve constant states exactly, then the error observed for this test case is of the order 10^{-11} if the velocity field is defined via (28) and 10^{-6} if the velocity field is defined via (29). This is independent of the use of subcycling.

Now we consider the swirl problem with piecewise constant initial values of the form

$$
q(x,y,0)=\begin{cases}1 & 0\le x\le 0.5\\ 0 & 0.5<x\le 1\end{cases}
$$

on the domain $[0,1]\times[0,1]$ with periodic boundaries on top and bottom and with zero-order extrapolation on the left and right boundary using refinement level 3-6. The reconstruction was limited using the bound preserving limiter described in [13]. Results are shown in Fig. 12. Subcycling was used although this slightly perturbs the exact preservation of constant states.

4.4 Burgers' Equation

It is straightforward to apply the adaptive Active Flux method to scalar nonlinear hyperbolic problems. We consider the Burgers' equation (7) on the domain $[0,1]\times[0,1]$ with initial values

$$
q(x,y,0)=\sin(2\pi x)\sin(2\pi y)+0.1. \tag{34}
$$

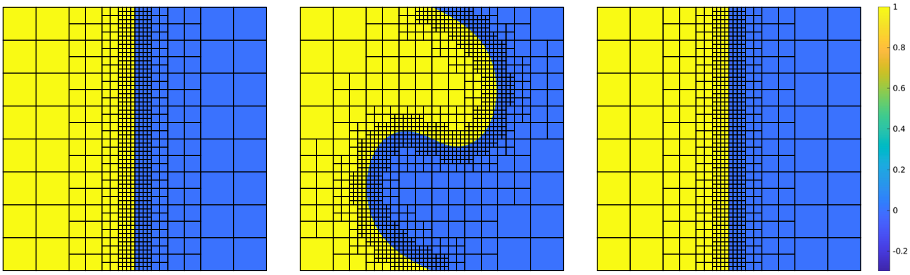

Fig. 12 Solution of the swirl problem with piecewise constant initial values at times $t = 0$ (left), $t = 0.5$ (middle) and $t = 1$ (right)

To check the accuracy we compute numerical solutions at time $t = 0.05$. At this time, no shocks have formed, and the solution structure is still smooth. The time steps satisfy CFL ≤ 0.5. Furthermore, we use subcycling and the conservative fix. To test the accuracy of the adaptive method we enforce refinement along the diagonal from the upper left patch to the lower right patch. This test (not shown here) confirms third order accuracy.

At later times shocks arise and we used the bound preserving limiter introduced in [13]. Note that the characteristic speed changes sign which has been observed to lead to some numerical difficulties as explained in [13, 19]. By updating the point values using the approach from Sect. 2.2 we can avoid those problems.

Our unlimited method shows some unphysical oscillations along shock curves as can be seen by zooming into the second and third plots in Fig. 13 (top). The use of the bound preserving limiter avoids these inaccuracies. In this simulation grids on levels $3-5$ are used.

As a refinement criterion we search for steep gradients and refine a patch if

$$\frac{q_{i+1,j} - q_{i-1,j}}{2\Delta x} \geq 15 \text{ or } \frac{q_{i,j+1} - q_{i,j+1}}{2\Delta y} \geq 15,$$

for any i, j. We also use the refinement criteria to determine whether we need to apply limiting.

4.5 Acoustics

4.5.1 Convergence Study for Acoustics

In order to investigate the order of convergence of the adaptive Active Flux method for acoustics, we consider a test problem from Lukácŏvá et al. [21], for which the exact solution is explicitly known. In this test problem the acoustic equations (11) with initial values of the form

$$\begin{aligned} p(x, y, 0) &= -\frac{1}{c}\left((\sin(2\pi x) + \sin(2\pi y)\right) \\ u(x, y, 0) &= 0 \\ v(x, y, 0) &= 0 \end{aligned} \tag{35}$$

