Journal of Computational Physics 229 (2010) 1467–1497

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods

Richard Liska [a], Mikhail Shashkov [b,*], Pavel Váchal [a], Burton Wendroff [b]

[a] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Břehová 7, Praha 1, 115 19, Czech Republic
[b] MS B284, Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 4 August 2009
Received in revised form 21 October 2009
Accepted 25 October 2009
Available online 1 November 2009

MSC:
65M06
65Z05

Keywords:
FCT
Flux-corrected remapping
ALE

ABSTRACT

A new optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian hydro methods is described. Fluxes of conserved variables – mass and momentum – are limited in a synchronous way to preserve local bounds of primitive variables – density and velocity.

Published by Elsevier Inc.

1. Introduction and background

In numerical simulations of fluid flow, the choice of the computational grid is crucial. Traditionally, there have been two viewpoints, utilizing the Lagrangian or the Eulerian framework, each with its own advantages and disadvantages. In a pioneering paper [8], Hirt et al. developed the formalism for a grid whose motion could be determined as an independent degree of freedom, and showed that this general framework could be used to combine the best properties of Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian–Eulerian or ALE. Many authors have described ALE strategies to optimize accuracy, robustness, or computational efficiency, see for example [2,3,21,10,11,24,17].

In this paper we consider only ”standard” ALE methods when connectivity of the mesh does not change during the calculation.

The ALE scheme can be formulated as a single algorithm [7] based on solving the equations in a moving coordinate frame. However it is more usual to split it into three separate phases. These are: (1) a Lagrangian phase in which the solution and grid are updated; (2) a rezoning phase in which the nodes of the computational grid are moved to a more optimal position; and (3) a remapping phase in which the Lagrangian solution is conservatively interpolated onto the rezoned grid.

[*] Corresponding author.
E-mail addresses: liska@siduri.fjfi.cvut.cz (R. Liska), shashkov@lanl.gov (M. Shashkov), vachal@galileo.fjfi.cvut.cz (P. Váchal), bbw@lanl.gov (B. Wendroff).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2009.10.039

It is possible to use the ALE formalism to run in a mainly Lagrangian mode, with an occasional rezone/remap whenever the grid becomes too distorted. However, it is generally more effective to rezone and remap on each cycle, a strategy termed continuous rezoning. One advantage of continuous rezone is that the individual grid movements can be constrained to be small, allowing the use of a local remapper where mass, momentum and total energy (conserved quantities) are only exchanged between neighboring cells.

In this paper we will consider a situation when all conserved quantities in the remapping phase are defined at cell centers (which is the case for cell-centered methods such as Godunov [19,20,18], or staggered methods where subcells formalism is used as a tool for remap [16]).

We will consider remap of mass (density) and momentum (velocity). Mass and momentum are conservative variables, density and velocity are primitive variables. The density in a rezoned cell is defined as the ratio of remapped mass and volume of the rezoned cell; and velocity in the rezoned cell is defined as the ratio of remapped momentum and remapped mass.

Three main properties of remapping are: conservation (in our case conservation of total mass and total momentum), accuracy and bounds preservation of primitive variables density and velocity.

For the case of cell-centered conserved quantities and continuous rezone strategy, the exchange of conserved quantities between neighboring cells can be formulated in flux form (see, for example, [23]). The flux form of remapping guarantees both local and global conservation.

Because we use a continuous rezone strategy it implies that a rezoned cell is contained in the union of its Lagrangian prototype and its neighbors. Therefore it is natural to require that the value of remapped quantity in a rezoned cell is bounded by the maximum and minimum values on corresponding Lagrangian cells.

Physically motivated bounds are imposed on primitive variables density and velocity, rather than on conserved quantities. Because remapped velocity is defined as the ratio of remapped momentum and remapped mass, accuracy and bound preservation property of the velocity depends on both remapped quantities. In other words remapping of mass and momentum has to be considered as coupled process, and therefore remapping of mass and momentum has to be synchronized (see, [25] – a paper which in some sense inspired our work). Using terminology introduced in [25] we will call fluxes of mass and momentum *compatible* if they guarantee bounds preservation for primitive variables.

The task of finding compatible fluxes can be considered as a constrained global optimization problem: an objective function related to some notion of accuracy and bounds preservation defines constraints.

Clearly the solution of a global optimization problem is not practical. The main goal of this paper is to show how using ideas inspired by *Flux-Corrected Transport (FCT)* method [4,30,14,25] we can replace the global optimization problem by a series of local problems.

FCT type methods are usually considered in the framework of solution of advection or system of hyperbolic partial differential equations, [14,15,1]. To the best of our knowledge the only attempt to directly use FCT ideas in remapping context is done in [28]. For this reason we will give a brief overview of relevant FCT ideas and references and then describe specific remapping related issues.

According to the Preface of the book [14] (which can be considered as the most comprehensive description of current state of flux-corrected methods) the FCT idea "... was to locally replace formal truncation error considerations with conservative monotonicity enforcement in those places in the flow where formal truncation error had lost its meaning, i.e. where the solution was not smooth and where formally high-order methods would violate physically-motivated upper and lower bounds on the solution." Technically monotonicity enforcement is achieved by combining low-order fluxes (which guarantee bound preservation) with high-order fluxes in such a way that resulting fluxes guarantee preservation of bounds and at the same time are as close as possible to high-order fluxes (low-order and high-order refers here to formal order of accuracy on smooth solutions). This logic usually applies to solution of one partial differential equation (PDE), e.g. the scalar advection equation. However, even for one equation it is not clear a priori why closeness of the fluxes to formally high-order fluxes gives better accuracy, because accuracy depends not only on how close fluxes are to formally high-order fluxes but also how consistent are fluxes contributing to change of particular quantity in the cell. Inconsistency in the fluxes can manifest itself in artificial steepening of the solution [14, p. 149], or other anomalies.

A more general approach will be again to consider a global constrained optimization problem. Moreover, even for one equation one of the main problems is to define "physically-motivated upper and lower bounds on the solution" because the solution is unknown (see [14] for relevant discussion). Our opinion is that the FCT approach is a process of replacing a global constrained optimization problem by series of local constrained optimization problems by considering the *worst case scenario*. By doing this one does not perform global optimization, but still is able to find fluxes which give a more accurate solution than the low-order fluxes, and which guarantee bounds preservation. Such solution of local problems is clearly a more practical approach, and also from the physical point of view one can say that causality principle is not violated. For one PDE the sufficient local condition which guarantees bound preservation is practically unique and based on separation of positive (*in*) and negative (*out*) fluxes. Then for upper bound the worst case scenario is if all negative limited andiffusive fluxes will be zeroed out by the limiter, and for lower bound the worst case scenario is if all positive limited antidiffusive fluxes are zeroed out by the limiter. For systems of PDEs there are no unique sufficient conditions and therefore one can construct several sets of compatible fluxes. Therefore under the assumption that the notions of local accuracy and bounds are the same, methods of construction compatible fluxes will differ only in what sufficient conditions on flux limiting (correction) are used.

As it has been mentioned in [14, p.150] "... in the attempt to extend the limiting process to systems of PDEs no immediately obvious or natural limiting procedure becomes apparent." On p. 218 of the same book the authors have mentioned that ‘‘Despite the remarkable progress made in the development of high-resolution schemes for scalar conservation laws, their extension to hyperbolic systems remains a challenging open problem.” We want to add to this statement that the question how accuracy depends on fluxes for system of PDEs is not obvious at all.

According to [14] there are several approaches to deal with systems of PDEs. Two approaches most often used in practice are: independent treatment of each equation as in operator-split FCT and use of the same limiter for all equations. Neither of these approaches guarantees bound preservation for primitive variables. In book [14] some abstract scheme for limiting any set of variables is presented (p. 151 and p. 223), however, as it has been mentioned on p. 224 – ‘‘At present, there is still a large degree of empiricism in the construction of synchronized FCT limiters, and their performance is strongly problem-dependent.” Therefore development of new FCT-based methods for systems of PDEs or similar problems (in our case simultaneous flux-based remapping of mass/density, momentum/velocity) is a very important area of the research.

Only a few papers have dealt with construction of compatible fluxes for systems of equations. In [26] the author considers the advection of passive scalars as well as equation of fluid density (continuity equation). There is assumption that at the stage of constructing a limiter for passive scalar the mass flux is already given. In general, we will call an approach *sequential* when mass flux is considered given when constructing fluxes for other quantities. In the sequential approach bounds on passive scalar (or other quantity) do not affect the limiting process for density. Let us also note that in [26] the author also considers a special form of flux for a passive scalar which is taken in the form of known mass flux multiplied by some value of the passive scalar. Such form of the flux for a passive scalar allows the construction of a compatible flux. In paper [6] authors consider an extension of ideas presented in [26] to the case of unstructured meshes. In paper [25], authors consider simultaneous advection of density-like dependent variables (for instance, densities of mass and momenta). In this paper, no assumption of a special form of the flux for momenta is made, and the limiting of transportive fluxes on the primary variables is derived from analytic constraints implied by the Lagrangian form of the governing continuity equations, which are imposed on the specific mixing ratios of variables (e.g. velocity components). This makes the derivation in [25] almost directly applicable to remapping. In the approach developed in [25], bounds on velocity components affect the limiting process for density. Following [25], we will call such compatible fluxes *synchronized* fluxes and the class of methods where bounds on one variable affect the limiting process for another variable will be called *synchronous*. One may hope that the synchronous approach can give overall less restrictive constraints on the fluxes than the sequential approach.

Now let us return to the topic of our paper – construction of compatible fluxes for remapping of mass/density and momentum/velocity, and draw some analogy with FCT methods for PDEs. To distinguish from FCT we will call our method flux-corrected remapping – FCR. First of all, as we have mentioned before, bounds for primitive variables(density and velocity) are well defined and are determined from corresponding values on Lagrangian mesh. Second, fluxes are usually computed from reconstruction of the function on Lagrangian mesh, where the function is given by its mean values. Fluxes are just approximation of some integrals of reconstructed functions over the pieces of intersections of Lagrangian and rezoned mesh. Usually low-order fluxes correspond to piecewise constant reconstruction and high-order fluxes correspond to piecewise linear reconstruction. Finally, the accuracy of remapped quantities also can be estimated from values on the Lagrangian mesh because the remapped discrete function is a representation of the same underlying unknown function on the new mesh.

We have introduced a new local measure of accuracy which is derived from the L_1 norm of the error for density and velocity. For each edge (face) where the flux is defined it reduces to linear function which coefficients depend on values of the density and velocity on old mesh and on low- and high-order fluxes. We also developed new local sufficient condition which guarantees bound preservation. Because we know only one truly synchronous method for systems, that is the method suggested in [25], we use this method as base line to compare with our new method. Let us consider the plane defined by limiters (correction factors) for density and momentum flux. The feasible set (that is part of the plane where any point corresponding to a pair of limiters guarantees bound preservation) for method from [25] is a rectangle and for our new method the feasible set is a convex polygon. In general feasible sets for the two methods do overlap at least at one point (which corresponds to low-order fluxes) but do not contain each other. We demonstrate superiority of our new method on a set of numerical examples.

Let us note that there are other approaches for computing compatible fluxes. For system of two advection equations one such approach is described in [29], where the authors use some special limiting procedure for *gradient limiting* in context of upstream-centered methods. In our terminology, this is a sequential method, that is, limiting of density gradient is not affected by bounds of other quantities. In the context of remapping, an interesting approach is introduced in [27]. It is also based on gradient limiting, but it is really a synchronous method. Unfortunately this method is very complicated and we do not have enough information about its performance. We are not considering these methods here because they are not FCT-based.

The rest of this paper is organized as follows: The rationale for our work is given in Section 2. In this section we describe a general problem of remapping. In Section 3, we describe the method for remapping of density by correcting mass fluxes, which follows the logic of the original FCT approach [4] as it was modified by Zalesak [30]. In Section 4 we describe our new method - Synchronized Flux-Corrected Remapping (SFCR) of constructing compatible fluxes for remapping of mass/density and momentum/velocity. We also describe our modification of method [25] for remapping. In Section 5 we demonstrate the performance of the SFCR method and compare it with method from [25] on series of 1D examples. Numerical examples in 2D are presented in Section 6.

2. Rationale

2.1. Grids

We consider a computational domain Ω. Most of the figures will show a two-dimensional mesh, however all the considerations and formulas can be used on one, two or three-dimensional meshes, structured or unstructured. We assume that we are given a mesh on Ω that consists of zones (cells) $z_i,\ i = 1, \ldots, i_{\max}$ that cover Ω without gaps or overlaps.

Each cell is defined by a set of vertices (which we will sometimes call points or nodes), denoted by $P(z_i)$, and a set of sides (which are segments of straight lines), denoted by $S(z_i)$. Each side s_k is shared by only two cells, denoted by $Z(s_k)$. Each vertex p_m may be shared by an arbitrary number of cells. We denote the set of cells that have a common vertex by $Z(p_m)$; similarly, we denote the set of all sides sharing a common vertex p_m by $S(p_m)$. The cells that share a side (edge) or vertex with a particular cell are called *neighbors* (in 3D it will be face, edge or vertex); the set of all the neighbors of a cell z_i is denoted $Z(z_i)$. The reciprocal relation of neighborhood defines the *connectivity* of the grid. In the context of ALE methods, we consider two grids with the same connectivity – i.e. the same number of cells and vertices, and the same neighbor relations. The grid that contains the cells z_i is called the Lagrangian or *old* grid. The second grid, containing the cells $\tilde{z}_i$, is called the rezoned or *new* grid. In the ALE method, the rezoned grid results from an algorithm (i.e. a rezoner) that identifies and mitigates inadequacies of the Lagrangian grid. In Figs. 1 and 2, we show examples of pairs of a Lagrangian grid (solid lines) and a rezoned grid (dashed lines). The rezoned grids were generated using the optimization-based reference Jacobian strategy described in [12]. The rezoned grid produced by this algorithm remains “close” to Lagrangian grid, but has better geometrical quality. Fig. 2 illustrates how complicated the relative locations of the two grids can be even when displacements of the nodes are small.

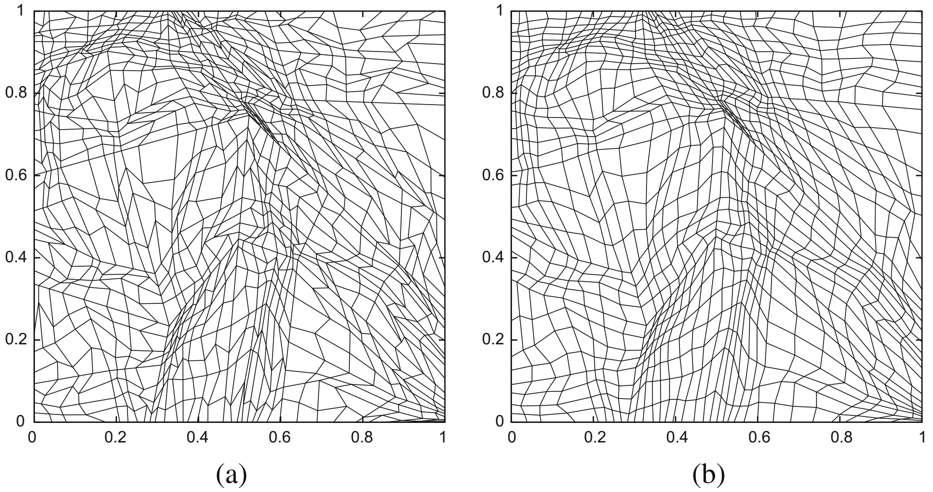

Fig. 1. (a) Old (Lagrangian) grid and (b) new (rezoned) grid.

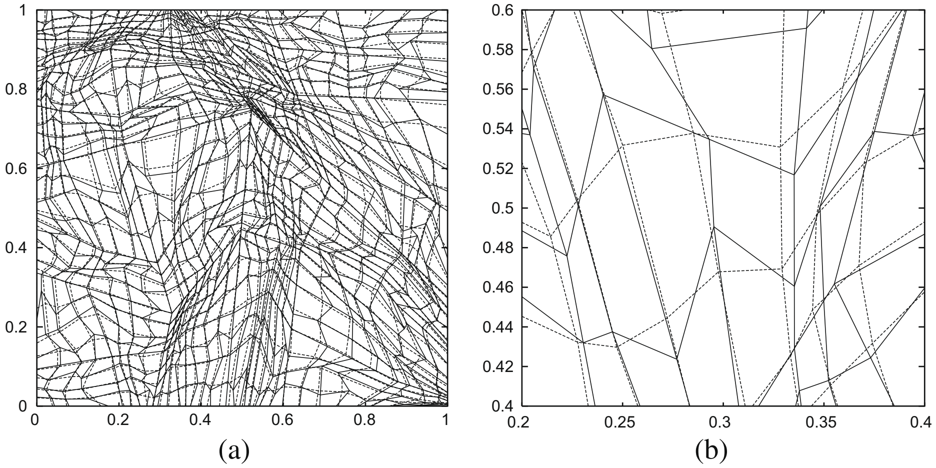

Fig. 2. Lagrangian (solid) and rezoned (dashed) grid. (a) Entire region and (b) fragment.

After rezoning, the old mesh $\{z_i\}$ is mapped into a new mesh $\{\tilde{z}_i\}$. We define a set $\mathcal{Z}(z_i) = \bigcup_k z_k$, such that

$$\tilde{z}_i \subset \mathcal{Z}(z_i). \tag{2.1}$$

For any two grids, such a set exists because $\tilde{z}_i \in \bigcup_{k=1}^{i_{\max}} z_k$. However we will always consider the minimal set for which (2.1) holds. For the rezoning method described in [12]

$$\mathcal{Z}(z_i) = z_i \bigcup Z(z_i); \tag{2.2}$$

that is, the new cell $\tilde{z}_i$ is contained in the union of the old cell z_i and its immediate neighbors, see, for example, Fig. 2b).