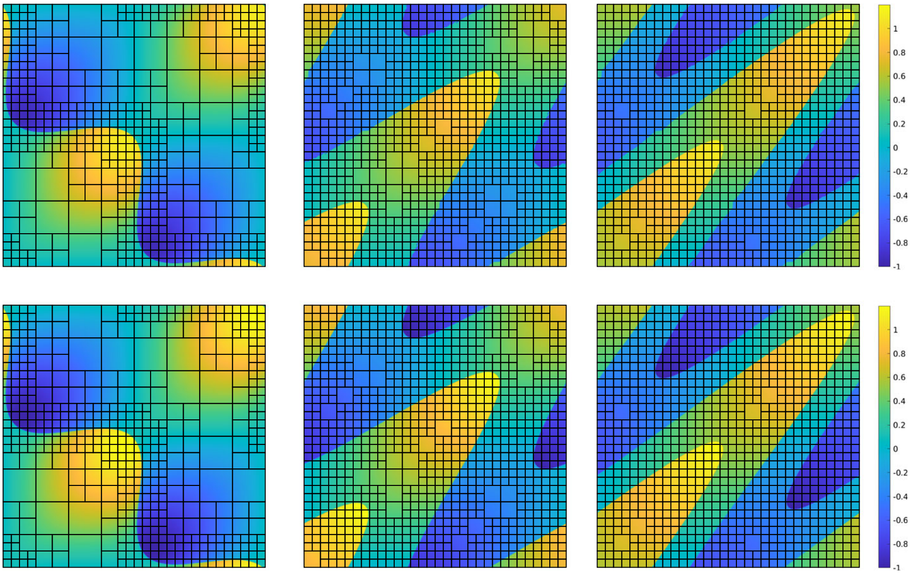

Fig. 13 Solution to (34) at times $t = 0.2$ (left), $t = 0.6$ (middle) and $t = 1.0$ (right) without limiter (top) and with limiter (bottom) for refinement level $3 - 5$

Table 4 Error at time $t = 1$ measured in $\|\cdot\|_1$-norm and EOC for the Lukáčová test problem

Level	Error		EOC	
	p	u, v	p	u, v
2	2.48×10^{-4}	8.85×10^{-6}	–	–
3	3.11×10^{-5}	1.07×10^{-6}	2.9952	3.0435
2-3	1.82×10^{-4}	2.91×10^{-5}	–	–
3-4	2.29×10^{-5}	3.68×10^{-6}	2.9936	2.9840

The top two rows show errors on uniformly refined grids at levels 2 and 3. The bottom rows show errors on adaptive mesh with two levels of refinement each

are considered on the domain $[-1, 1] \times [-1, 1]$. The speed of sound is set to $c = 1$ and periodic boundary conditions are imposed. The exact solution has the form

$$\begin{aligned} p(x, y, t) &= -\frac{1}{c}\cos(2\pi ct)\,(\sin(2\pi x) + \sin(2\pi y)) \\ u(x, y, t) &= \frac{1}{c}\sin(2\pi ct)\cos(2\pi x) \\ v(x, y, t) &= \frac{1}{c}\sin(2\pi ct)\cos(2\pi y). \end{aligned}$$

We compute numerical solutions at time $t = 1$ using time steps which allow us to maintain a CFL condition of 0.4. For the update of the point values we use the method from [17], which was briefly reviewed in Sect. 2.3. For this problem, adaptive mesh refinement does not offer any obvious benefit. Nevertheless, we can use this problem to test the accuracy of the adaptive method by constructing a static mesh using refinement criteria that is independent of the solution. We refine patches along the diagonal of the domain, and compare grids with

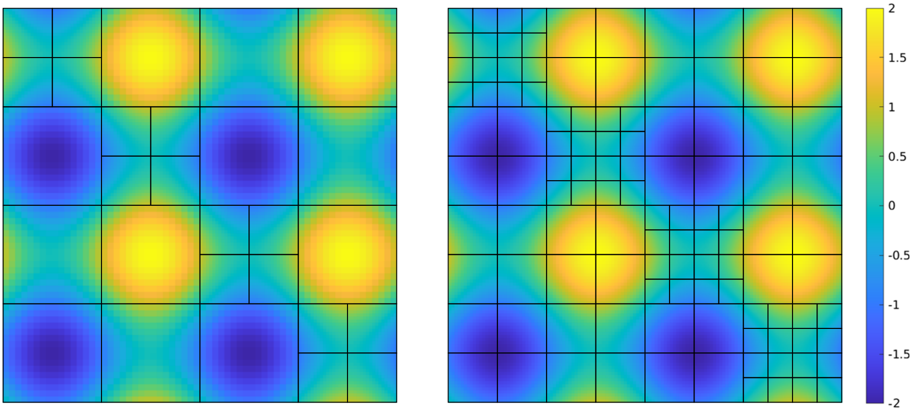

Fig. 14 Pressure and grid patches for the acoustics problem from Lukáčǒvá et al. [21] with refinement levels 2-3 (left) and 3-4 (right) at $t = 1.0$

refinement levels 2-3 and 3-4 with results obtained on uniform grids at levels 2 and 3. The results of our convergence study for pressure and velocity are shown in Table 4.

The accuracy obtained on the adaptively refined mesh is comparable to the accuracy obtained on a mesh refined uniformly to the coarser level. This test shows that the accuracy is maintained at the interface even as the solutions moves between coarser and finer levels.