The volume of cell z_i is denoted by

$$V(z_i) = \int_{z_i} \mathrm{d}V.$$

In the following text, all physical quantities (e.g. the density ρ or the mass m) remapped onto the new grid will have the tilde accent (e.g. $\tilde{\rho}$) whereas values on the old (Lagrangian) grid will have no accent.

2.2. Statement of the remapping problem

2.2.1. Remapping of mass-density

We assume that there is a positive function $\rho(\mathbf{r}) > 0,\ \mathbf{r} = (x, y, z)$, which we call density, that is defined throughout the problem domain. The only information that we are given about this function is its mean value in each of the cells of the old grid:

$$\rho_i = \frac{\int_{z_i} \rho(\mathbf{r})\,\mathrm{d}V}{V(z_i)}, \tag{2.3}$$

where $V(z_i)$ is the volume of cell C_i. The numerator of (2.3) is the cell mass:

$$m_i := \int_{z_i} \rho(\mathbf{r})\,\mathrm{d}V,$$

and so the density is

$$\rho_i = \frac{m_i}{V(z_i)}.$$

The total problem mass is

$$M := \int_{\Omega} \rho(\mathbf{r})\,\mathrm{d}V = \sum_{i=1}^{i_{\max}} \int_{z_i} \rho(\mathbf{r})\,\mathrm{d}V = \sum_{i=1}^{i_{\max}} m_i = \sum_{i=1}^{i_{\max}} \rho_i V(z_i).$$

The task is to find accurate approximations $\tilde{m}_i$ for the masses of the new cells

$$\tilde{m}_i \approx m_i^{\mathrm{exact}} = \int_{\tilde{z}_i} \rho(\mathbf{r})\,\mathrm{d}V. \tag{2.4}$$

For high-order remapping we typically require *linearity preservation*, which means that if $\rho(\mathbf{r})$ is a linear function of $\mathbf{r}$ on the whole computational domain Ω then the linearity preserving remapping produces remapped masses (and densities below) which are exact, i.e. $\tilde{m}_i^{\mathrm{LP}} = m_i^{\mathrm{exact}}$.

First constraint is that the new cell masses must satisfy the global conservation of the total mass:

$$\sum_{i=1}^{i_{\max}} \tilde{m}_i = M. \tag{2.5}$$

The approximate mean values of density in the new cells are then defined by

$$\tilde{\rho}_i = \frac{\tilde{m}_i}{V(\tilde{z}_i)}. \tag{2.6}$$

Second requirement follows from assumption that we are using continuous rezone strategy and therefore new cell $\tilde{z}_i$ is contained in the union of the old cell z_i and its immediate neighbors which share with it face, edge or node, that is,

$$\tilde{z}_i \in \mathcal{Z}(z_i) = z_i \bigcup Z(z_i). \tag{2.7}$$

This assumption leads to natural requirement that the value of the remapped density, $\tilde{\rho}_i$, has to be in the following bounds:

$$\rho_i^{\min} \leqslant \tilde{\rho}_i \leqslant \rho_i^{\max}, \quad \rho_i^{\min} = \min_{\mathcal{Z}(\ddagger_\rangle)}(\rho_i), \quad \rho_i^{\max} = \max_{\mathcal{Z}(\ddagger_\rangle)}(\rho_i). \tag{2.8}$$

Using (2.6), the requirement (2.8) for remapped density can be also expressed in terms of masses

$$m_i^{\min} = \rho_i^{\min} V(\tilde{z}_i) \leqslant \tilde{m}_i \leqslant \rho_i^{\max} V(\tilde{z}_i) = m_i^{\max}. \tag{2.9}$$

2.2.2. *Remapping of momentum-velocity*

We assume that there is a vector function $\mathbf{u}(\mathbf{r})$, $\mathbf{r} = (x, y, z)$, which we call velocity, that is defined throughout the problem domain. The only information that we are given about this function is its mean value in each of the cells of the old grid:

$$\mathbf{u}_i = \frac{\int_{z_i} \mathbf{u}\rho \, \mathrm{d}V}{m_i}. \tag{2.10}$$

The numerator of (2.10) is the cell momentum

$$\boldsymbol{\mu}_i := \int_{z_i} \mathbf{u}\rho \, \mathrm{d}V.$$

The total problem momentum is

$$\boldsymbol{\mathcal{M}} := \int_{\Omega} \mathbf{u}\rho \, \mathrm{d}V = \sum_{i=1}^{i_{\max}} \int_{z_i} \mathbf{u}\rho \, \mathrm{d}V = \sum_{i=1}^{i_{\max}} \boldsymbol{\mu}_i = \sum_{i=1}^{i_{\max}} \mathbf{u}_i m_i.$$

The problem statement is to find accurate approximations $\tilde{\boldsymbol{\mu}}_i$ for the momentum of the new cells

$$\tilde{\boldsymbol{\mu}}_i \approx \boldsymbol{\mu}_i^{\mathrm{ex}} = \int_{\tilde{z}_i} \mathbf{u}\rho \, \mathrm{d}V.$$

First constraint is that the new cell momentum must satisfy the global conservation of the total momentum:

$$\sum_{i=1}^{i_{\max}} \tilde{\boldsymbol{\mu}}_i = \boldsymbol{\mathcal{M}}. \tag{2.11}$$

The approximate mean values of velocity in the new cells are defined by

$$\tilde{\mathbf{u}}_i = \frac{\tilde{\boldsymbol{\mu}}_i}{\tilde{m}_i}. \tag{2.12}$$

It is clear that remapped velocity depends both on remapped momentum and on remapped mass.

Similarly to density there are natural bounds for velocity:

$$\mathbf{u}_i^{\min} \leqslant \tilde{\mathbf{u}}_i \leqslant \mathbf{u}_i^{\max}, \quad \mathbf{u}_i^{\min} = \min_{\mathcal{Z}(\ddagger_{\rangle})}(\mathbf{u}_i), \quad \mathbf{u}_i^{\max} = \max_{\mathcal{Z}(\ddagger_{\rangle})}(\mathbf{u}_i), \tag{2.13}$$

where maximum resp. minimum are taken component by component and vector inequality is defined below. To simplify the notation we use the inequality relations operators $\geqslant$, $>$, $\leqslant$ and < applied to vectors. Having two vectors $\mathbf{A} = (A_x, A_y, A_z)$ and $\mathbf{B} = (B_x, B_y, B_z)$, we define

$$\mathbf{A} \lhd \mathbf{B} \iff (\forall \zeta \in \{x, y, z\},\ A_\zeta \lhd B_\zeta), \quad \text{for } \lhd \in \{\geqslant, >, \leqslant, <\}. \tag{2.14}$$

The constraint (2.13) can be written (using (2.12)) in equivalent form for momentum $\boldsymbol{\mu}$

$$\boldsymbol{\mu}_i^{\min} = \tilde{m}_i \mathbf{u}_i^{\min} \leqslant \tilde{\boldsymbol{\mu}}_i \leqslant \tilde{m}_i \mathbf{u}_i^{\max} = \boldsymbol{\mu}_i^{\max}, \tag{2.15}$$

or for each its component

$$\left(\boldsymbol{\mu}_i^{\min}\right)_\zeta = \tilde{m}_i (\mathbf{u}_i^{\min})_\zeta \leqslant (\tilde{\boldsymbol{\mu}}_i)_\zeta \leqslant \tilde{m}_i (\mathbf{u}_i^{\max})_\zeta = (\boldsymbol{\mu}_i^{\max})_\zeta, \tag{2.16}$$

where $\zeta \in \{x, y, z\}$.

The main topic of this paper is how to find accurate, bounded approximations for the densities (masses) and the velocities (momentum) on the new mesh, such that total mass and momentum are conserved. This task is usually referred to as *bounds-preserving remapping*, or alternately, as *bounds-preserving conservative interpolation*.

As for mass-density high-order remapping we require linearity preservation (see explanation below (2.4)) also for high-order momentum remapping. Another property worth to consider for density and momentum remapping is the DeBar consistency condition [5,3]. The remapping satisfies the DeBar condition if for constant velocity field (in the whole domain Ω) and arbitrary density the remapping reproduces the constant velocity field exactly. For example it can be easily proven, that if the mass and momentum fluxes (see below) are related by $\mathbf{F}_{i,k}^{\mu} = F_{i,k}^{m} \mathbf{u}_{i,k}^{*}$, where $\mathbf{u}_{i,k}^{*}$ is a reasonable approximation of velocity on face (i, k), then remapping satisfies the DeBar condition. We however do not use this form of momentum fluxes.

2.3. Flux form. high-order and low-order fluxes

Because we are using continuous rezone strategy we can represent values of the mass and momentum on the new mesh in flux form (see, for example, [23]):

$$\tilde{m}_i = m_i + \sum_{k \in Z(z_i)} F^m_{i,k}, \quad \tilde{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_i + \sum_{k \in Z(z_i)} \mathbf{F}^{\mu}_{i,k}, \tag{2.17}$$

where

$$F^m_{i,k} = -F^m_{k,i}, \quad \mathbf{F}^{\mu}_{i,k} = -\mathbf{F}^{\mu}_{k,i} \tag{2.18}$$

are mass and momentum fluxes. In general, these fluxes are supposed to approximate following exact fluxes:

$$F^m_{i,k} \approx \int_{\tilde{z}_i \bigcap z_k} \rho \, dV - \int_{z_i \bigcap \tilde{z}_k} \rho \, dV, \quad \mathbf{F}^{\mu}_{i,k} \approx \int_{\tilde{z}_i \bigcap z_k} \mathbf{u}\rho \, dV - \int_{z_i \bigcap \tilde{z}_k} \mathbf{u}\rho \, dV. \tag{2.19}$$

For derivation of formulas (2.19) interested reader can refer for example to [23]. These formulas take into account the exchange of mass and momentum between cells which share the vertex.

In more traditional methods (see, for example, [23]), one uses simplified version of (2.17), where the exchange of mass and momentum is allowed only between cells that share an face (edge in 2D):

$$\tilde{m}_i = m_i + \sum_{s_{i,k} \in S(z_i)} F^m_{i,k}, \quad \tilde{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_i + \sum_{s_{i,k} \in S(z_i)} \mathbf{F}^{\mu}_{i,k}. \tag{2.20}$$

In this case fluxes correspond to integrals over so-called swept regions [23,13].

For purposes of this paper it does not matter what set of fluxes is used and therefore we will just simply write

$$\tilde{m}_i = m_i + \sum_k F^m_{i,k}, \quad \tilde{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_i + \sum_k \mathbf{F}^{\mu}_{i,k}. \tag{2.21}$$

Conservation of mass and momentum is guaranteed if we use flux form (2.21) and assume that fluxes satisfy (2.18).

Accuracy of remapping depends on how accurate are the fluxes. Let us denote fluxes which produce high-order accuracy by $F^{m,H}$, $\mathbf{F}^{\mu,H}$, and fluxes which produce low-order accuracy by $F^{m,L}$, $\mathbf{F}^{\mu,L}$.

One of the standard ways to obtain intercell fluxes is to first do some reconstruction of functions ρ and $\rho\mathbf{u}$ on the old mesh from their mean values and then use some approximation of exact fluxes (2.19) or similar integrals over the swept regions. For example, low-order fluxes can correspond to piecewise constant reconstruction, and high-order fluxes can correspond to piecewise linear reconstruction [23,13].

2.4. Constraints for fluxes

Having the flux form for remapped mass and momentum we can write constraints related to bounds preservation for density and velocity as follows:

$$m_i^{\min} \leqslant m_i + \sum_k F^m_{i,k} \leqslant m_i^{\max}, \tag{2.22}$$

$$\mathbf{u}_i^{\min}\left(m_i + \sum_k F^m_{i,k}\right) \leqslant \boldsymbol{\mu}_i + \sum_k \mathbf{F}^{\mu}_{i,k} \leqslant \mathbf{u}_i^{\max}\left(m_i + \sum_k F^m_{i,k}\right). \tag{2.23}$$

From the formal point of view it is a global system of linear inequalities with respect to mass and momentum fluxes. It is important to note that inequalities (2.23), which are coming from bounds for velocity, contain both mass and momentum fluxes.

Usually the low-order fluxes are constructed so that inequalities (2.22) and (2.23) are satisfied, which ensures that the system of inequalities has at least one solution. We assume in this paper that low-order fluxes satisfy the density and velocity bounds (2.22) and (2.23).

Our goal is to construct fluxes which are as close as possible to high-order fluxes but still satisfy the system of inequalities (2.22) and (2.23).

Formally this problem can be formulated in different ways. For example, one can formulate optimization problem as follows: find fluxes $F^m_{i,k}$, $\mathbf{F}^{\mu}_{i,k}$ such that in some norm these fluxes are as close as possible to high-order fluxes, that is,

$$\min\left(\kappa_m \|F^m - F^{m,H}\|_m + \kappa_\mu \|\mathbf{F}^{\mu} - \mathbf{F}^{\mu,H}\|_\mu\right), \tag{2.24}$$

where κ_i are some weights, and fluxes satisfy linear constraints (2.22) and (2.23). That is, we can formulate our problem as a global minimization problem with linear constraints. The minimization problem is treated in detail in Section 4.3.

3. FCR for mass-density

In this section we describe flux-corrected remapping (FCR) for mass-density. In general it follows the standard FCT derivation (see, for example, [14]) with some specifics related to remapping. It is not new, however it is instructional and serves as preparation for dealing with the system including density and momentum in the following section.

If we need to remap only mass-density then the goal is to find mass fluxes, $F^m_{i,k}$, such that linear constraints (2.22) are satisfied and these fluxes are as close as possible to high-order fluxes:

$$\min \|F^m - F^{m,H}\|_m. \tag{3.1}$$

An example of possible norm in (3.1) is the L_1 norm

$$\|F^m - F^{m,H}\|^{L_1}_m = \sum_{i,k} \left|F^m_{i,k} - F^{m,H}_{i,k}\right|,$$

where the sum is taken over all faces (i,k) in the grid.

Goal of the FCR is to replace global minimization problem (3.1) and (2.22) by a series of local problems. As we have already mentioned, by doing this we do not solve global optimization problem, but still are able to find fluxes which are more accurate than the low-order fluxes, and which guarantee the preservation of bounds.

First we assume that mass flux is a combination of the low-order and high-order flux:

$$F^m_{i,k} = F^{m,L}_{i,k} + C^m_{i,k} dF^m_{i,k}, \quad dF^m_{i,k} = -dF^m_{k,i} = \left(F^{m,H}_{i,k} - F^{m,L}_{i,k}\right), \tag{3.2}$$

where

$$0 \leqslant C^m_{i,k} = C^m_{k,i} \leqslant 1$$

is some coefficient to be defined, such that if it is equal to zero we recover low-order flux and if it is one we recover high-order flux. $dF^m_{i,k}$ are called antidiffusive fluxes. Our goal is to find $C^m_{k,i}$ such that resulting fluxes guarantee bounds preservation and are as close as possible to high-order fluxes in some sense.

In FCT methodology this problem is replaced by series of simple problems for each face. Such replacement does not give the optimal solution, but bounds are satisfied and flux may be more accurate than low-order flux.

Using (3.2), we can rewrite equation (2.21) for new mass as

$$\tilde{m}_i = \tilde{m}^L_i + \sum_k C^m_{i,k} dF^m_{i,k}, \quad \tilde{m}^L_i = m_i + \sum_k F^{m,L}_{i,k}. \tag{3.3}$$

Then constraints (2.9), resp. (2.22), can be written as

$$m^{\min}_i - \tilde{m}^L_i \leqslant \sum_k C^m_{i,k} dF^m_{i,k} \leqslant m^{\max}_i - \tilde{m}^L_i. \tag{3.4}$$

By assumption the low-order fluxes satisfy the bounds $m^{\min}_i - \tilde{m}^L_i \leqslant 0 \leqslant m^{\max}_i - \tilde{m}^L_i$ and therefore constraints (3.4) always can be satisfied by choosing all $C^m_{i,k}$ equal to zero. These differences are usually denoted by

$$Q^{m,\max}_i = m^{\max}_i - \tilde{m}^L_i \geqslant 0, \quad Q^{m,\min}_i = m^{\min}_i - \tilde{m}^L_i \leqslant 0.$$

Let us now consider the right-hand inequality in (3.4)

$$\sum_k C^m_{i,k} dF^m_{i,k} \leqslant Q^{m,\max}_i. \tag{3.5}$$

The sum on the left-hand side of (3.5) can be subdivided into two sums depending on the sign of $dF^m_{i,k}$ and estimated by the first sum, since the second sum is negative:

$$\sum_k C^m_{i,k} dF^m_{i,k} = \sum_{dF^m_{i,k}>0} C^m_{i,k} dF^m_{i,k} + \sum_{dF^m_{i,k}<0} C^m_{i,k} dF^m_{i,k} \leqslant \sum_{dF^m_{i,k}>0} C^m_{i,k} dF^m_{i,k}. \tag{3.6}$$

So if

$$\sum_{dF^m_{i,k}>0} C^m_{i,k} dF^m_{i,k} \leqslant Q^{m,\max}_i \tag{3.7}$$

is satisfied then (3.5) is satisfied too.