4.5.2 High Frequency Acoustics

We consider a test problem where adaptive mesh refinement offers obvious benefit. We consider the acoustic equations (11) with initial condition

$$\begin{aligned} p(x, y, 0) &= 2 + \exp(-100(r(x, y) - 0.5)^2)) \sin(100r(x, y)), \\ u(x, y, 0) &= 0, \quad v(x, y, 0) = 0 \end{aligned} \tag{36}$$

on the domain $[-1.5, 1.5] \times [-1.5, 1.5]$. The method from [17] was again used to update the point values. Initially a circular shaped acoustic wave with high frequency pressure oscillations is given. This leads to acoustic waves moving outwards and inwards. The inward moving acoustic wave gets reflected in the center of the domain (around the time shown in the third plot) and afterwards propagates outwards. Adaptive mesh refinement is used to resolve these high frequency waves. Figure 15 shows solutions at times $t = 0, 0.3, 0.6, 0.9$ with refinement level $3 - 6$. A patch is refined if $|p_{max} - p_{min}| \geq 0.001$. Figure 16 shows scatter plots of the solution at two different times. There are no visible spurious grid effects apart from some smearing of the solution structure caused by the boundary of the computational domain. For this simulation periodic boundary conditions were used.

4.6 Approximation of Steady States

Finally, we consider the approximation of a stationary vortex as described by Barsukow et al. [2, 5]. The initial values have the form

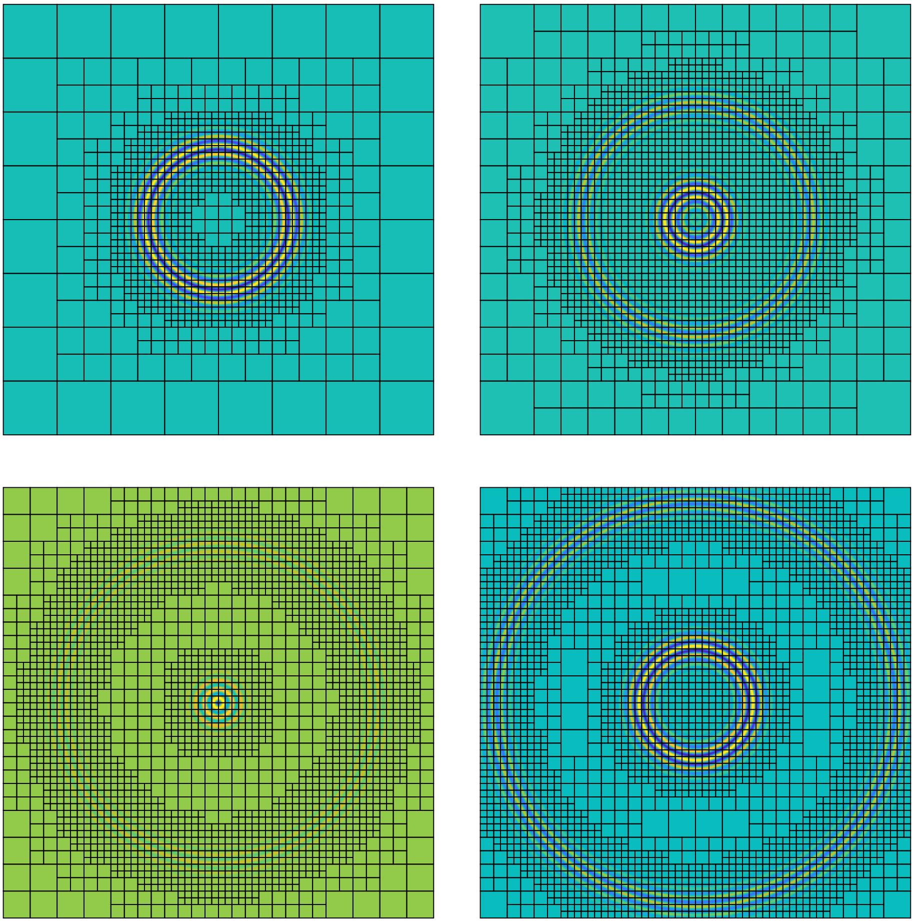

Fig. 15 Plots of pressure of the high frequency acoustic wave at times $t = 0.0$ (top left), $t = 0.3$ (top right), $t = 0.6$ (bottom left) and $t = 0.9$ (bottom right). In the bottom left plot the pressure at the center is much larger and therefore a different color map is used to visualize the solution structure

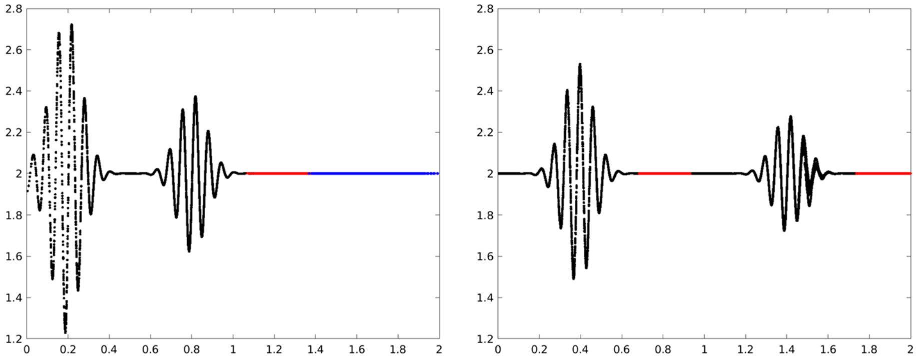

Fig. 16 Scatter plots of pressure for the high frequency acoustic wave at time $t = 0.3$ and $t = 0.9$. On level 6 grids the solution is plotted in black, on level 5 grids in red and on level 4 & 3 grids in blue