Let us define

$$P^{m,+}_i := \sum_{dF^m_{i,k}>0} dF^m_{i,k} \geqslant 0 \tag{3.8}$$

and first check the special case when $P^{m,+}_i = 0$. In this case both sums in (3.8) and (3.7) are empty since for all faces (i,k) of cell i the antidiffusive fluxes $dF^m_{i,k}$ are non-positive $dF^m_{i,k} \leqslant 0$ and the inequality (3.5) is satisfied because $C^m_{i,k} \geqslant 0,\ dF^m_{i,k} \leqslant 0$ and $Q^{m,\max}_i \geqslant 0$.

Now we can assume that $P_i^{m,+} > 0$ and rewrite (3.7) as

$$\sum_{dF_{i,k}^m > 0} C_{i,k}^m dF_{i,k}^m \leqslant D_i^{m,+} P_i^{m,+}, \tag{3.9}$$

where

$$D_i^{m,+} = Q_i^{m,\max} / P_i^{m,+}. \tag{3.10}$$

Substituting definitions of $P_i^{m,+}$ into (3.9) and moving $D_i^{m,+}$ inside the sum defining $P_i^{m,+}$ we get

$$\sum_{dF_{i,k}^m > 0} C_{i,k}^m dF_{i,k}^m \leqslant \sum_{dF_{i,k}^m > 0} D_i^{m,+} dF_{i,k}^m, \tag{3.11}$$

which is clearly satisfied if

$$C_{i,k}^m \leqslant D_i^{m,+} \quad \text{for } dF_{i,k}^m > 0. \tag{3.12}$$

This is the final constraint on $C_{i,k}^m$ for face (i,k) coming from the upper density-mass bound in cell i.

For the left inequality in (3.4) we can use similar derivation to obtain the expression for $D_i^{m,-}$

$$D_i^{m,-} = Q_i^{m,\min} / P_i^{m,-}, \tag{3.13}$$

where

$$P_i^{m,-} = \sum_{dF_{i,k}^m < 0} dF_{i,k}^m \leqslant 0, \tag{3.14}$$

and the final constraint on $C_{i,k}^m$ coming from the density-mass lower bound

$$C_{i,k}^m \leqslant D_i^{m,-} \quad \text{for } dF_{i,k}^m < 0. \tag{3.15}$$

Note that this constraint is activated only when $dF_{i,k}^m < 0$ in which case $P_i^{m,-} < 0$ so that (3.13) makes sense. If $dF_{i,k}^m \geqslant 0$ then this constraint is not activated.

So being on face (i,k) we look at the sign of $dF_{i,k}^m$ and activate either constraint (3.12) or (3.15) coming either from the upper or lower density-mass bound in cell i. To recall completely: if $dF_{i,k}^m > 0$ then we require $C_{i,k}^m \leqslant D_i^{m,+}$ and if $dF_{i,k}^m < 0$ then we require $C_{i,k}^m \leqslant D_i^{m,-}$ (if $dF_{i,k}^m = 0$ then the bounds in cell i do not require any constraint on $C_{i,k}^m$). However being on face (i,k) we have to incorporate also the constraints coming from the bounds in the second cell k sharing this face.

Clearly if $dF_{i,k}^m > 0$, then $dF_{k,i}^m = -dF_{i,k}^m < 0$ and the corresponding coefficient $C_{i,k}^m = C_{k,i}^m$ has to satisfy two inequalities

$$C_{i,k}^m \leqslant D_i^{m,+}, \quad C_{i,k}^m \leqslant D_k^{m,-},$$

and we can choose any $C_{i,k}^m \leqslant C_{i,k}^{m,\rho}$, where $C_{i,k}^m$ has the upper bound

$$C_{i,k}^{m,\rho} = \min\left(D_i^{m,+}, D_k^{m,-}, 1\right) \quad \text{for } dF_{i,k}^m > 0, \tag{3.16}$$

and similarly

$$C_{i,k}^{m,\rho} = \min\left(D_i^{m,-}, D_k^{m,+}, 1\right) \quad \text{if } dF_{i,k}^m < 0. \tag{3.17}$$

For completeness, we can formally set $C_{i,k}^{m,\rho} = 1$ if $dF_{i,k}^m = 0$, but in this case the final product will be zero anyway. Now if we want to satisfy only density-mass bounds we choose $C_{i,k}^m = C_{i,k}^{m,\rho}$ to be as close as possible to the high-order approximation (for which $C_{i,k}^m = 1$).

In summary, for each face (i,k) we have found coefficients $C_{i,k}^m$ in formulas (3.2) such that constraints (2.22) are satisfied, and resulting mass fluxes are in general more accurate than the low-order fluxes.

The logic of derivation of formulas (3.16) and (3.17) will be used in next section, where we will consider simultaneous remap of density-mass and velocity-momentum.

4. FCR for mass-density and momentum-velocity in 1D

For simplicity of explanation we start with a 1D case when the momentum μ and velocity u are scalars. We will deal with the general case of vector velocities and momenta later. We assume that similarly to mass flux the momentum flux is a combination of the low-order and high-order flux:

$$F_{i,k}^{\mu} = F_{i,k}^{\mu,L} + C_{i,k}^{\mu} dF_{i,k}^{\mu}, \quad dF_{i,k}^{\mu} = -dF_{k,i}^{\mu} = \left(F_{i,k}^{\mu,H} - F_{i,k}^{\mu,L}\right). \tag{4.1}$$

Then constraints (2.23) can be rewritten as

$$u_i^{\min}\tilde{m}_i^L + u_i^{\min}\sum_k C_{i,k}^m dF_{i,k}^m \leqslant \tilde{\mu}_i^L + \sum_k C_{i,k}^\mu dF_{i,k}^\mu \leqslant u_i^{\max}\tilde{m}_i^L + u_i^{\max}\sum_k C_{i,k}^m dF_{i,k}^m, \tag{4.2}$$

where $\tilde{\mu}_i^L$ is defined similarly to $\tilde{m}_i^L$ in (3.3):

$$\tilde{\mu}_i^L = \mu_i + \sum_k F_{i,k}^{\mu,L}.$$

It is clear that constraints (4.2) form a coupled system of linear inequalities with respect to coefficients $C_{i,k}^m,\ C_{i,k}^\mu$ (all other quantities are known numbers). Remember that this system has to be solved together with inequalities (3.4) arising from mass-density constraints.

Let us consider the right-hand inequality in (4.2)

$$\tilde{\mu}_i^L + \sum_k C_{i,k}^\mu dF_{i,k}^\mu \leqslant u_i^{\max}\tilde{m}_i^L + u_i^{\max}\sum_k C_{i,k}^m dF_{i,k}^m \tag{4.3}$$

and move terms with C's to left-hand side and other terms to right-hand side:

$$\sum_k C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\max}\sum_k C_{i,k}^m dF_{i,k}^m \leqslant u_i^{\max}\tilde{m}_i^L - \tilde{\mu}_i^L. \tag{4.4}$$

Because we assume that the low-order fluxes satisfy the bounds, the expression $u_i^{\max}\tilde{m}_i^L - \tilde{\mu}_i^L \geqslant 0$ on the right-hand side is non-negative for any cell i, and if all C's are zero, then the inequality (4.4) is satisfied (similarly the left inequality of (4.2) is also satisfied), that is there is at least one solution for (4.4). Our goal is to find another solution (if it exists) where C's are closer to one.

4.1. The method by Schär and Smolarkiewicz

There are several possible approaches to deal with system (4.4). For example, in [25] (paper which in some sense inspired our work) authors do the following. They analyze inequality (4.4) for cell i depending on the sign of $u_i^{\max}$. Let us assume that $u_i^{\max} > 0$. Then we can estimate the left-hand side of (4.4) from above as follows

$$\sum_k C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\max}\sum_k C_{i,k}^m dF_{i,k}^m \leqslant \sum_{dF_{i,k}^\mu > 0} C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\max}\sum_{dF_{i,k}^m < 0} C_{i,k}^m dF_{i,k}^m, \tag{4.5}$$

where we have removed from the left-hand side of (4.5) all terms which are negative or zero ($C_{i,k}^m \geqslant 0, C_{i,k}^\mu \geqslant 0, u_i^{\max} > 0$). Now if we find $C_{i,k}^m$ and $C_{i,k}^\mu$ such that

$$\sum_{dF_{i,k}^\mu > 0} C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\max}\sum_{dF_{i,k}^m < 0} C_{i,k}^m dF_{i,k}^m \leqslant Q_i^{\mu,\max}, \tag{4.6}$$

where

$$Q_i^{\mu,\max} := u_i^{\max}\tilde{m}_i^L - \tilde{\mu}_i^L \geqslant 0, \tag{4.7}$$

then (4.4) is satisfied for these $C_{i,k}^m$ and $C_{i,k}^\mu$.

We define

$$P_i^{\mu,u^{\max}>0,+} := \sum_{dF_{i,k}^\mu > 0} dF_{i,k}^\mu - u_i^{\max}\sum_{dF_{i,k}^m < 0} dF_{i,k}^m \geqslant 0. \tag{4.8}$$

$P_i^{\mu,u^{\max}>0,+}$ is zero if and only if both sums are empty. Then also both sums on the left-hand side of (4.6) are empty, the left-hand side is zero and (4.6) is satisfied trivially as $Q_i^{\mu,\max} \geqslant 0$.

Assuming $P_i^{\mu,u^{\max}>0,+} > 0$, we rewrite (4.6) as

$$\sum_{dF_{i,k}^\mu > 0} C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\max}\sum_{dF_{i,k}^m < 0} C_{i,k}^m dF_{i,k}^m \leqslant D_i^{\mu,u^{\max}>0,+} P_i^{\mu,u^{\max}>0,+}, \tag{4.9}$$

where

$$D_i^{\mu,u^{\max}>0,+} = Q_i^{\mu,\max}/P_i^{\mu,u^{\max}>0,+}. \tag{4.10}$$

Substituting definitions of $P_i^{\mu,u^{\max}>0,+}$ into (4.9) and moving $D_i^{\mu,u^{\max}>0,+}$ inside the two sums, we get

$$\sum_{dF_{i,k}^\mu > 0} C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\max}\sum_{dF_{i,k}^m < 0} C_{i,k}^m dF_{i,k}^m \leqslant \sum_{dF_{i,k}^\mu > 0} D_i^{\mu,u^{\max}>0,+} dF_{i,k}^\mu - u_i^{\max}\sum_{dF_{i,k}^m < 0} D_i^{\mu,u^{\max}>0,+} dF_{i,k}^m, \tag{4.11}$$

which is clearly satisfied if

$$C^{\mu}_{i,k} \leqslant D^{\mu,u^{\max}>0,+}_{i} \quad \text{for } dF^{\mu}_{i,k} > 0, \tag{4.12}$$

$$C^{m}_{i,k} \leqslant D^{\mu,u^{\max}>0,+}_{i} \quad \text{for } dF^{m}_{i,k} < 0. \tag{4.13}$$

Note that $D^{\mu,u^{\max}>0,+}_{i}$ is the upper bound for $C^{\mu}_{i,k}$ for face (i,k) such that $dF^{\mu}_{i,k} > 0$ and for $C^{m}_{i,k}$ such that $dF^{m}_{i,k} < 0$, assuming $u^{\max}_i > 0$. In general it may be different sets of faces which are actually involved by a nonzero contribution. Other C's are only required to be positive with respect to constraint (4.4).

Similar formulas can be obtained assuming that $u^{\max}_i < 0$ (or $u^{\max}_i = 0$) and for the left inequality in (4.2) depending on the assumption of the sign of $u^{\min}_i$. Clearly final values of the $C^{\mu}_{i,k},\ C^{m}_{i,k}$ will depend on active constraints from both cells sharing face (i,k). As all the constructed active constraints have the form of the upper bound on $C^{m}_{i,k}$ or $C^{\mu}_{i,k}$ and all constraints have to hold simultaneously, the resulting set of admissible values of $C^{m}_{i,k}$ and $C^{\mu}_{i,k}$ in the $(C^{m}_{i,k}, C^{\mu}_{i,k})$ plane will be a rectangle (see examples in Fig. 5).

We refer the interested reader to the original paper [25] for details of their method which is there developed also in 2D (our presentation here can be extended also to multiple dimensions). In this paper we will use this method as a baseline for comparison with the new method which we describe next.

4.2. Synchronized FCR (SFCR) in 1D

We first consider the right-hand side constraint of (4.2) written as (4.4), i.e. the constraint containing the upper bound for velocity $u^{\max}$. Let us denote the left-hand side of (4.4) by $S^{\mu,\max}_i$ and group the terms on the left-hand side by faces

$$S^{\mu,\max}_i := \sum_k \left(C^{\mu}_{i,k} dF^{\mu}_{i,k} - u^{\max}_i C^{m}_{i,k} dF^{m}_{i,k} \right) \leqslant u^{\max}_i \tilde{m}^{L}_i - \tilde{\mu}^{L}_i. \tag{4.14}$$

In general, the terms in the sum corresponding to the particular face (i,k) in (4.14) can be positive or negative.

We define the "limited" contribution to the sum $S^{\mu,\max}_i$ from face (i,k) appearing in (4.14)

$$\Phi^{i,\mu,\max}_{i,k} := C^{\mu}_{i,k} dF^{\mu}_{i,k} - u^{\max}_i C^{m}_{i,k} dF^{m}_{i,k} \tag{4.15}$$

and the "unlimited" contribution

$$\Psi^{i,\mu,\max}_{i,k} := dF^{\mu}_{i,k} - u^{\max}_i dF^{m}_{i,k}. \tag{4.16}$$

Both $\Phi^{i,\mu,\max}_{i,k}$ and $\Psi^{i,\mu,\max}_{i,k}$ are related to cell i denoted by the superscript i coming from velocity $u^{\max}_i$ and to face (i,k) denoted by subscripts (i,k) and coming from the antidiffusive fluxes $dF^{m}_{i,k}$ and $dF^{\mu}_{i,k}$.

To follow the logic of the FCR derivation for density-mass in Section 3, we split the sum $S^{\mu,\max}_i$ in (4.14) according to the signs of the "unlimited" contributions $\Psi^{i,\mu,\max}_{i,k}$ (4.16) and rewrite (4.14) as

$$S^{\mu,\max,+}_i + S^{\mu,\max,-}_i \leqslant Q^{\mu,\max}_i, \tag{4.17}$$

where $Q^{\mu,\max}_i = u^{\max}_i \tilde{m}^{L}_i - \tilde{\mu}^{L}_i \geqslant 0$ was defined in (4.7) and

$$S^{\mu,\max,+}_i := \sum_{\Psi^{i,\mu,\max}_{i,k} > 0} \Phi^{i,\mu,\max}_{i,k}, \quad S^{\mu,\max,-}_i := \sum_{\Psi^{i,\mu,\max}_{i,k} \leqslant 0} \Phi^{i,\mu,\max}_{i,k}. \tag{4.18}$$

Now if we find $C^{m}_{i,k}$ and $C^{\mu}_{i,k}$ so that

$$S^{\mu,\max,-}_i \leqslant 0, \tag{4.19a}$$

$$S^{\mu,\max,+}_i \leqslant Q^{\mu,\max}_i, \tag{4.19b}$$

then (4.17) holds.

Let us first look at the inequality (4.19a)

$$S^{\mu,\max,-}_i = \sum_{\Psi^{i,\mu,\max}_{i,k} \leqslant 0} \Phi^{i,\mu,\max}_{i,k} \leqslant 0. \tag{4.20}$$

Clearly it is satisfied if we require that

$$\Phi^{i,\mu,\max}_{i,k} = C^{\mu}_{i,k} dF^{\mu}_{i,k} - u^{\max}_i C^{m}_{i,k} dF^{m}_{i,k} \leqslant 0 \quad \text{for } \Psi^{i,\mu,\max}_{i,k} \leqslant 0. \tag{4.21}$$

So on faces where the unlimited contribution $\Psi^{i,\mu,\max}_{i,k}$ is non-positive we require that the limited contribution $\Psi^{i,\mu,\max}_{i,k}$ is also non-positive, which gives some of the conditions which have to be satisfied by $C^{m}_{i,k}$ and $C^{\mu}_{i,k}$. The inequality (4.21) is not equivalent to (4.20), it is only a sufficient condition for (4.20), i.e. (4.21) implies (4.20). The inequality (4.21) is the local constraint on the face (i,k) which is obtained from maximum bound condition in cell i in case of $\Psi^{i,\mu,\max}_{i,k} \leqslant 0$. Note that the only unknowns in the linear inequality (4.21) are $C^{m}_{i,k}$ and $C^{\mu}_{i,k}$, all other quantities including $dF^{\mu}_{i,k},\ dF^{m}_{i,k},\ u^{\max}_i,\ \Psi^{i,\mu,\max}_{i,k}$ are known numbers at this stage.

Before dealing with the second inequality (4.19b) we define

$$P_i^{\mu,\max,+} := \sum_{\Psi_{i,k}^{i,\mu,\max}>0} \Psi_{i,k}^{i,\mu,\max} \geqslant 0.$$

$P_i^{\mu,\max,+}$ is zero if and only if the sum is empty, which means that $\Psi_{i,k}^{i,\mu,\max} \leqslant 0$ for all faces (i,k) of cell i. So if $P_i^{\mu,\max,+} = 0$ then also $S_i^{\mu,\max,+} = 0$ as the sum which defines it in (4.18) is also empty. As $Q_i^{\mu,\max} \geqslant 0$ (the low-order remapping satisfies the bounds), the inequality (4.19b) is satisfied in this case, and for all faces (i,k) of cell i the constraints (4.21) (which are the only constraints coming from the maximum bound in cell i) on $C_{i,k}^m$ and $C_{i,k}^\mu$ will be activated.