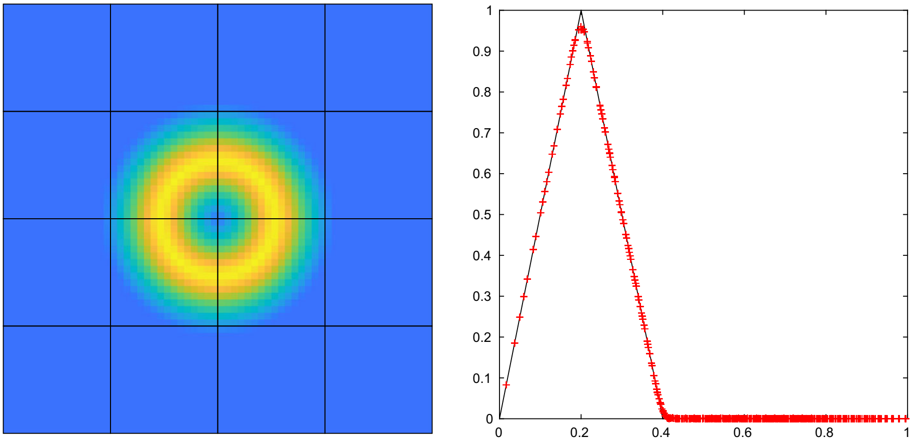

Fig. 17 Computation of the stationary vortex at time $t = 100$ using a uniformly refined Cartesian grid at level 2 The left plot shows $|\boldsymbol{u}|$, the right plot shows a scatter plot of $|\boldsymbol{u}|$. For the update of the point values the exact evolution operator in the form described in [5] was used

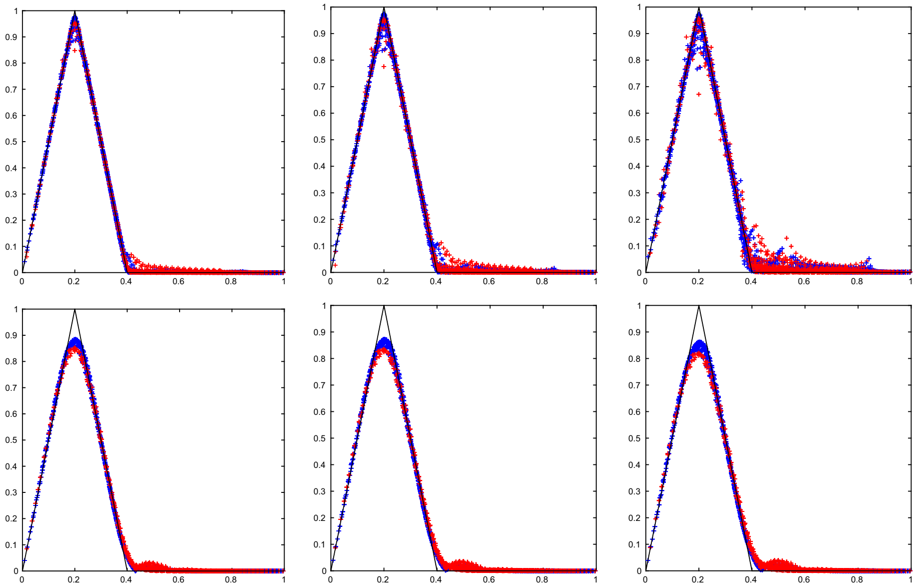

Fig. 18 Scatter plots of the vortex problem with adaptive mesh refinement along the diagonal with grids of level 2–3 using the exact evolution operator from [5] (top) and the method from [17], which is exact for irrotational flows only (bottom). The level 2 grid solution is plotted in red and the level 3 grid solution is plotted in blue. The solution structure is shown at $t = 30$, $t = 40$ and $t = 50$

$$p(r,0) = 0\boldsymbol{u}(r) = \boldsymbol{n}\begin{cases} 5r & : \quad 0 \leq r \leq 0.2 \\ 2 - 5r & : 0.2 < r \leq 0.4 \\ 0 & : \qquad r > 0.4, \end{cases}$$