Assuming $P_i^{\mu,\max,+} > 0$, the second inequality (4.19b) is rewritten as

$$S_i^{\mu,\max,+} \leqslant D_i^{\mu,\max,+} P_i^{\mu,\max,+}, \tag{4.22}$$

where and $D_i^{\mu,\max,+}$ is given by

$$D_i^{\mu,\max,+} = Q_i^{\mu,\max}/P_i^{\mu,\max,+}. \tag{4.23}$$

Substituting definitions of $S_i^{\mu,\max,+}$ and $P_i^{\mu,\max,+}$ into (4.22) we get

$$\sum_{\Psi_{i,k}^{i,\mu,\max}>0} \Phi_{i,k}^{i,\mu,\max} \leqslant D_i^{\mu,\max,+} \sum_{\Psi_{i,k}^{i,\mu,\max}>0} \Psi_{i,k}^{i,\mu,\max}, \tag{4.24}$$

which is clearly satisfied if

$$\Phi_{i,k}^{i,\mu,\max} \leqslant D_i^{\mu,\max,+} \Psi_{i,k}^{i,\mu,\max} \quad \text{for } \Psi_{i,k}^{i,\mu,\max} > 0. \tag{4.25}$$

The inequality (4.25) is not equivalent to (4.22), it is only a sufficient condition for (4.22), i.e. (4.25) implies (4.22). The inequality (4.25) is the local constraint on face (i,k) which is obtained from maximum bound condition in cell i in case of $\Psi_{i,k}^{i,\mu,\max} > 0$. The condition (4.25) rewritten in the form

$$C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\max} C_{i,k}^m dF_{i,k}^m \leqslant \frac{Q_i^{\mu,\max}}{P_i^{\mu,\max,+}} \Psi_{i,k}^{i,\mu,\max} \quad \text{for } \Psi_{i,k}^{i,\mu,\max} > 0 \tag{4.26}$$

gives us the constraint on $C_{i,k}^m$ and $C_{i,k}^\mu$ on faces (i,k) where the contribution $\Psi_{i,k}^{i,\mu,\max}$ is positive. Again the only unknowns in the linear inequality (4.26) are $C_{i,k}^m$ and $C_{i,k}^\mu$, all other quantities including $dF_{i,k}^\mu,\ dF_{i,k}^m,\ u_i^{\max},\ \Psi_{i,k}^{i,\mu,\max},\ Q_i^{\mu,\max},\ P_i^{\mu,\max,+}$ are known numbers at this stage.

To summarize, we derived the linear inequalities ((4.21) or (4.26) depending on the sign of $\Psi_{i,k}^{i,\mu,\max}$) for $C_{i,k}^m$ and $C_{i,k}^\mu$ on the particular face (i,k) which are sufficient for satisfying the maximum bound in velocity (the right inequality of (4.2)) in cell i. In the following we are going to derive in the same manner the constraints which will be sufficient for satisfying the minimum bound in velocity.

The process for the constraint with the lower velocity bound $u^{\min}$, i.e. the left inequality of (4.2) resp. (2.23), is a full analogy of the above process for the upper velocity bound: Let us move the terms with C's to the left-hand side and other terms to the right-hand side and further denote the left-hand side by $S_i^{\mu,\min}$:

$$S_i^{\mu,\min} := \sum_k \left(C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\min} C_{i,k}^m dF_{i,k}^m \right) \geqslant u_i^{\min} \tilde{m}_i^L - \tilde{\mu}_i^L, \tag{4.27}$$

where we have grouped terms on the left-hand side by faces. In general, the terms in the sum corresponding to the particular face (i,k) in (4.27) can be positive or negative.

We define the "limited" contribution to the sum $S_i^{\mu,\min}$ from face (i,k) appearing in (4.27)

$$\Phi_{i,k}^{i,\mu,\min} := C_{i,k}^\mu dF_{i,k}^\mu - u_i^{\min} C_{i,k}^m dF_{i,k}^m \tag{4.28}$$

and the "unlimited" contribution

$$\Psi_{i,k}^{i,\mu,\min} := dF_{i,k}^\mu - u_i^{\min} dF_{i,k}^m, \tag{4.29}$$

which actually corresponds to the "limited" contribution $\Phi_{i,k}^{i,\mu,\min}$ (4.28) with $C_{i,k}^\mu = C_{i,k}^m = 1$.

Again we split the sum $S_i^{\mu,\min}$ in (4.27) according to the signs of the "unlimited" contributions $\Psi_{i,k}^{i,\mu,\min}$ (4.29) and rewrite (4.27) as

$$S_i^{\mu,\min,+} + S_i^{\mu,\min,-} \geqslant Q_i^{\mu,\min}, \tag{4.30}$$

where $Q_i^{\mu,\min} = u_i^{\min} \tilde{m}_i^L - \tilde{\mu}_i^L \leqslant 0$ and

$$S_i^{\mu,\min,+} := \sum_{\Psi_{i,k}^{i,\mu,\min}\geqslant 0} \Phi_{i,k}^{i,\mu,\min}, \quad S_i^{\mu,\min,-} := \sum_{\Psi_{i,k}^{i,\mu,\min}<0} \Phi_{i,k}^{i,\mu,\min}. \tag{4.31}$$

Now if we find $C^m_{i,k}$ and $C^\mu_{i,k}$ so that

$$S_i^{\mu,\min,+} \geqslant 0, \tag{4.32a}$$
$$S_i^{\mu,\min,-} \geqslant Q_i^{\mu,\min}, \tag{4.32b}$$

then (4.30) holds.

Let us first look at the inequality (4.32a)

$$S_i^{\mu,\min,+} = \sum_{\Psi^{i,\mu,\min}_{i,k} \geqslant 0} \Phi^{i,\mu,\min}_{i,k} \geqslant 0. \tag{4.33}$$

Clearly it is satisfied if we require that

$$\Phi^{i,\mu,\min}_{i,k} = C^\mu_{i,k} dF^\mu_{i,k} - u_i^{\min} C^m_{i,k} dF^m_{i,k} \geqslant 0 \quad \text{for } \Psi^{i,\mu,\min}_{i,k} \geqslant 0. \tag{4.34}$$

This is the linear inequality in $C^m_{i,k}$ and $C^\mu_{i,k}$ for the case $\Psi^{i,\mu,\min}_{i,k} \geqslant 0$.

Before dealing with the second inequality (4.32b) we define

$$P_i^{\mu,\min,-} := \sum_{\Psi^{i,\mu,\min}_{i,k} < 0} \Psi^{i,\mu,\min}_{i,k} \leqslant 0.$$

$P_i^{\mu,\min,-}$ is zero if and only if the sum is empty, which means that $\Psi^{i,\mu,\min}_{i,k} \geqslant 0$ for all faces (i,k) of cell i. So if $P_i^{\mu,\min,-} = 0$ then also $S_i^{\mu,\min,-} = 0$ as the sum which defines it in (4.31) is also empty. As $Q_i^{\mu,\min} \leqslant 0$ (the low-order remapping satisfies the bounds) the inequality (4.32b) is satisfied in this case and for all faces (i,k) of cell i the constraints (4.34) (which are the only constraints coming from the minimum bound) on $C^m_{i,k}$ and $C^\mu_{i,k}$ will be activated.

Assuming $P_i^{\mu,\min,-} < 0$, the second inequality (4.32b) is rewritten as

$$S_i^{\mu,\min,-} \geqslant D_i^{\mu,\min,-} P_i^{\mu,\min,-}, \tag{4.35}$$

where $D_i^{\mu,\min,-}$ is given by

$$D_i^{\mu,\min,-} = Q_i^{\mu,\min} / P_i^{\mu,\min,-}. \tag{4.36}$$

Substituting definitions of $S_i^{\mu,\min,-}$ and $P_i^{\mu,\min,-}$ into (4.35) we get

$$\sum_{\Psi^{i,\mu,\min}_{i,k} < 0} \Phi^{i,\mu,\min}_{i,k} \geqslant D_i^{\mu,\min,-} \sum_{\Psi^{i,\mu,\min}_{i,k} < 0} \Psi^{i,\mu,\min}_{i,k}, \tag{4.37}$$

which is clearly satisfied if

$$\Phi^{i,\mu,\min}_{i,k} \geqslant D_i^{\mu,\min,-} \Psi^{i,\mu,\min}_{i,k} \quad \text{for } \Psi^{i,\mu,\min}_{i,k} < 0. \tag{4.38}$$

The previous condition rewritten in the form

$$C^\mu_{i,k} dF^\mu_{i,k} - u_i^{\min} C^m_{i,k} dF^m_{i,k} \geqslant \frac{Q_i^{\mu,\min}}{P_i^{\mu,\min,-}} \Psi^{i,\mu,\min}_{i,k} \quad \text{for } \Psi^{i,\mu,\min}_{i,k} < 0 \tag{4.39}$$

gives us the constraint on $C^m_{i,k}$ and $C^\mu_{i,k}$ on faces (i,k) where the contribution $\Psi^{i,\mu,\min}_{i,k}$ is negative. Again, the only unknowns in the linear inequality (4.39) are $C^m_{i,k}$ and $C^\mu_{i,k}$.

To summarize we combine here the linear inequalities for $C^m_{i,k}$ and $C^\mu_{i,k}$ (on face (i,k)) sufficient for upper (4.21) and (4.26) and lower (4.34) and (4.39) velocity bounds in cell i into one formula for each bound:

$$C^\mu_{i,k} dF^\mu_{i,k} - u_i^{\max} C^m_{i,k} dF^m_{i,k} \leqslant \frac{Q_i^{\mu,\max}}{P_i^{\mu,\max,+}} \max\left(\Psi^{i,\mu,\max}_{i,k}, 0\right), \tag{4.40a}$$

$$C^\mu_{i,k} dF^\mu_{i,k} - u_i^{\min} C^m_{i,k} dF^m_{i,k} \geqslant \frac{Q_i^{\mu,\min}}{P_i^{\mu,\min,-}} \min(\Psi^{i,\mu,\min}_{i,k}, 0). \tag{4.40b}$$

This gives us on face (i,k) one inequality sufficient for maximum velocity bound and one inequality sufficient for minimum bound in the cell i. To get the full set of constraints on $C^m_{i,k}$ and $C^\mu_{i,k}$ for the particular face (i,k), we need to apply both maximum and minimum constraints also in the second cell k sharing face (k,i) (for which $C^m_{k,i} = C^m_{i,k}$ and $C^\mu_{k,i} = C^\mu_{i,k}$). So on each face we will have the system of four linear inequalities for $C^m_{i,k}$ and $C^\mu_{i,k}$ which guarantee the preservation of velocity bounds in both neighboring cells.

The additional inequality $C^m_{i,k} \leqslant C^{m,\rho}_{i,k}$, where the upper bound $C^{m,\rho}_{i,k}$ is given by (3.16) and (3.17), is coming from the density bound (3.4) and from the basic assumptions on $C^m_{i,k}$ and $C^\mu_{i,k}$ we have $0 \leqslant C^m_{i,k}$ and $0 \leqslant C^\mu_{i,k} \leqslant 1$. So on face (i,k) the admissible set of $C^m_{i,k}$ and $C^\mu_{i,k}$ is defined by potentially as many as eight linear inequalities in $C^m_{i,k}$ and $C^\mu_{i,k}$. Thus the admissible set is a convex polygon defined by intersection of eight half-planes in $(C^m_{i,k}, C^\mu_{i,k})$ plane (each half-plane is defined by one inequality).

We denote this set $A_{i,k}$ and clearly $A_{i,k} = A_{k,i}$. Both density and velocity bounds are satisfied by remapping solution obtained from any pair $(C^m_{i,k}, C^\mu_{i,k})$ from the admissible set. As we know that low-order remapping satisfies both bounds, we know that the point $(C^m_{i,k}, C^\mu_{i,k}) = (0,0)$ belongs into the admissible set $A_{i,k}$ and therefore the set is not empty.

4.3. Optimal choice of C's

In the previous section we have constructed for each face (i,k) the admissible set $A_{i,k}$ of $C^m_{i,k}$ and $C^\mu_{i,k}$ in the $(C^m_{i,k}, C^\mu_{i,k})$ plane. Any $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ from the admissible set gives the solution which satisfies the bounds in density and velocity. The admissible set $A_{i,k}$ is a convex polygon which is a subset of $(0,1) \times (0,1)$ and contains the origin, i.e. $(0,0) \in A_{i,k}$. Now we need to decide which pair $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ to use for remapping.

For the method by Schär and Smolarkiewicz described in Section 4.1 the admissible set is a rectangle in the $(C^m_{i,k}, C^\mu_{i,k})$ plane and their natural choice is the upper right corner of the admissible set for which both C's attain the maximum values on the admissible set.

Our original wish was to minimize the deviations of the new density $\tilde{\rho}$ and velocity $\tilde{u}$ from their high-order approximations (denoted by superscript H) computed by high-order fluxes, that is,

$$\min\left(\kappa_\rho\|\tilde{\rho} - \tilde{\rho}^H\|_\rho + \kappa_u\|\tilde{u} - \tilde{u}^H\|_u\right), \tag{4.41}$$

where the minimization goes over all $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ on all faces (i,k) of the grid and where $\kappa_\rho,\ \kappa_u$ are some suitable weights. Substituting the formulas for the new, remapped $\tilde{\rho}, \tilde{u}$ we see that this is a global minimization problem, for which we do not know how to transform it into a local one. The FCT approach allowed us to make bounds local and we want to choose the $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ also locally. So instead of global minimization (4.41) we choose to minimize the sum of relative (w.r.t. the old values) deviations of the new mass density $\tilde{\rho}$ and momentum density $\tilde{n} = \tilde{\rho}\tilde{u}$ from their high-order approximations

$$\min\left(\left\|\frac{\tilde{\rho} - \tilde{\rho}^H}{\rho}\right\|_\rho + \left\|\frac{\tilde{n} - \tilde{n}^H}{n}\right\|_n\right), \tag{4.42}$$

and we are going to show that this global minimization (again over all $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ on all faces (i,k) of the grid) can be bounded from above by local minimization problems on all faces (i,k). The velocity and momentum density can be zero. To avoid division by zero in the second norm in (4.42) we add below in (4.44) a small positive quantity to the denominator to make it always positive. Since density is assumed to be positive everywhere, similar problem does not appear for the first norm in (4.42).

Now we choose the L_1 norm for both norms in (4.42) and we are minimizing

$$E^{L_1} = \sum_i \left(\frac{|\tilde{\rho}_i - \tilde{\rho}^H_i|}{\rho_i} + \frac{|\tilde{n}_i - \tilde{n}^H_i|}{|n_i|}\right) V(\tilde{z}_i), \tag{4.43}$$

which can be rewritten as

$$E^{L_1} = \sum_i \left(\frac{|\tilde{m}_i - \tilde{m}^H_i|}{\rho_i} + \frac{|\tilde{\mu}_i - \tilde{\mu}^H_i|}{|n_i|}\right) = \sum_i \left(\left|\sum_{i,k} \frac{F^m_{i,k} - F^{m,H}_{i,k}}{\rho_i}\right| + \left|\sum_{i,k} \frac{F^\mu_{i,k} - F^{\mu,H}_{i,k}}{n_i}\right|\right).$$

The latest form of E^{L_1} can be estimated by

$$E^{L_1} \leqslant \sum_i \sum_{i,k} \left(\frac{|F^m_{i,k} - F^{m,H}_{i,k}|}{\rho_i} + \frac{|F^\mu_{i,k} - F^{\mu,H}_{i,k}|}{|n_i|}\right) = \sum_{i,k}\left((1 - C^m_{i,k})|dF^m_{i,k}|\left(\frac{1}{\rho_i} + \frac{1}{\rho_k}\right) + \left(1 - C^\mu_{i,k}\right)|dF^\mu_{i,k}|\left(\frac{1}{|n_i|} + \frac{1}{|n_k|}\right)\right),$$

where now the sum in the last expression runs over all faces (i,k) of the grid. The global minimum of E^{L_1} (4.43) over all $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ on all faces (i,k) of the grid is estimated from above by the sum of local minima on each face (i,k):

$$\min_{(i,k),(C^m_{i,k},C^\mu_{i,k})\in A_{i,k}} (E^{L_1}) \leqslant \sum_{i,k} \min_{\left(C^m_{i,k},C^\mu_{i,k}\right)\in A_{i,k}} E^{L_1}_{i,k},$$

where the minimized local deviations $E^{L_1}_{i,k}$ are given by

$$E^{L_1}_{i,k} = \left(1 - C^m_{i,k}\right)\left|dF^m_{i,k}\right|\left(\frac{1}{\rho_i} + \frac{1}{\rho_k}\right) + \left(1 - C^\mu_{i,k}\right)\left|dF^\mu_{i,k}\right|\left(\frac{1}{|n_i| + \epsilon} + \frac{1}{|n_k| + \epsilon}\right) \tag{4.44}$$

and the local minimizations are going over $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ on each particular face (i,k). As mentioned before, to avoid division by zero we added to $|n|$ a suitable small quantity ϵ (density is assumed to be positive). If the mesh has N_f faces then the original minimization of (4.43) over all $2N_f$ parameters $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ is replaced by N_f local minimizations of (4.44) over two particular parameters $(C^m_{i,k}, C^\mu_{i,k}) \in A_{i,k}$ defined for face (i,k).