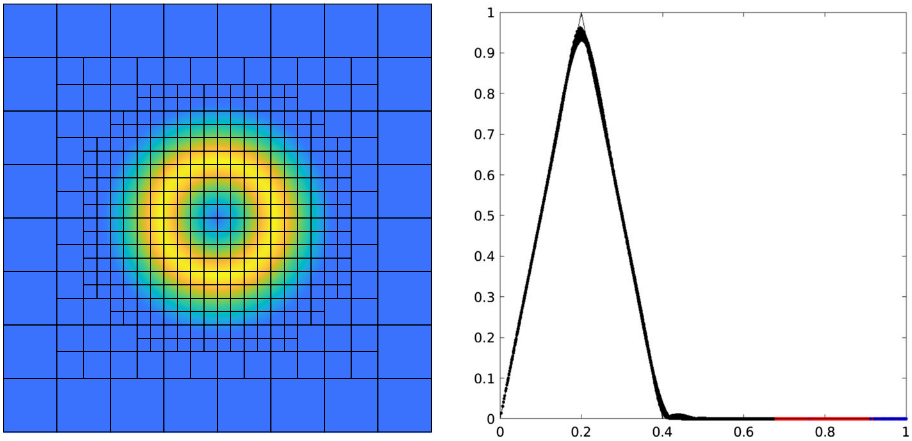

Fig. 19 Computation of the stationary vortex at time $t = 100$ using an adaptively refined grid of level 3-5. The left plot shows $|\boldsymbol{u}|$, the right plot shows a scatter plot of $|\boldsymbol{u}|$. For the update of the point values the evolution operator form [17] was used. On level 5 grids the solution is plotted in black, on level 4 grids in red and on level 3 grids in blue

with $r = \sqrt{x^2 + y^2}$, $\boldsymbol{n} = (-\sin\phi, \cos\phi)^T$, $\phi \in [0, 2\pi)$ and $\boldsymbol{u} = (u, v)^T$. In order to test how well the method preserves the steady state, the numerical solution is computed at time $t = 100$. These initial values do not have constant vorticity and thus the evolution formula of [17] (compare with Sect. 2.3) is not exact.

In [2, 5], Barsukow et al. showed that the Cartesian grid Active Flux method, with the evolution operator presented in [5], is stationary preserving. Numerical results shown in Fig. 17 confirm these findings. Here the solution structure at time $t = 100$ was computed on level 2 grids, each patch using 16×16 cells plus ghost cells. This results in a 64×64 grid for the whole domain. For this particular solution the numerical method does not introduce any numerical viscosity.

In order to test accuracy and stability with adaptive mesh refinement, we also performed computations using refinement along the diagonal of the computational domain as shown in Fig. 14 (left). For the AMR computation with exact evolution on each Cartesian grid patch we observe an instability while the method from [17], which is not exact for this flow and instead introduces some numerical viscosity, remains stable. Figure 18 shows scatter plots for both methods at different times. One possible approach to overcome this instability might consist of an exact approximation of point values at grid cell boundaries with different resolution on both sides of the interface. This has not been considered as it would make the AMR approach more complicated.

Finally, in Fig. 19, we approximate the vortex problem with refinement around the vortex structure using the method from [17] for the update of the point values. Grids at level 3–5 are used, which leads to an accurate approximation of the solution structure.

Conclusions

We showed that the Active Flux method can be used on Cartesian grids with adaptive mesh refinement and subcycling. The transfer of grid information between different Cartesian grid patches can be implemented without loss of third order accuracy by making use of the degrees of freedom of the Active Flux method. Our approach benefits from the local stencil of the Active Flux method.

For advective transport in a spatially and temporally varying velocity field, new Active Flux methods have been presented. We also discussed an Active Flux method which preserve constant states on regular Cartesian grids and grids with adaptive mesh refinement without subcycling. In practical computations with subcycling we also observed good accuracy although constant states are not exactly preserved.

The AMR concept of the Active Flux method can also be used for two-dimensional linear hyperbolic systems as illustrated by the acoustic equations. However, the preservation of steady states, a property that was recently shown for the Active Flux method on regular Cartesian grids [2], does not carry over to adaptively refined meshes with the approaches presented in this paper.

Our AMR version of the Active Flux method makes use of the compact stencil and the fact that the reconstruction can be evaluated whenever needed. The update of the point values could be replaced by alternative third order accurate evolution formulas as long as they use the same compact stencil. Therefore, our AMR approach should be useful also for other Active Flux methods that might be developed in the future. Active Flux methods with larger stencils would lead to a less efficient communication between grid patches in particular if subcycling is used but the general idea would still extend to these stencils.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 325695158. Furthermore, research was supported by a subcontract with Boise State University, Boise, ID, under DARPA Cooperative Agreement HR00112120003 with Embry-Riddle Aeronautical University, Daytona Beach, FL, USA. This work is approved for public release; distribution is unlimited. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

Data Availability No experimental data was produced and used in this paper. For our implementations we used ForestClaw, which is publicly available on GitHub. We added the Active Flux method as a new solver in ForestClaw.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

A Second Order Accuracy of the Active Flux Method which Exactly Preserves Constant States

In Sect. 4.3.2 we introduced an Active Flux method that preserves steady states for advective transport in a divergence free velocity field defined via a stream function. The method is based on the approximation (31), which was used in the definition of numerical fluxes (32) and (33). Here we prove second order accuracy of the method on uniform grids assuming that the solution is sufficiently smooth. For simplicity we assume $\Delta x = \Delta y =: h$.