The minimum of linear objective function $E^{L_1}_{i,k}$ (4.44) over the (convex polygonal) admissible set $A_{i,k}$ is achieved on the boundary of the polygon $A_{i,k}$. In the special case case of non-unique minimum when the minimum of objective function is achieved on one edge of polygon $A_{i,k}$ (i.e. when the constraint corresponding to this edge is parallel to the isolines of the objective function $E^{L_1}_{i,k}$), we choose the pair of $C^m_{i,k}$ and $C^\mu_{i,k}$ which corresponds to middle point of corresponding segment. In our examples we have not experienced any problems due to this special case (it is rather rare), however in general it can cause discontinuity in fluxes dependence on the data (remapped density and momentum and change of mesh).

If the non-uniqueness would pose a problem, one can change the norm in (4.42) from L_1 to L_2 norm, which would (by a similar procedure as above) lead to local minimizations of quadratic objective functions $E^{L_2}_{i,k}$ instead of linear ones (4.44) as done in [26].

Note that as the admissible set is a polygon with less than nine vertices we do not use any optimization routine and simply test the values of objective functions at all vertices of the admissible set to find minimum. Before constructing the admissible set by half-planes intersections we test if the point $(C^m_{i,k}, C^\mu_{i,k}) = (C^{m,\rho}_{i,k}, 1)$ (where the bound $C^{m,\rho}_{i,k}$ is given by (3.16) and (3.17)) satisfies all the constraints and if it does then this point is clearly solution of our minimization problem.

5. Numerical results in 1D

In this section we present several numerical tests showing the performance of our SFCR remapping method in 1D scalar case. For low-order approximation we use donor fluxes based on piecewise constant reconstruction on the old grid. For high-order approximation we use remapping fluxes based on piecewise linear reconstruction of the conserved quantities ρ and $n = \rho u$ on the old mesh, where the slopes of linear reconstruction inside one cell are given by centered differences.

5.1. Demonstrative simple example

We start here with a very simple demonstrative example. Our computational domain $\Omega = [0, 8]$ is divided into eight equidistant cells giving the initial mesh. The density and velocity in the cells are given by discretization of functions

$$\rho(x) = x + 10 \quad \text{for } 0 \leqslant x \leqslant 8, \quad u(x) = \begin{cases} x & \text{for } 0 \leqslant x \leqslant 4 \\ -x & \text{for } 4 < x \leqslant 8 \end{cases}. \tag{5.1}$$

The new mesh is obtained from the old one by moving all seven internal nodes (at $x = 1, 2, \ldots, 7$) to the left by $\delta x = -0.2$. The old original mesh and the new mesh to which we remap are presented in Fig. 3, where for easier description five cells of interest are marked by letters $i,\ j,\ k,\ l,\ m$.

The resulting density and velocity remapped by low-order, high-order and SFCR methods, together with original values on the old mesh and bounds for density and velocity, are presented in Fig. 4. The bounds in density and velocity are depicted by gray rectangles in each old cell. Old cells are separated by black vertical lines and new cells by red vertical lines. Density is linear and the old mesh is uniform, so all three methods produce the same result for density and there are no problems with density bounds. The high-order method however produces an overshoot (violates maximum velocity bound) in cell k (being the interval [2.8, 3.8] in the new mesh). Our SFCR method satisfies the bounds and is closer to the high-order solution than the low-order method. On all interfaces (mesh nodes) except (j, k) and (k, l) the high-order fluxes satisfy all bounds, so that at these interfaces we can use $C^m = C^\mu = 1$. We look in more detail on the situation at (j, k) and (k, l) interfaces of cell k, where the high-order method violates the upper bound.

The admissible sets in (C^m, C^μ) plane constructed by our SFCR and Schär and Smolarkiewicz [25] (further referred to as S&S) methods with our and S&S final C's on (j, k) interface at $x \approx 3$ and on (k, l) interface at $x \approx 4$ are shown in Fig. 5. S&S admissible sets are rectangles while SFCR admissible sets are convex polygons. On the (j, k) interface the S&S admissible set is the rectangle with lower left corner (0, 0) and upper right corner (1.0, 0.4682), while the SFCR admissible set is the quadrilateral with vertices (0, 0), (1.0, 0.2333), (1.0, 0.6121), (0.0, 0.3788), with solutions (1.0, 0.4682) for S&S and (1.0, 0.6121) for SFCR method. On the (k, l) interface the S&S admissible set is the rectangle with lower left corner (0, 0) and upper right corner (0.4682, 0.4682), while the SFCR admissible set is the quadrilateral with vertices (0, 0), (1.0, 0.0933), (1.0, 0.4574), (0.0, 0.5299), with solutions (0.4682, 0.4682) for S&S and (1.0, 0.4574) for SFCR method. On both interfaces the SFCR admissible set is bigger than S&S admissible set in a sense, that we can choose C's closer to the high-order values $C^m = C^\mu = 1$ (we have numerically observed this property also for other problems, however we have no proof of this property and we cannot claim that there is not a problem for which S&S method would give better C's closer to the high-order values $C^m = C^\mu = 1$). The intersection of S&S and SFCR admissible sets is always non-empty as the point (0, 0) belongs

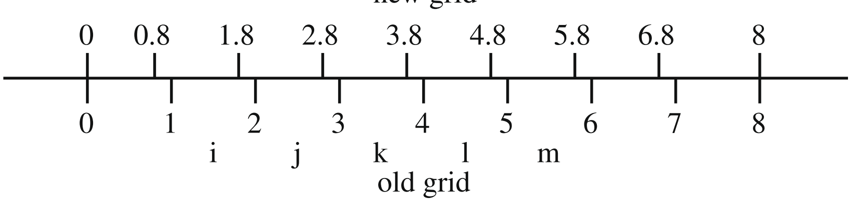

Fig. 3. Meshes for demonstrative 1D example: old mesh before remap and new mesh to which we remap. Five cells are marked by $i,\ j,\ k,\ l,\ m$.

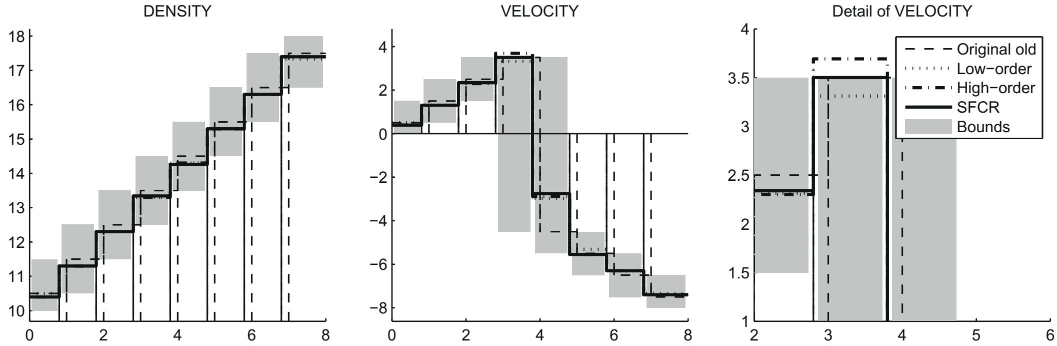

Fig. 4. Remapped density and velocity by low-order, high-order and our SFCR methods, together with original values on the old mesh and bounds for density and velocity.

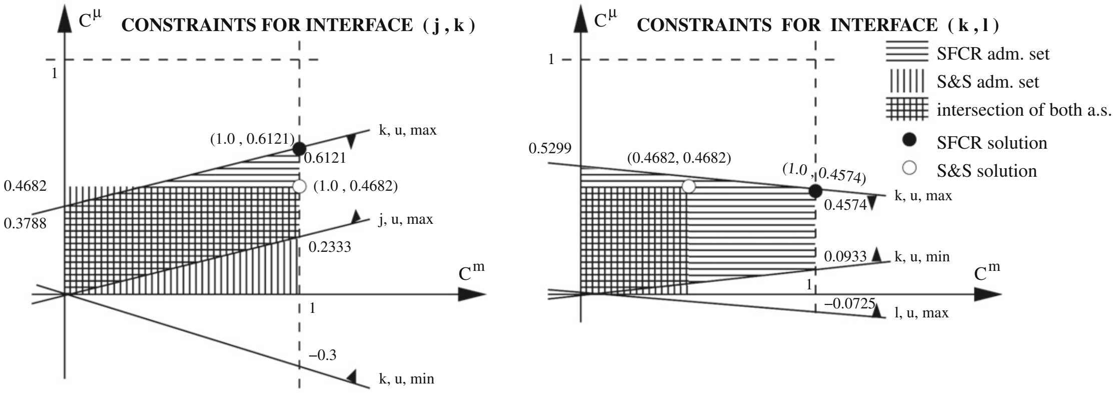

Fig. 5. Admissible sets in (C^m, C^μ) plane by our SFCR and Schär and Smolarkiewicz (S&S) methods with our and S&S final C's at (j,k) interface at $x \approx 3$ and (k,l) interface at $x \approx 4$. Different patterns denote SFCR, S&S admissible sets and their intersections. The lines with small triangles describe half-planes coming from bounds constraints, with lines labeled by constraints, i.e. the lines labeled $k,\ u,\ $ max result from maximum bounds in the cell k.

to both of them. On both (j,k) and (k,l) interfaces both admissible set differences ($A^{\text{S\&S}} - A^{\text{SFCR}}$ and $A^{\text{SFCR}} - A^{\text{S\&S}}$) are non-empty, i.e. one is not a subset of the other, part of the S&S admissible set does not belong into the SFCR admissible set and vice versa.

If we would include density and velocity results in the plots in Fig. 4, one would not be able to distinguish them as they are close to SFCR results for this oversimplified remapping problem. However we believe that Fig. 5 provides an insight into the question how the SFCR and S&S methods differ. We repeat that SFCR and S&S methods provide two different sufficient conditions on C's under which the remapped values satisfy all bounds.

5.2. Cyclic remapping

To compare properties of the SFCR and S&S methods, let us now present the results of two cyclic remapping tests from [22]. In both of them, we generate a sequence of meshes on the domain $[x_{\min}, x_{\max}]$, using the analytical function

$$\begin{aligned} x(\xi,t) &= x_{\min} + (x_{\max} - x_{\min})\tilde{\xi}(\xi,t), \\ \tilde{\xi}(\xi,t) &= [1-\alpha(t)]\xi + \alpha(t)\xi^3, \\ \alpha(t) &= \frac{\sin(4\pi t)}{2} \end{aligned}$$

for $0 \leqslant \xi \leqslant 1$ and $0 \leqslant t \leqslant 1$. The mesh node positions $\{x_i^k\}$ are defined as

$$x_i^k = x(\xi_i, t^k) = x\left(\frac{i-1}{N}, \frac{k}{k_{\max}}\right), \quad k = 0, \ldots, k_{\max}, \;\; i = 1, \ldots, N+1,$$

where N is the number of spatial cells and $k_{\max}$ the number of pseudo-time steps. Example of such sequence of 1D meshes is shown in Fig. 6. Three different resolutions will be used, with both the number of cells and number of remaps increased by factor of two with respect to the previous resolution. In particular, we use $(N, k_{\max}) = (64, 320), (128, 640), (256, 1280)$.

5.2.1. Shock

The first test is a simple shock with the profile

$$\rho(x) = \begin{cases} 4 & \text{for } 0 \leqslant x \leqslant 0.5 \\ 1 & \text{for } 0.5 < x \leqslant 1 \end{cases}, \quad u(x) = \begin{cases} 1 & \text{for } 0 \leqslant x \leqslant 0.5 \\ 0 & \text{for } 0.5 < x \leqslant 1 \end{cases}$$

The result after 320 remaps on grid with 64 cells is shown in Fig. 7. We clearly see, that while the solution obtained with low-order fluxes is too diffusive, the high-order fluxes produce overshoots near the shock in density as well as in velocity. The SFCR solution follows the high-order profile on the shock, but correctly stays in bounds everywhere. At the final pseudo-time $t = 1$ the mesh is at its original position at $t = 0$, so the original old values are exact solutions to the cyclic remapping problem.

In Fig. 8 we provide for comparison the final results of SFCR and S&S methods on two resolutions, namely after 320 remaps on grid with 64 cells and after 1280 remaps on grid with 256 cells. Both SFCR and S&S results stay in bounds. The shock profiles by SFCR method are steeper than those obtained by S&S method and from the convergence rate estimates in Table 1 we can see that SFCR method is more accurate. The original S&S method [25] was designed for transport and did not require preservation of bounds in density. For fair comparison we have modified it by requiring that C^m on each face is bounded by the inequalities (3.16) and (3.17) derived in Section 3 on flux-corrected remap (FCR) for mass-density. Without the last requirement, the results by the original S&S method do not satisfy the density bounds, they satisfy only velocity bounds, however the final velocity profiles are almost the same as those presented in Fig. 8.

The relative L_1 errors and convergence rate estimates (ratio of error on current resolution to error on the previous one) are presented in Table 1. Notice that SFCR is more precise than both high-order and S&S methods.

CPU times for this convergence test are summarized in Table 2. SFCR method needs about five times more CPU time than the high-order method. S&S method consumes approximately three times more CPU time than the high-order method. Note however that these CPU times were obtained by our experimental non-optimized implementation of these methods. They are presented here only to demonstrate differences in complexity of the methods.

5.2.2. Exponential shock

The second test is the so-called exponential shock, with initial density and velocity profiles

$$\rho(x) = \begin{cases} 3\rho_0 \exp\left(\frac{x_F - x_0}{\delta}\right)(1+2\eta)^{-5/2} & \text{for } 0 \leqslant x \leqslant x_F \\ \rho_0 \exp\left(\frac{x - x_0}{\delta}\right) & \text{for } x_F < x \leqslant 15 \end{cases}, \tag{5.2a}$$

$$u(x) = \begin{cases} \frac{1-\eta}{t}\delta & \text{for } 0 \leqslant x \leqslant x_F \\ 0 & \text{for } x_F < x \leqslant 15 \end{cases} \tag{5.2b}$$

with

$$x_F = x_0 + \frac{3}{2}\delta \log\left(\frac{t}{t_0}\right), \quad \eta = \frac{x_F - x}{\delta}, \quad t_0 = 2, \quad x_0 = 6, \quad \rho_0 = 1, \quad \delta = 4, \quad t = 6.$$

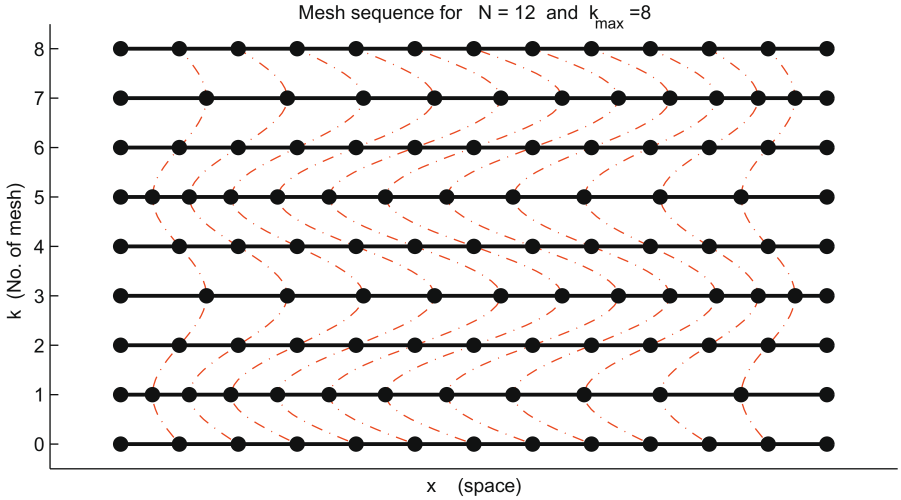

Fig. 6. Sequence of 1D meshes for cyclic remapping tests with $N = 12$ and $k_{\max} = 8$. Red dash-dot lines highlight how node positions change from mesh to mesh.

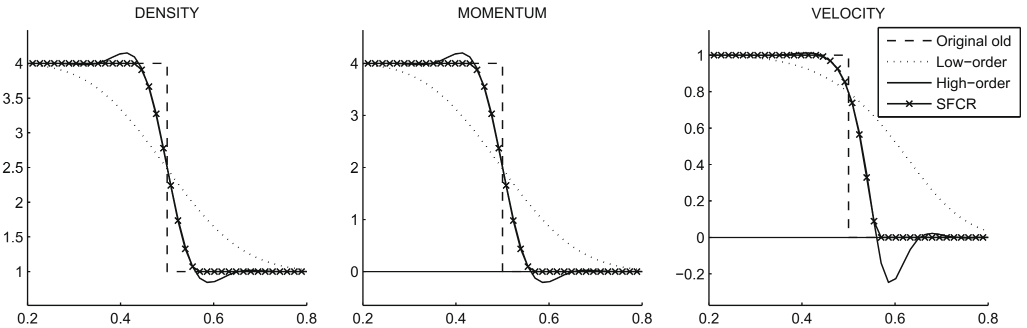

Fig. 7. The final results of cyclic remapping for shock problem after 320 remaps on mesh with 64 cells by low-order, high-order and SFCR approximations. The original old values are also the exact solution.

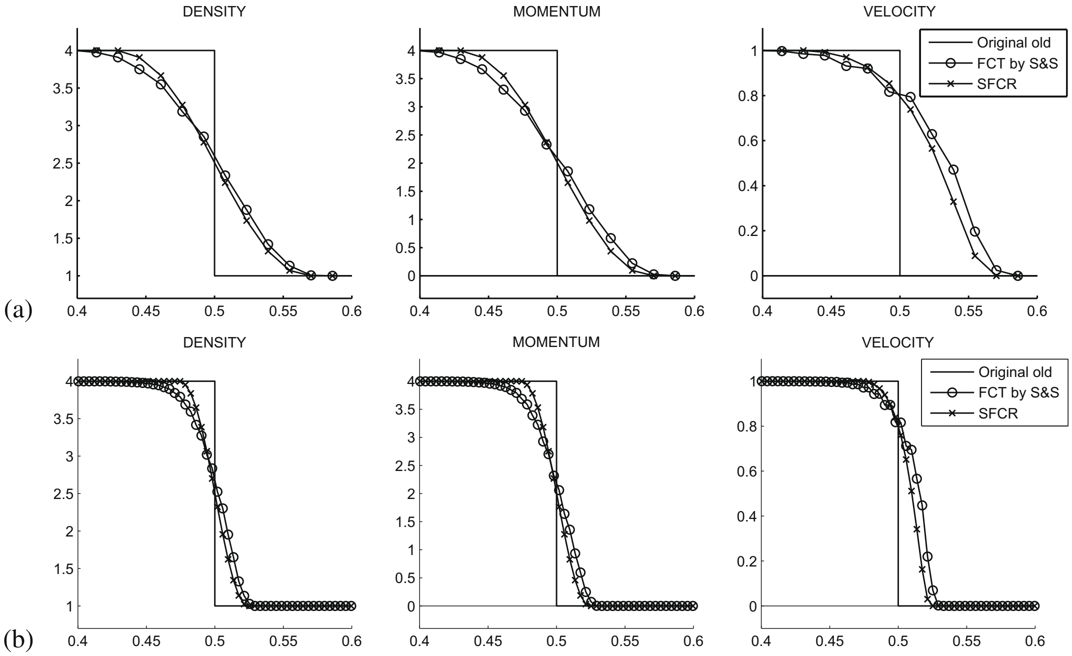

Fig. 8. The final results of cyclic remapping for shock problem by SFCR and S&S methods: (a) after 320 remaps on grid with 64 cells, (b) after 1280 remaps on grid with 256 cells. The original old values are also the exact solution.

Table 1
Convergence tables in density and velocity for cyclic remapping of shock by low-order, high-order, SFCR and S&S methods.

Method	Resolution		L_1 error		Err. prev./curr. resol.	
	N	$k_{\max}$	ρ	u	ρ	u
Low-order	64	320	0.121	0.262		
	128	640	0.086	0.186	1.4	1.4
	256	1280	0.061	0.131	1.4	1.4
High-order	64	320	0.038	0.093		
	128	640	0.022	0.055	1.7	1.7
	256	1280	0.013	0.033	1.7	1.7
SFCR	64	320	0.030	0.062		
	128	640	0.018	0.037	1.7	1.7
	256	1280	0.011	0.022	1.7	1.7
S&S	64	320	0.035	0.078		
	128	640	0.023	0.049	1.5	1.6
	256	1280	0.015	0.032	1.5	1.5

Table 2
CPU times in seconds (on a PC with AMD Opteron Processor 8384 running at 2.7 GHz) for cyclic remapping of shock by low-order, high-order, SFCR and S&S methods.

N	$k_{\max}$	Low-order	High-order	SFCR	S§
64	320	0.001	0.001	0.010	0.006
128	640	0.005	0.006	0.038	0.025
256	1280	0.020	0.030	0.151	0.100

As seen in Fig. 9, after 640 remaps on 128 cells grid, the low-order fluxes almost removed the peak in density, while the high-order fluxes produced a dip in velocity at right from the shock. Again, SFCR tracks the high-order solution and resolves the peaks reasonably, but avoids the undershoots.

In Fig. 10 we provide for comparison the final results of SFCR and S&S methods after 640 remaps on grid with 128 cells. Both SFCR and S&S results stay in bounds. The SFCR profiles are closer to the exact (old) solution than S&S profiles and from the convergence rate estimates in Table 3 we can see that also for this problem the SFCR method is more accurate.

The relative L_1 errors and convergence rate estimates are presented in Table 3.

6. FCR for mass-density and momentum-velocity in 2D

We are going to extend the SFCR method, developed in Section 4 for the scalar 1D case, to general multidimensional case. Most of the formulas from the scalar case remain the same, only we need to realize which quantities are vectors now. Clearly velocities $\mathbf{u}_i^{\max}$, $\mathbf{u}_i^{\min}$, momenta $\boldsymbol{\mu}_i$, $\tilde{\boldsymbol{\mu}}_i^L$ and momentum fluxes $\mathbf{F}_{i,k}^{\mu}$, $\mathbf{F}_{i,k}^{\mu,L}$, $\mathbf{F}_{i,k}^{\mu,H}$, $d\mathbf{F}_{i,k}^{\mu}$ are vectors, so that e.g. the velocity bounds (2.23), resp. (4.2) are written as

$$\mathbf{u}_i^{\min}\tilde{m}_i^L + \mathbf{u}_i^{\min}\sum_k C_{i,k}^m dF_{i,k}^m \leqslant \tilde{\boldsymbol{\mu}}_i^L + \sum_k C_{i,k}^{\mu} d\mathbf{F}_{i,k}^{\mu} \leqslant \mathbf{u}_i^{\max}\tilde{m}_i^L + \mathbf{u}_i^{\max}\sum_k C_{i,k}^m dF_{i,k}^m \tag{6.1}$$

and the upper bound, i.e. left inequality of (6.1) with C's moved to the left-hand side (multidimensional analogy of (4.4)) is

$$\sum_k C_{i,k}^{\mu} d\mathbf{F}_{i,k}^{\mu} - \mathbf{u}_i^{\max}\sum_k C_{i,k}^m dF_{i,k}^m \leqslant \mathbf{u}_i^{\max}\tilde{m}_i^L - \tilde{\boldsymbol{\mu}}_i^L. \tag{6.2}$$

Here we need to recall that all vector inequalities have to be understood in component by component sense as defined in (2.14).

6.1. Synchronized FCR (SFCR) in multidimensions

Extending the method from Section 4.2 to deal with vector velocity and momentum is quite straightforward. We apply the method for scalar velocity to each component of vector velocity and momentum, so that for each component we obtain two linear inequality constraints (4.40a) and (4.40b) on $C_{i,k}^m$ and $C_{i,k}^{\mu}$ and the admissible set is the intersection of all these constraints (half-planes) for all velocity components, together with $C_{i,k}^m \leqslant C_{i,k}^{m,\rho}$ which is coming from density-mass bounds (3.16) and (3.17). Thus again the admissible set is a convex polygon containing at least one point $(C_{i,k}^m, C_{i,k}^{\mu}) = (0,0)$.

Note that even when dealing with the vector velocity, the variable $C_{i,k}^{\mu}$ remains scalar, because during remapping we do not want to change the direction of velocity and momentum.

The derivation of SFCR for vector velocity can be written also in compact vector notation, basically almost exactly as in Section 4.2 with vectors $d\mathbf{F}_{i,k}^{\mu}$, $\mathbf{u}_i^{\max}$, $\mathbf{S}_i^{\mu,\max}$, $\Phi_{i,k}^{i,\mu,\max}$, $\Psi_{i,k}^{i,\mu,\max}$, $\mathbf{S}_i^{\mu,\max,+}$, $\mathbf{S}_i^{\mu,\max,-}$, $\mathbf{Q}_i^{\mu,\max}$, $\mathbf{P}_i^{\mu,\max,+}$, $\mathbf{D}_i^{\mu,\max,+}$ (and similarly for the

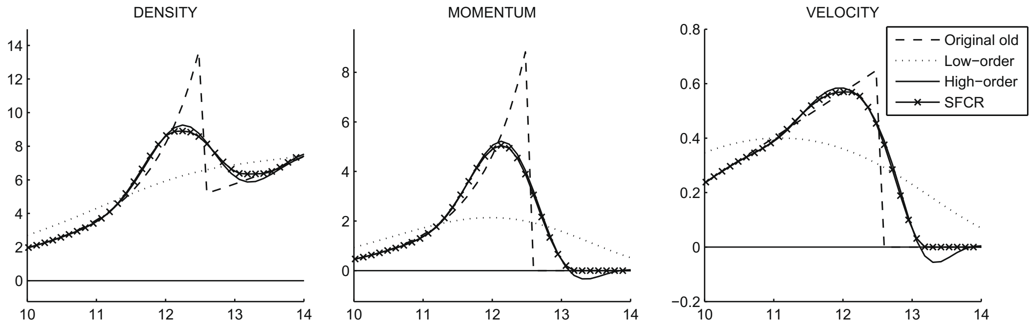

Fig. 9. The final results of cyclic remapping for exponential shock problem after 640 remaps on grid with 128 cells by low-order, high-order and SFCR approximations. The original old values are also the exact solution.

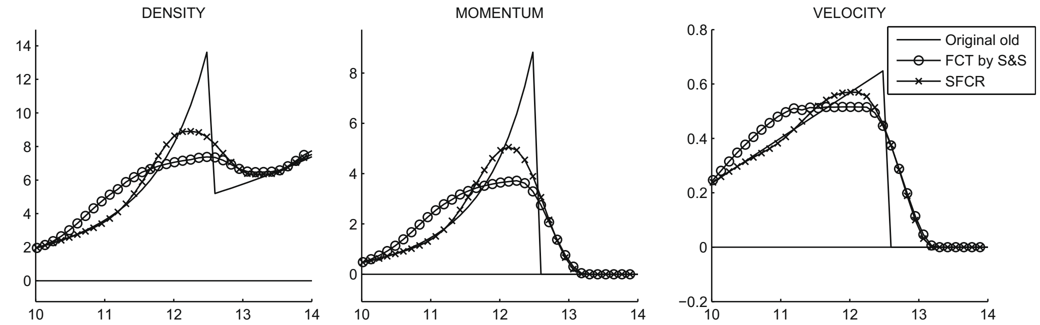

Fig. 10. The final results of cyclic remapping for exponential shock problem by SFCR and S&S methods: after 640 remaps on grid with 128 cells. The original old values are also the exact solution.

Table 3
Convergence tables in density and velocity for cyclic remapping of exponential shock by low-order, high-order, SFCR and S&S methods.

Method	Resolution		L_1 error		Err. prev./curr. resol.	
	N	$k_{\max}$	ρ	u	ρ	u
Low-order	64	320	0.257	0.341		
	128	640	0.190	0.205	1.4	1.7
	256	1280	0.163	0.128	1.2	1.6
High-order	64	320	0.122	0.056		
	128	640	0.068	0.030	1.8	1.9
	256	1280	0.046	0.018	1.5	1.7
SFCR	64	320	0.146	0.049		
	128	640	0.080	0.025	1.8	2.0
	256	1280	0.050	0.014	1.6	1.7
S&S	64	320	0.213	0.083		
	128	640	0.134	0.039	1.6	2.1
	256	1280	0.095	0.022	1.4	1.8

minimum bound). We are not going to repeat all this in the vector notation, we rather present only the final results, being the vector form of the constraints (4.40a) and (4.40b), in the compact vector notation:

$$C_{i,k}^{\mu} d\mathbf{F}_{i,k}^{\mu} - \mathbf{u}_i^{\max} C_{i,k}^{m} dF_{i,k}^{m} \leqslant \frac{\mathbf{Q}_i^{\mu,\max}}{\mathbf{P}_i^{\mu,\max,+}} \max\left(\mathbf{\Psi}_{i,k}^{i,\mu,\max}, \mathbf{0}\right), \tag{6.3a}$$

$$C_{i,k}^{\mu} d\mathbf{F}_{i,k}^{\mu} - \mathbf{u}_i^{\min} C_{i,k}^{m} dF_{i,k}^{m} \geqslant \frac{\mathbf{Q}_i^{\mu,\min}}{\mathbf{P}_i^{\mu,\min,-}} \min\left(\mathbf{\Psi}_{i,k}^{i,\mu,\min}, \mathbf{0}\right). \tag{6.3b}$$

where

$$\mathbf{Q}_i^{\mu,\max} = \mathbf{u}_i^{\max} \tilde{m}_i^L - \tilde{\boldsymbol{\mu}}_i^L \geqslant \mathbf{0},$$

$$\mathbf{Q}_i^{\mu,\min} = \mathbf{u}_i^{\min} \tilde{m}_i^L - \tilde{\boldsymbol{\mu}}_i^L \leqslant \mathbf{0},$$

$$\mathbf{\Psi}_{i,k}^{i,\mu,\max} = d\mathbf{F}_{i,k}^{\mu} - \mathbf{u}_i^{\max} dF_{i,k}^{m},$$

$$\mathbf{\Psi}_{i,k}^{i,\mu,\min} = d\mathbf{F}_{i,k}^{\mu} - \mathbf{u}_i^{\min} dF_{i,k}^{m},$$

$$\mathbf{P}_i^{\mu,\max,+} = \sum_{\mathbf{\Psi}_{i,k}^{i,\mu,\max} \ogreaterthan 0} \mathbf{\Psi}_{i,k}^{i,\mu,\max} \geq \mathbf{0},$$

$$\mathbf{P}_i^{\mu,\min,-} = \sum_{\mathbf{\Psi}_{i,k}^{i,\mu,\min} \olessthan 0} \mathbf{\Psi}_{i,k}^{i,\mu,\min} \leq \mathbf{0},$$

and where the relation operators $\circledgt$, $\circledlt$, in the k ranges of sums defining $\mathbf{P}'s$ mean, that the ζ component of the sum (i.e. $\mathbf{P}_\zeta$) includes only k for which corresponding component $(\mathbf{\Psi}_{i,k})_\zeta$ of given $\mathbf{\Psi}$ has given sign, so e.g. the ζ-component for $\mathbf{P}_i^{\mu,\max,+}$ is

$$\left(\mathbf{P}_i^{\mu,\max,+}\right)_\zeta = \sum_{\left(\mathbf{\Psi}_{i,k}^{i,\mu,\max}\right)_\zeta > 0} \left(\mathbf{\Psi}_{i,k}^{i,\mu,\max}\right)_\zeta \geqslant 0.$$

All the operators (i.e. division, multiplication, maximum and minimum) on the right-hand side of inequalities (6.3a) and (6.3b) are to be understood in a component-wise sense, so that e.g.

$$\left(\frac{\mathbf{Q}_i^{\mu,\max}}{\mathbf{P}_i^{\mu,\max,+}} \max\left(\mathbf{\Psi}_{i,k}^{i,\mu,\max}, \mathbf{0}\right)\right)_\zeta = \frac{\left(\mathbf{Q}_i^{\mu,\max}\right)_\zeta}{\left(\mathbf{P}_i^{\mu,\max,+}\right)_\zeta} \max\left(\left(\mathbf{\Psi}_{i,k}^{i,\mu,\max}\right)_\zeta, 0\right).$$

6.2. Optimal choice of C's in multidimensions

The optimal choice of C's in multidimensions proceeds in the same way as in 1D in Section 4.3. The admissible set is again a convex polygon in the $(C_{i,k}^m, C_{i,k}^\mu)$ plane containing the origin $(0,0)$. The derivation would proceed in the same way as in 1D, only the momentum density $\mathbf{n} = \rho\mathbf{u}$ and momentum fluxes are vectors and their absolute values have to be replaced by norms of vectors, so that the final minimized local function (4.44) in multidimensions is written as

$$E_{i,k}^{L_1} = \left(1 - C_{i,k}^m\right)\left|dF_{i,k}^m\right|\left(\frac{1}{\rho_i} + \frac{1}{\rho_k}\right) + \left(1 - C_{i,k}^\mu\right)\left\|d\mathbf{F}_{i,k}^\mu\right\|\left(\frac{1}{\|\mathbf{n}_i\| + \epsilon} + \frac{1}{\|\mathbf{n}_k\| + \epsilon}\right) \tag{6.4}$$

and we minimize this function on face (i,k) with $(C_{i,k}^m, C_{i,k}^\mu)$ from the admissible set. The objective function $E_{i,k}^{L_1}$ is linear in $C_{i,k}^m$ and $C_{i,k}^\mu$, the admissible set is convex and we take the minimum in one vertex of the admissible set as in 1D.

6.3. Numerical results in 2D

In this section we present several numerical tests of our SFCR method in 2D. Our current implementation works with quadrilateral, logically rectangular meshes in 2D, however the method presented above can be directly applied also to unstructured meshes and to 3D. For low-order approximation we use donor fluxes based on piecewise constant reconstruction on the old grid. For high-order approximation we use remapping fluxes based on piecewise linear reconstruction of the conserved quantities ρ and $\mathbf{n} = \rho\mathbf{u}$ on the old grid, where the slopes of linear reconstruction inside one cell are obtained by least square minimization of deviations of given linear reconstruction from mean values in the neighboring cells.

6.3.1. Demonstrative simple example in 2D

Let us take a particular example, which is a generalization of the 1D demonstrative example from Section 5.1. In the two-dimensional domain $[0,8] \times [0,8]$, we have a structured quadrilateral mesh of 9×9 equidistant nodes with cell-centered values of density and velocity given by the discretization of functions

$$\rho(x,y) = x + 10 \quad \text{for } 0 \leqslant x \leqslant 8, \quad \mathbf{u}(x,y) = \begin{cases} (x,y) & \text{for } 0 \leqslant x \leqslant 4, \\ (-x,y) & \text{for } 4 < x \leqslant 8. \end{cases} \tag{6.5}$$

We see that density and the y-component of velocity are given by a smooth linear function, while in the linear function defining the x-component of velocity there is a jump across the line $x = 4$. Then we move the mesh and remap the values. To make the example simple, we only move the nodes in x-direction. In particular, we shift all mesh nodes except those at the left and right boundaries by $\delta x = -0.2$, i.e. to the left as shown in Fig. 11.