Let

$$
\begin{aligned}
R_{i\pm\frac{1}{2},j} &:= F^{Simpson}_{i\pm\frac{1}{2},j} - F_{i\pm\frac{1}{2},j} \\
&= \frac{1}{36}\sum_{\ell=0}^{2} w_\ell \Big[q(x_{i\pm\frac{1}{2}}, y_{j-\frac{1}{2}}, t_{n+\ell+\frac{1}{2}}) \\
&\quad \Big(a(x_{i\pm\frac{1}{2}}, y_{j-\frac{1}{2}}, t_{n+\ell-\frac{1}{2}}) - a(x_{i\pm\frac{1}{2}}, y_j, t_{n+\ell-\frac{1}{2}}) \Big) \\
&\quad + q(x_{i\pm\frac{1}{2}}, y_{j+\frac{1}{2}}, t_{n+\ell+\frac{1}{2}}) \\
&\quad \Big(a(x_{i\pm\frac{1}{2}}, y_{j+\frac{1}{2}}, t_{n+\ell-\frac{1}{2}}) - a(x_{i\pm\frac{1}{2}}, y_j, t_{n+\ell-\frac{1}{2}}) \Big) \Big],
\end{aligned}
\tag{37}
$$

with $w_0 = w_2 = 1$ and $w_1 = 4$, describe the difference between a flux computation using Simpson's rule and the flux approximation (32). Analogously we define $R_{i,j\pm\frac{1}{2}} := G^{Simpson}_{i,j\pm\frac{1}{2}} - G_{i,j\pm\frac{1}{2}}$.

We will show that

$$
\frac{\Delta t}{h}\left(R_{i+\frac{1}{2},j} - R_{i-\frac{1}{2},j}\right) + \frac{\Delta t}{h}\left(R_{i,j+\frac{1}{2}} - R_{i,j-\frac{1}{2}}\right) = \mathcal{O}(\Delta t^3 + h^2). \tag{38}
$$

Thus, during each time step the cell average values computed with the method from Sect. 4.3.2 differ from the cell average values computed with the third order accurate Active Flux method by a term of order $\mathcal{O}(h^2)$. Since both $F^{Simpson}_{i\pm\frac{1}{2},j}$ as well as (32) use Simpson's rule for the integration in time, we only need to study the accuracy in space. Therefore, it is enough to show

$$
R^n := \frac{1}{h}\left((R^n_{i+\frac{1}{2},j} - R^n_{i-\frac{1}{2},j}) + (R^n_{i,j+\frac{1}{2}} - R^n_{i,j-\frac{1}{2}})\right) = \mathcal{O}(h^2), \tag{39}
$$

where R^n describes the update due to the terms at level t_n only. To simplify notation we will in the following omit the argument t_n. Using the finite difference approximations (29), R^n can be expressed in the form

$$
\begin{aligned}
R^n = \frac{1}{h^2}\Big[& q(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}})\big(\Psi(x_{i+\frac{1}{2}}, y_{j-1}) - \cancel{\Psi(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}})} - \Psi(x_{i+\frac{1}{2}}, y_j) + \Psi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) \\
& - \Psi(x_{i+1}, y_{j-\frac{1}{2}}) + \cancel{\Psi(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}})} + \Psi(x_i, y_{j-\frac{1}{2}}) - \Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}})\big) \\
& + q(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})\big(- \Psi(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + \Psi(x_{i+\frac{1}{2}}, y_j) + \cancel{\Psi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})} - \Psi(x_{i+\frac{1}{2}}, y_{j+1}) \\
& + \Psi(x_{i+1}, y_{j+\frac{1}{2}}) - \cancel{\Psi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})} - \Psi(x_i, y_{j+\frac{1}{2}}) + \Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}})\big) \\
& + q(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}})\big(- \Psi(x_{i-\frac{1}{2}}, y_{j-1}) + \cancel{\Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}})} + \Psi(x_{i-\frac{1}{2}}, y_j) - \Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) \\
& + \Psi(x_{i-1}, y_{j-\frac{1}{2}}) - \cancel{\Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}})} - \Psi(x_i, y_{j-\frac{1}{2}}) + \Psi(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}})\big) \\
& + q(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}})\big(\Psi(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}) - \Psi(x_{i-\frac{1}{2}}, y_j) - \cancel{\Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}})} + \Psi(x_{i-\frac{1}{2}}, y_{j+1}) \\
& - \Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) + \cancel{\Psi(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}})} + \Psi(x_i, y_{j+\frac{1}{2}}) - \Psi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})\big)\Big].
\end{aligned}
$$

Here we ordered the terms in four groups, which we denote with I_1, I_2, I_3 and I_4, respectively. We first expand I_1 around $(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}})$, I_2 around $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$, I_3 around $(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}})$

and I_4 around $(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}})$. By ignoring higher order terms and omitting the argument for the Ψ terms, we obtain

$$\begin{aligned}
I_1 &= q(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) \cdot \left(\frac{1}{2}h^2\Psi_{yy} - \frac{1}{2}h^2\Psi_{xx} + \frac{1}{8}h^3\Psi_{yyy} + \frac{1}{8}h^3\Psi_{xxx} + \frac{1}{24}h^4\Psi_{yyyy} - \frac{1}{24}h^4\Psi_{xxxx}\right)\\
I_2 &= q(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) \cdot \left(-\frac{1}{2}h^2\Psi_{yy} + \frac{1}{2}h^2\Psi_{xx} + \frac{1}{8}h^3\Psi_{yyy} - \frac{1}{8}h^3\Psi_{xxx} - \frac{1}{24}h^4\Psi_{yyyy} + \frac{1}{24}h^4\Psi_{xxxx}\right)\\
I_3 &= q(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}) \cdot \left(-\frac{1}{2}h^2\Psi_{yy} + \frac{1}{2}h^2\Psi_{xx} - \frac{1}{8}h^3\Psi_{yyy} + \frac{1}{8}h^3\Psi_{xxx} - \frac{1}{24}h^4\Psi_{yyyy} + \frac{1}{24}h^4\Psi_{xxxx}\right)\\
I_4 &= q(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) \cdot \left(\frac{1}{2}h^2\Psi_{yy} - \frac{1}{2}h^2\Psi_{xx} - \frac{1}{8}h^3\Psi_{yyy} - \frac{1}{8}h^3\Psi_{xxx} + \frac{1}{24}h^4\Psi_{yyyy} - \frac{1}{24}h^4\Psi_{xxxx}\right).
\end{aligned}$$

Next we expand $I_1 + I_2$ around $(x_{i+\frac{1}{2}}, y_j)$ and $I_3 + I_4$ around $(x_{i-\frac{1}{2}}, y_j)$ and obtain, by again omitting the arguments in the Ψ terms,

$$\begin{aligned}
I_1 + I_2 &= q(x_{i+\frac{1}{2}}, y_j) \cdot \left(-\frac{h^3}{4}\Psi_{yyy} + \frac{h^3}{2}\Psi_{xxy} - \frac{h^4}{8}\Psi_{xxxy}\right)\\
&\quad + q_y(x_{i+\frac{1}{2}}, y_j) \cdot \left(-\frac{h^3}{2}\Psi_{yy} + \frac{h^3}{2}\Psi_{xx} - \frac{h^4}{8}\Psi_{xxx}\right)\\
I_3 + I_4 &= q(x_{i-\frac{1}{2}}, y_j) \cdot \left(\frac{h^3}{4}\Psi_{yyy} - \frac{h^3}{2}\Psi_{xxy} - \frac{h^4}{8}\Psi_{xxxy}\right)\\
&\quad + q_y(x_{i-\frac{1}{2}}, y_j) \cdot \left(\frac{h^3}{2}\Psi_{yy} - \frac{h^3}{2}\Psi_{xx} - \frac{h^4}{8}\Psi_{xxx}\right).
\end{aligned}$$

In the final step we expand all the terms $I_1, \ldots, I_4$ around (x_i, y_j) and obtain for the sum of the four terms

$$\begin{aligned}
&I_1 + I_2 + I_3 + I_4 = q(x_i, y_j)\left(-\frac{3h^4}{8}\Psi_{xyyy} + \frac{h^4}{4}\Psi_{xxxy}\right)\\
&\quad + q_x(x_i, y_j)\left(-\frac{h^4}{4}\Psi_{yyy} + \frac{h^4}{2}\Psi_{xxy}\right)\\
&\quad + q_y(x_i, y_j)\left(-\frac{h^4}{2}\Psi_{xyy} + \frac{h^4}{4}\Psi_{xxx}\right) + q_{xy}(x_i, y_j)\left(-\frac{h^4}{2}\Psi_{yy} + \frac{h^4}{2}\Psi_{xx}\right).
\end{aligned}$$

Thus, we have

$$R^n = \frac{1}{h^2}\left(I_1 + I_2 + I_3 + I_4\right) = \mathcal{O}(h^2),$$

i.e. the method, which exactly preserves constant states, is second order accurate.

In the special case of solid body rotation, the derivatives of Ψ arising in the leading order error term vanish and we obtain full third order accuracy.

In Table 5 we show a numerical convergence study for the smooth swirl flow test problem of Sect. 4.3.1 but at time $t = 0.1$ instead of $t = 1$. For this time instance we do not have an exact solution but compare the numerical solution on level 3 and 4 grids with a reference solution computed on a level 6 grid. For the method which exactly preserves constant states (right table) the convergence study confirms second order accuracy, while the classical Active Flux method (left table) is third order accurate.