We focus here only on one row of cells with centers at $y = 3.5$, five of which are denoted by $i,\ j,\ k,\ l,\ m$ in Fig. 11. The detailed derivation of all constraints on C's (defining the admissible sets) is presented in Appendix A, here we present only shortly the final results. First as the density is linear there are no constraints on C^m coming from density bounds. As in the case of 1D demonstrative example, which is very close to this 2D example, only on two interfaces, namely (j,k) and (k,l) around the velocity discontinuity, the velocity bounds impose constraints on C's. On the other interfaces we use $C^m = C^\mu = 1$, i.e. high-order method, which does not violate any bounds. The constraints affecting the admissible sets together with the admissible sets and final solution for both (j,k) and (k,l) interfaces are plotted in Fig. 12. Some constraints are satisfied but they are too far to be shown in Fig. 12.

Note, that if we would change the initial density and velocity to

$$\rho(x,y) = x + 10 \quad \text{for } 0 \leqslant x \leqslant 8, \quad \mathbf{u}(x,y) = \begin{cases} (x,0) & \text{for } 0 \leqslant x \leqslant 4 \\ (-x,0) & \text{for } 4 < x \leqslant 8 \end{cases} \tag{6.6}$$

instead of (6.5), we would get exactly the 1D demonstrative example from Section 5.1, only generalized to the 2D mesh.

6.3.2. Cyclic remapping in 2D

To present results of our SFCR remapping in 2D we use the two different cyclic sequences of meshes from [22] on the unit square domain $[0,1] \times [0,1]$.

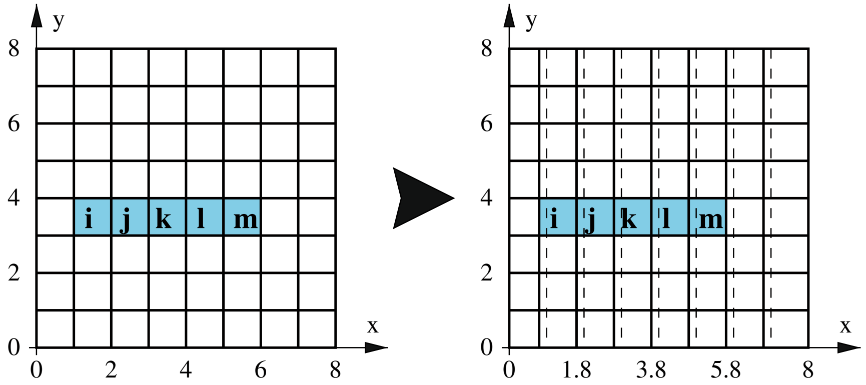

Fig. 11. Demo example. Left: old mesh with highlighted cells of interest. Right: New mesh by solid line, old mesh by dashed line.

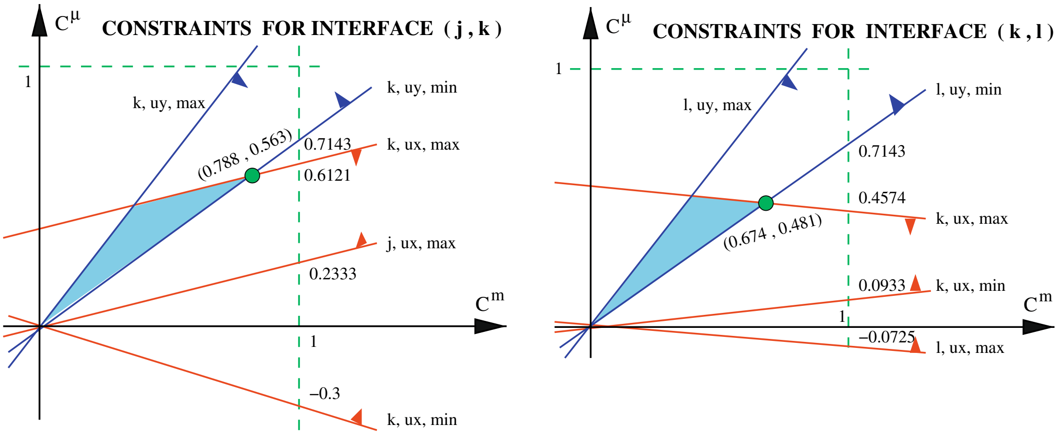

Fig. 12. Demo example in 2D (6.5). Constraints, admissible sets and final solutions for interfaces (j,k) (left panel) and (k,l) (right panel).

The sequence of tensor product meshes is defined by

$$\begin{aligned} x(\xi,t) &= [1-\alpha(t)]\xi + \alpha(t)\xi^3,\\ y(\eta,t) &= [1-\alpha(t)]\eta + \alpha(t)\eta^2,\\ \alpha(t) &= \frac{\sin(4\pi t)}{2} \end{aligned} \tag{6.7}$$

for $0 \leqslant \xi \leqslant 1, 0 \leqslant \eta \leqslant 1$ and $0 \leqslant t \leqslant 1$. The mesh node positions $\{\mathbf{x}^k_{i,j}\}$ are defined as

$$\mathbf{x}^k_{i,j} = \left(x\left(\xi_i,t^k\right), y\left(\eta_j,t^k\right)\right) = \left(x\left(\frac{i-1}{N},\frac{k}{k_{\max}}\right), y\left(\frac{j-1}{N},\frac{k}{k_{\max}}\right)\right), \quad k=0,\ldots,k_{\max},\; i=1,\ldots,N+1,\; j=1,\ldots,N+1,$$

where N is the number of spatial cells in both directions and $k_{\max}$ the number of pseudo-time steps.

The sequence of smooth non-orthogonal meshes is given by

$$\begin{aligned} x(\xi,\eta,t) &= \xi + \alpha(t)\sin(2\pi\xi)\sin(2\pi\eta),\\ y(\xi,\eta,t) &= \eta + \alpha(t)\sin(2\pi\xi)\sin(2\pi\eta),\\ \alpha(t) &= \begin{cases} t/5 & \text{for } 0 \leqslant t \leqslant 0.5,\\ (1-t)/5 & \text{for } 0.5 \leqslant t \leqslant 1, \end{cases} \end{aligned} \tag{6.8}$$

for $0 \leqslant \xi \leqslant 1,\; 0 \leqslant \eta \leqslant 1$ and $0 \leqslant t \leqslant 1$. The mesh node positions $\{\mathbf{x}^k_{i,j}\}$ are defined as

$$\mathbf{x}^k_{i,j} = \left(x\left(\xi_i,\eta_j,t^k\right), y\left(\xi_i,\eta_j,t^k\right)\right) = \left(x\left(\frac{i-1}{N},\frac{j-1}{N},\frac{k}{k_{\max}}\right),\quad y\left(\frac{j-1}{N},\frac{i-1}{N},\frac{k}{k_{\max}}\right)\right), \quad k=0,\ldots,k_{\max},$$
$$i=1,\ldots,N+1,\; j=1,\ldots,N+1.$$

Table 4
Cyclic remapping of 2D shock on the sequence of tensor product meshes: convergence tables in density and velocity components for cyclic remapping by low-order, high-order and SFCR methods.

Method	Resolution		L_1 error			Err. prev./curr. resol.		
	N	$k_{\max}$	ρ	u_x	u_y	ρ	u_x	u_y
Low-order	64	320	0.0708	0.0762	0.0762			
	128	640	0.0503	0.0541	0.0541	1.41	1.41	1.41
	256	1280	0.0356	0.0384	0.0384	1.41	1.41	1.41
High-order	64	320	0.0222	0.0247	0.0247			
	128	640	0.0133	0.0148	0.0148	1.67	1.67	1.67
	256	1280	0.0080	0.0088	0.0088	1.67	1.67	1.67
SFCR	64	320	0.0177	0.0189	0.0189			
	128	640	0.0106	0.0113	0.0113	1.68	1.68	1.68
	256	1280	0.0063	0.0067	0.0067	1.68	1.68	1.68

6.3.2.1. 2D shock. The first test is a simple shock on domain $[0,1]\times[0,1]$ with the profile

$$\rho(x,y)=\begin{cases}2 & \text{for } y>(x-0.4)/0.3,\\ 1 & \text{for } y\leqslant(x-0.4)/0.3,\end{cases}$$
$$\mathbf{u}(x,y)=\begin{cases}(2,-0.6) & \text{for } y>(x-0.4)/0.3,\\ (1,-0.3) & \text{for } y\leqslant(x-0.4)/0.3.\end{cases}$$

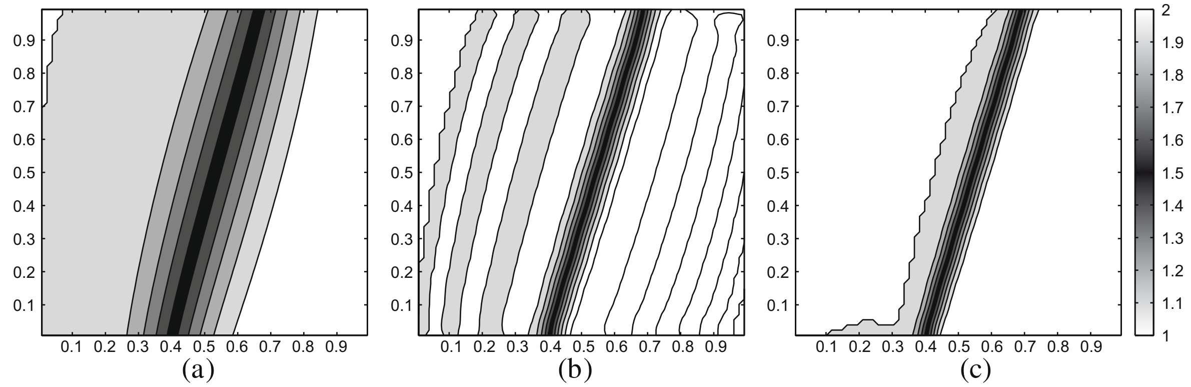

Fig. 13. Cyclic remapping of 2D shock on the sequence of tensor product meshes with $N=64,\ k_{\max}=320$: density by (a) low-order, (b) high-order and (c) SFCR method; 11 contours 1.0, 1.1, …, 1.9, 2.0.

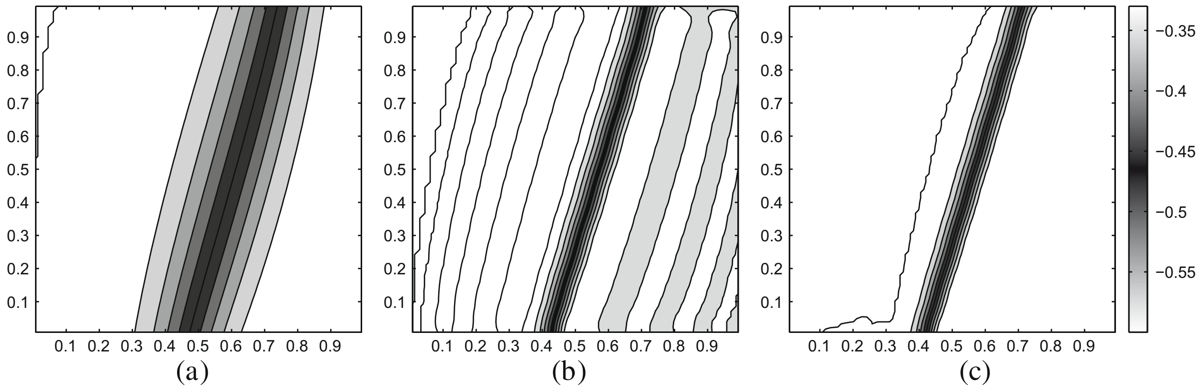

Fig. 14. Cyclic remapping of 2D shock on the sequence of tensor product meshes with $N=64,\ k_{\max}=320$: y-component of velocity by (a) low-order, (b) high-order and (c) SFCR method; 11 contours −0.6, −0.57, …, −0.33, −0.3.

Table 5
CPU times in seconds (on a PC with AMD Opteron Processor 8384 running at 2.7 GHz) for cyclic remapping of 2D shock on the sequence of tensor product meshes by low-order, high-order and SFCR methods.

N	$k_{\max}$	Low-order	High-order	SFCR
64	320	1.1	1.7	4.9
128	640	11	16	50
256	1280	117	170	498

Table 6
Cyclic remapping of 2D shock on the sequence of smooth non-orthogonal meshes: convergence tables in density and velocity components for cyclic remapping by low-order, high-order and SFCR methods.

Method	Resolution		L_1 error			Err. prev./curr. resol.		
	N	$k_{\max}$	ρ	u_x	u_y	ρ	u_x	u_y
Low-order	64	320	0.0141	0.0152	0.0152			
	128	640	0.0101	0.0109	0.0109	1.39	1.39	1.39
	256	1280	0.0072	0.0078	0.0078	1.4	1.40	1.40
High-order	64	320	0.0086	0.0097	0.0097			
	128	640	0.0053	0.0059	0.0059	1.62	1.63	1.63
	256	1280	0.0033	0.0036	0.0036	1.63	1.64	1.64
SFCR	64	320	0.0068	0.0073	0.0073			
	128	640	0.0042	0.0045	0.0045	1.61	1.62	1.62
	256	1280	0.0026	0.0027	0.0027	1.63	1.64	1.64

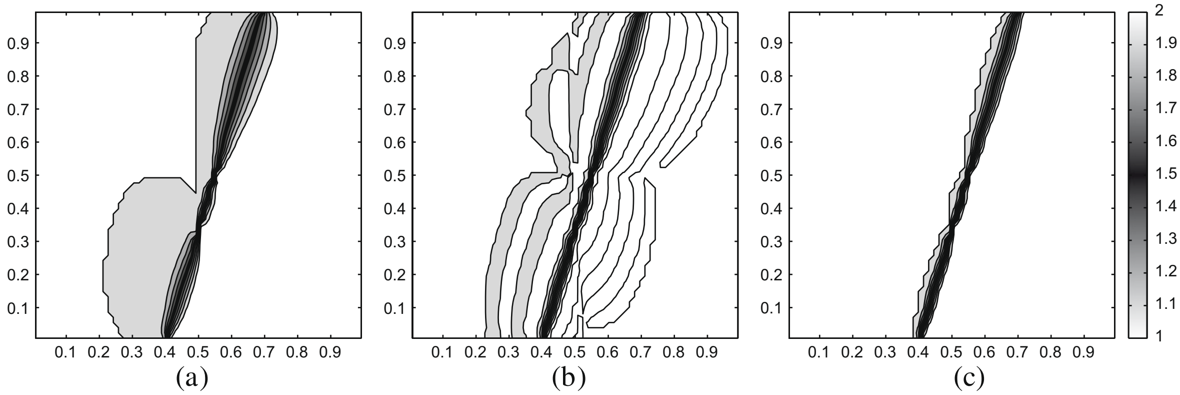

Fig. 15. Cyclic remapping of 2D shock on the sequence of smooth non-orthogonal meshes with $N = 64,\ k_{\max} = 320$: density by (a) low-order, (b) high-order and (c) SFCR method; 11 contours $1.0, 1.1, \ldots, 1.9, 2.0$.

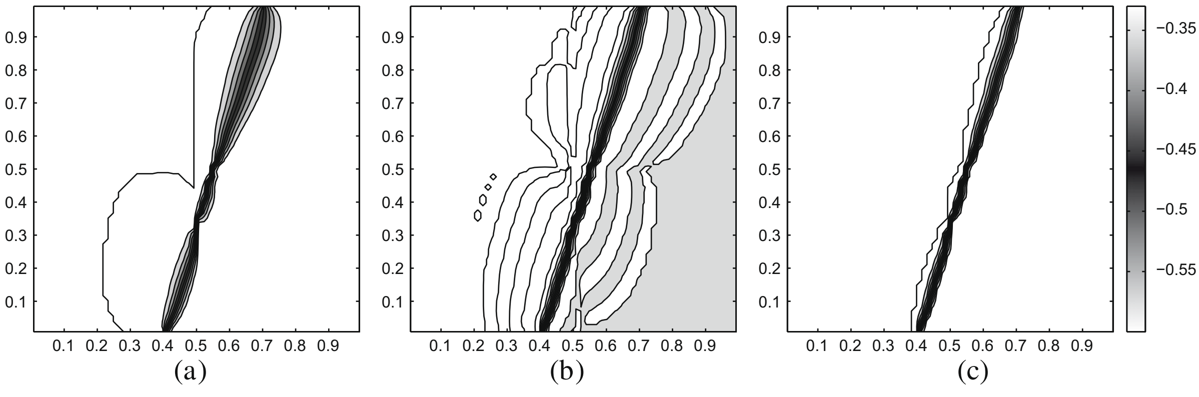

Fig. 16. Cyclic remapping of 2D shock on the sequence of smooth non-orthogonal meshes with $N = 64,\ k_{\max} = 320$: y-component of velocity by (a) low-order, (b) high-order and (c) SFCR method; 11 contours $-0.6, -0.57, \ldots, -0.33, -0.3$.

Table 7
Cyclic remapping of 2D Sedov analog on the sequence of smooth non-orthogonal meshes: convergence tables in density and velocity components for cyclic remapping by low-order, high-order and SFCR methods.