Table 5 Error comparing to a level 6 reference solution at time $t = 0.1$ measured in the 1-norm and convergence rates for the advection problem using Simpson's rule and exact derivatives, i.e. $u = -\Psi_y$, $v = \Psi_x$, (left table) and using the approach of Sect. 4.3.1 (right table)

Level	Error	EOC	Level	Error	EOC
3	2.16×10^{-6}	–	3	1.32×10^{-5}	–
4	2.70×10^{-7}	2.9983	4	3.26×10^{-6}	2.011

References

1. Barsukow, W.: Low mach number finite volume methods for the acoustic and euler equations. Doctoral thesis, Universität Würzburg (2018)
2. Barsukow, W.: Stationarity preservation properties of the active flux scheme on cartesian grids. Commun. Appl. Comput. (2020)
3. Barsukow, W.: The Active Flux scheme for nonlinear problems. J. Sci. Comput. **86**(3), 1–34 (2021)
4. Barsukow, W., Berberich, J.P., Klingenberg, C.: On the active flux scheme for hyperbolic PDEs with source terms. SIAM J. Sci. Comput. **43**, A4015–A4042 (2021)
5. Barsukow, W., Hohm, J., Klingenberg, C., Roe, P.L.: The active flux scheme on Cartesian grids and its low Mach number limit. J. Sci. Comput. **81**(1), 594–622 (2019)
6. Berger, M.J., Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. J. Comput. Phys. **82**, 64–84 (1989)
7. Berger, M.J., LeVeque, R.J.: Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. SIAM J. Numer. Anal. **35**, 2298–2316 (1998)
8. Berger, M.J., Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. J. Comput. Phys. **53**, 484–512 (1984)
9. Burstedde, C., Calhoun, D., Mandli, K., Terrel, A.R.: Forestclaw: hybrid forest-of-octrees amr for hyperbolic conservation laws. Parallel Comput. Accel. Computat. Sci. Eng. (CSE) **25**, 253–262 (2014)
10. Burstedde, C., Wilcox, L.C., Ghattas, O.: p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. SIAM J. Sci. Comput. **33**, 1103–1133 (2011)
11. Calhoun, D., Burstedde, C.: Forestclaw software. http://www.forestclaw.org/ForestClaw/index.html
12. Calhoun, D., Burstedde, C.: ForestClaw: a parallel algorithm for patch-based adaptive mesh refinement on a forest of quadtrees (2017). arXiv:1703.03116
13. Chudzik, E., Helzel, C., Kerkmann, D.: The Cartesian grid active flux method: linear stability and bound preserving limiting. Appl. Math. Comput. **393**, 125501, 19 (2021)
14. Courant, R., Hilbert, D.: Methods of Mathematical Physics, vol. 2. Wiley-VCH, New Jersey (1962)
15. Eymann, T.A., Roe, P.L.: Active flux schemes. AIAA. 2011-382
16. Eymann, T.A., Roe, P.L.: Active flux schemes for systems. AIAA 2011-3840
17. Eymann, T.A., Roe, P.L.: Multidimensional active flux schemes. AIAA Conference Paper, June 2013
18. Fan, D., Roe, P.L.: Investigations of a new scheme for wave propagation. AIAA Aviation Forum, 22-26 June 2015, Dallas, TX, 22nd AIAA Computational Fluid Dynamics Conference
19. Helzel, C., Kerkmann, D., Scandurra, L.: A new ADER method inspired by the active flux method. J. Sci. Comput. **80**(3), 1463–1497 (2019)
20. LeVeque, R.J.: Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge (2002)
21. Lukáčová-Medvid'ová, M., Morton, K.W., Warnecke, G.: Evolution galerkin methods for hyperbolic systems in two space dimensions. Math. Comput. **69**, 1355–1384 (2000)
22. Lukáčová-Medvid'ová, M., Saibertová, J., Warnecke, G.: Finite volume evolution Galerkin methods for nonlinear hyperbolic systems. J. Comput. Phys. **183**, 533–562 (2002)
23. Roe, P.: Is discontinuous reconstruction really a good idea? J. Sci. Comput. **73**(2–3), 1094–1114 (2017)
24. Roe, P.: Did numerical methods for hyperbolic problems take a wrong turning? In: Klingenberg, C., Westdickenberg, M. (eds.) Theory, Numerics and Applications of Hyperbolic Problems II, pp. 517–534. Springer, Cham (2018)
25. Roe, P.: Designing CFD methods for bandwidth - a physical approach. Comput. Fluids **214**, 104774 (2021)
26. Roe, P.L., Maeng, J., Fan, D.: Comparing active flux and discontinuous galerkin methods for compressible flow. 2018 AIAA Aerospace Science Meeting

27. Zhang, W., Almgren, A., Beckner, V., Bell, J., Blaschke, J., Chan, C., Day, M., Friesen, B., Gott, K., Graves, D., Katz, M.P., Myers, A., Nguyen, T., Nonaka, A., Rosso, M., Williams, S., Zingale, M.: AMReX: a framework for block-structured adaptive mesh refinement. J. Open Source Softw. **4**(37), 1370 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.