Method	Resolution		L_1 error			Err. prev./curr. resol.		
	N	$k_{\max}$	ρ	u_x	u_y	ρ	u_x	u_y
Low-order	64	320	0.2870	0.4604	0.4604			
	128	640	0.2300	0.3247	0.3247	1.25	1.42	1.42
	256	1280	0.1761	0.2257	0.2257	1.31	1.44	1.44
High-order	64	320	0.1800	0.3899	0.3899			
	128	640	0.1143	0.1592	0.1592	1.57	2.45	2.45
	256	1280	0.0705	0.0947	0.0947	1.62	1.68	1.68
SFCR	64	320	0.1684	0.2387	0.2386			
	128	640	0.1054	0.1114	0.1114	1.6	2.14	2.14
	256	1280	0.0642	0.0623	0.0623	1.64	1.79	1.79

The results for cyclic remapping of 2D shock on the sequence of tensor product meshes (6.7) are presented in Table 4 showing the relative L_1 errors of the final solution by low-order, high-order and SFCR methods at three mesh-pseudo-time resolutions $(N, k_{\max}) = (64, 320), (128, 640), (256, 1280)$. The SFCR method is producing the smallest error while staying in bounds which is documented in Fig. 13 resp. Fig. 14 showing density resp. y-component of the velocity at the lowest resolution. The high-order method produces oscillations in both density and velocity on both sides of the shock and these oscillations violate the local bounds (monotonicity) of the remapped function.

CPU times for cyclic remapping of 2D shock on the sequence of tensor product meshes are summarized in Table 5. SFCR method needs about three times more CPU time than the high-order method. Note however again that these CPU times were obtained by our experimental non-optimized implementation of these methods. They are presented here only to demonstrate differences in complexity of the methods.

The results for cyclic remapping of 2D shock on the sequence of smooth non-orthogonal meshes (6.8) are presented in Table 6 showing the relative L_1 errors of the final solution by low-order, high-order and SFCR methods at three mesh-pseudo-time resolutions. SFCR method is producing the smallest error while staying in bounds which is documented in Fig. 15 resp. Fig. 16 showing density resp. y-component of the velocity. The high-order method produces oscillations in both density

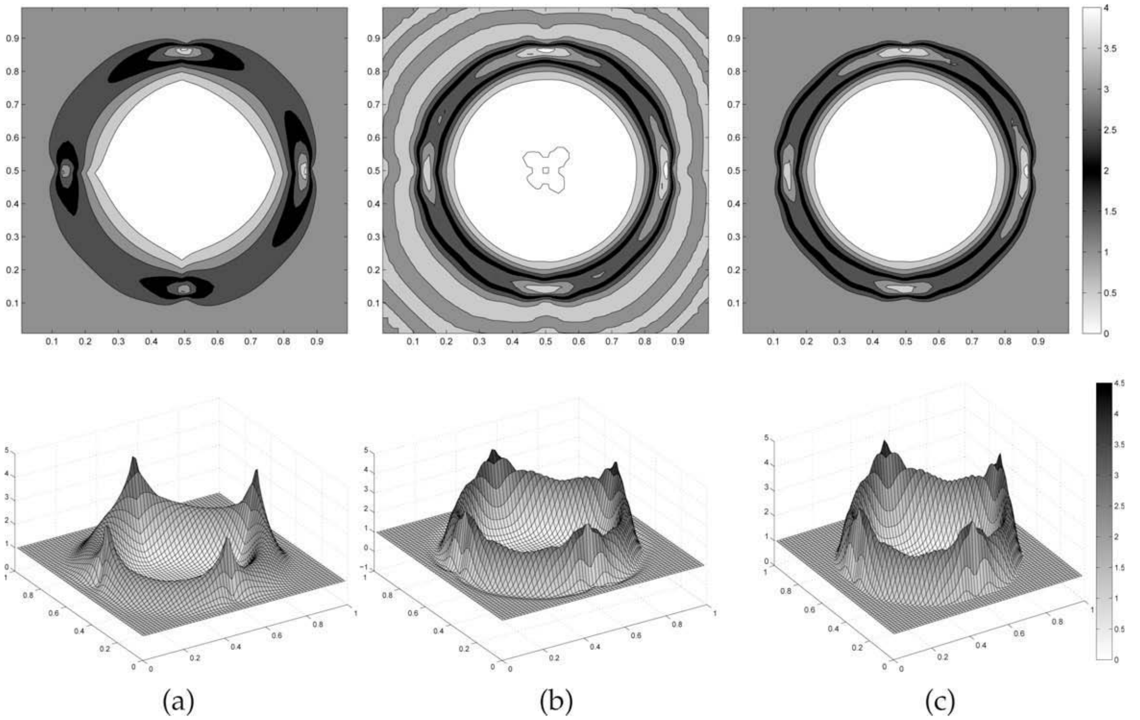

Fig. 17. Cyclic remapping of 2D Sedov analog on the sequence of smooth non-orthogonal meshes with $N = 64,\ k_{\max} = 320$: density by (a) low-order, (b) high-order and (c) SFCR method; 13 contours $0, 0.5, \ldots, 5.5, 6$.

and velocity on both sides of the shock and these oscillations violate the local bounds (monotonicity) of the remapped function. Curved shape of isolines showing the oscillations for the high-order method in Figs. 15(b) and 16(b) is caused by the fact that the central point of the meshes $(x, y) = (0.5, 0.5)$ is not moving during the mesh movement, see (6.8) with $\xi = \eta = 0.5$.

6.3.2.2. Analog of 2D Sedov. The second test is defined by the functions centered at $(x_0, y_0) = (0.5, 0.5)$ on the domain $[0, 1] \times [0, 1]$

$$\rho(x,y) = \begin{cases} 6\left(\frac{r}{r_0}\right)^8 & \text{for } r < r_0, \\ 1 & \text{for } r \geqslant r_0, \end{cases}$$
$$\mathbf{u}(x,y) = \begin{cases} 0.83(x - x_0, y - y_0) & \text{for } r < r_0, \\ (0,0) & \text{for } r \geqslant r_0, \end{cases}$$

where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$, $r_0 = 0.375$, which is quite close to the scaled solution of cylindrical 2D Sedov problem [9] (the case of cylindrical Sedov problem with initial conditions: constant density $\rho_0 = 1$, initial energy $E_0 = 0.311357$, $\gamma = 1.4$ at time $t = 1$).

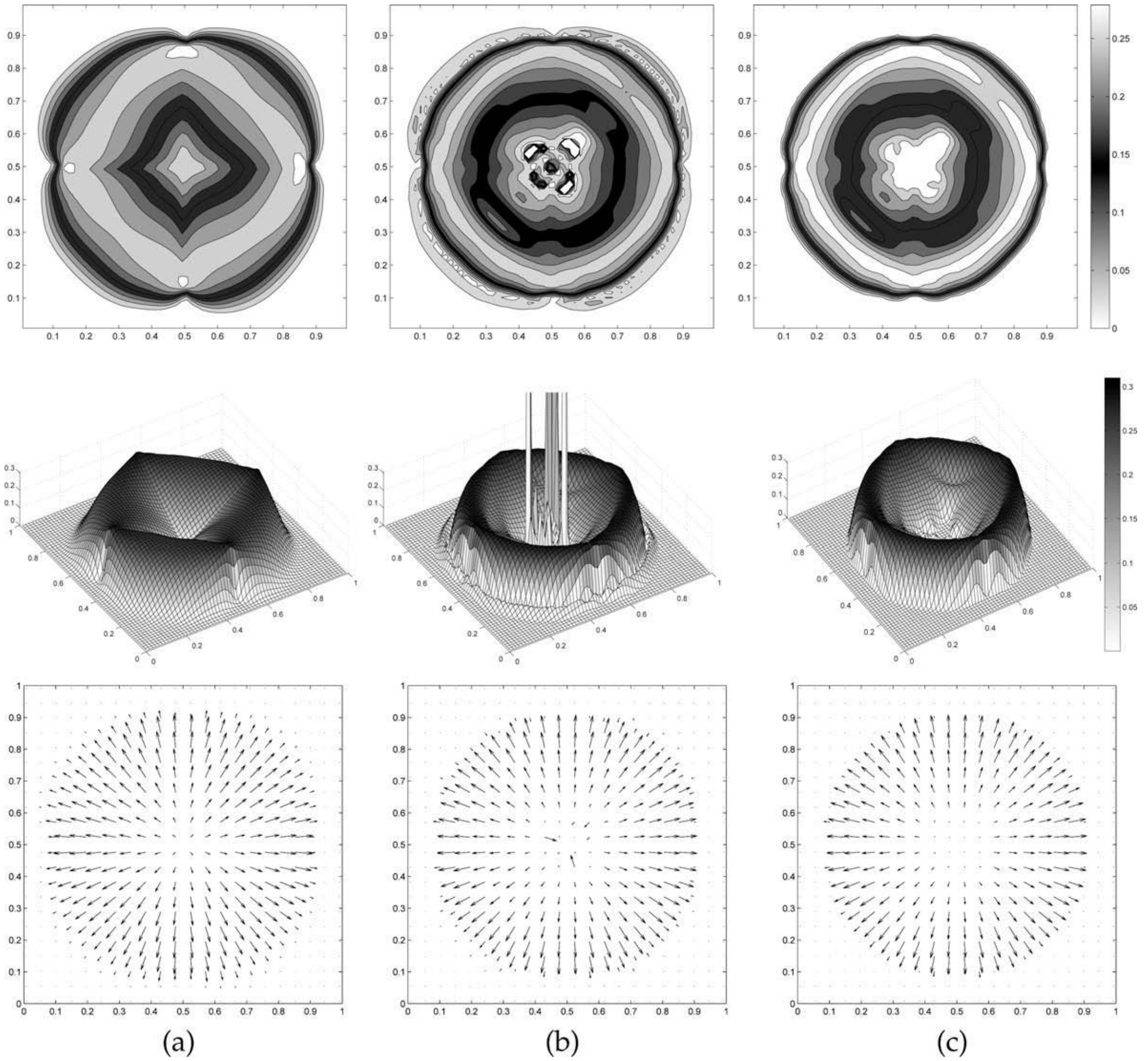

Fig. 18. Cyclic remapping of 2D Sedov analog on the sequence of smooth non-orthogonal meshes with $N = 64$, $k_{\max} = 320$: velocity by (a) low-order, (b) high-order and (c) SFCR method; 11 contours 0,0.031,...,0.31. The upper row is showing the contour plots of the norm of velocity, the middle row the surface plot of the norm of velocity and the lower row the directional arrows of the velocity.

The results for cyclic remapping of this 2D Sedov analog on the sequence of smooth non-orthogonal meshes (6.8) are presented in Table 7 showing the relative L_1 errors of the final solution by low-order, high-order and SFCR methods at three mesh-pseudo-time resolutions. SFCR method is again producing the smallest error while staying in bounds which is documented in Fig. 17 resp. Fig. 18 showing density resp. velocity at the lowest resolution. In the density contour plot of high-order method in Fig. 17(b) one clearly sees the density oscillations outside the circular shock. However note also the zero density contour in the center of the contour plot which demonstrates that the density is getting negative there for the high-order method while it should be very small, but still positive as it is for the low-order and SFCR methods. The situation is getting even worse for velocity. As can be seen in Fig. 18(b), velocity from the high-order method around the center point $(x_0, y_0) = (0.5, 0.5)$ has huge oscillations. The reason of these oscillations is negative density in this region. When density is negative, then also masses of cells in this region become negative and the negative masses are changing the direction of the velocity vector (see the arrows in the lower plot in Fig. 18(b)) computed from remapped momentum and mass (2.12). The changed direction of the velocity introduces oscillations which are further amplified by the high-order method during following remaps. Note however that this is a difficult problem as the density in the four central cells at the lowest resolution is of the order 10^{-12}, the cell masses are even smaller and division by a small number can cause big numerical errors.

7. Conclusions and future work

In this paper we have developed a new optimization-based synchronized flux-corrected remapping of mass/density and momentum/velocity. This new method is intended to be used in arbitrary Lagrangian–Eulerian methods. The main new results are deriving local linear objective function and new sufficient conditions for bound preservation. We have shown that in 1D our new method is superior in comparison with the method from [25]. On 1D and 2D examples we have demonstrated the performance of a new method.

There are several interesting problems which we are planning to investigate in future paper(s). First of all, the results may depend on the choice of low-order and high-order fluxes because it will affect the objective function. As it was mentioned in [14, p. 224], one can consider limiting different components of the velocity, that is components related to main direction of the flow. Also, as it is clear from results presented in [26], accuracy of advection can strongly depend on the choice of bounds. In context of remapping this means that we need to take into account which old cells the new cell intersect. We also plan to use iterative solution strategies [25,14], where compatible fluxes obtained in this paper will be used as low-order fluxes. Next, in ALE methods we also need to remap internal energy. Internal energy is usually obtained as the difference of total and kinetic energy (which is a quadratic function of the velocity). Bound preservation requirements for internal energy will lead to quadratic constraints.

Acknowledgments

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract DE-AC52-06NA25396. The authors acknowledge the partial support of the DOE Advance Simulation and Computing (ASC) Program and the DOE Office of Science ASCR Program, and the Laboratory Directed Research and Development program (LDRD) at the Los Alamos National Laboratory. The first and the third author have been supported in part by the Czech Ministry of Education grants MSM 6840770022, MSM 6840770010 and LC528.

The authors thank P. Smolarkiewicz, D. Kuzmin, C. Schär, B. Kashiwa, T. Ringler, J. Shadid, L. Margolin and B. Rider for fruitful discussions and constructive comments.

Appendix A. Detailed description of demonstrative 2D example

A.1. Constraints from density bounds

Let us focus on five cells denoted in Fig. 11 by indices $i,\ j,\ k,\ l$ and m. The bounds for density and mass are

Cell	i	j	k	l	m
$\rho^{\min}$	10.5	11.5	12.5	13.5	14.5
$\rho^{\max}$	12.5	13.5	14.5	15.5	16.5
$m^{\min}$	10.5	11.5	12.5	13.5	14.5
$m^{\max}$	12.5	13.5	14.5	15.5	16.5

For simplicity, we moved the mesh nodes only in x-direction. Therefore there is no mass or momentum flux across the logically horizontal faces and thus e.g. cell k only exchanges mass and momentum with cells j and l, etc. For the interfaces between our cells of interest we have the following low-order, high-order and antidiffusive fluxes:

Interfaces $a,\ b$	i,j	j,k	k,l	l,m
$F^{m,L}_{a,b} = -F^{m,L}_{b,a}$	−2.3	−2.5	−2.7	−2.9
$F^{m,H}_{a,b} = -F^{m,H}_{b,a}$	−2.38	−2.58	−2.78	−2.98
$dF^{m}_{a,b} = -dF^{m}_{b,a}$	−0.08	−0.08	−0.08	−0.08

Because the values of $m^L,\ Q^m,\ P^m$ etc. for cells i and m depend also on fluxes from cells left of i resp. right of m, we further show only values for cells j, k and l, which are fully given by the fluxes shown in the table above. The formulas yield

Cell	j	k	l
m^{old}	12.5	13.5	14.5
m^L	12.3	13.3	14.3
$Q^{m,\min}$	0.8	0.8	0.8
$Q^{m,\max}$	1.2	1.2	1.2
$P^{m,-}$	0.08	0.08	0.08
$P^{m,+}$	0.08	0.08	0.08
$Q^{m,\min}/R^{m,-}$	10	10	10
$Q^{m,\max}/R^{m,+}$	15	15	15
$C^{m,-}$	1	1	1
$C^{m,+}$	1	1	1

and therefore at interface (j,k) as well as (k,l) the high-order mass flux does not need to be corrected due to velocity constraints:

$$C^{m,\rho}_{j,k} = C^{m,\rho}_{k,j} = 1, \quad C^{m,\rho}_{k,l} = C^{m,\rho}_{l,k} = 1.$$

Note that the values $C^{m,\rho}$ defined in (3.16) and (3.17) are only density-imposed upper bounds for final C^m's, which can be further decreased by the velocity constraints.

A.2. Constraints from velocity bounds

Now let us consider the lower and upper bounds for velocity:

Cell	i	j	k	l	m
$\mathbf{u}^{\min}$	(0.5,2.5)	(1.5,2.5)	(−4.5,2.5)	(−5.5,2.5)	(−6.5,2.5)
$\mathbf{u}^{\max}$	(2.5, 4.5)	(3.5,4.5)	(3.5,4.5)	(3.5,4.5)	(−4.5,4.5)

The nonzero fluxes are

Interfaces a,b	$i,\ j$	$j,\ k$	$k,\ l$	$l,\ m$
$\mathbf{F}^{\mu,L}_{a,b} = -\mathbf{F}^{\mu,L}_{b,a}$	(−3.45, −8.05)	(−6.25, −8.75)	(−9.45, −9.45)	(13.05, −10.15)
$\mathbf{F}^{\mu,H}_{a,b} = -\mathbf{F}^{\mu,H}_{b,a}$	(−4.49, −8.33)	(−7.45, −9.03)	(−5.59, −9.73)	(18.35, −10.43)
$d\mathbf{F}^{\mu}_{a,b} = -d\mathbf{F}^{\mu}_{b,a}$	(−1.04, −0.28)	(−1.20, −0.28)	(3.86, −0.28)	(5.30, −0.28)

where the high-order fluxes were computed by integration of piecewise linear approximation of momentum. (Because of the simple setup and initial condition we have, the low-order momentum fluxes and all mass fluxes are exact so far.)

The "unlimited" terms in cells with respect to interfaces with their neighbors are

Cell a	i	j		k	
Neighb. b	j	i	k	j	l
$\mathbf{\Psi}^{\mu,\min}_{a,b}$	(−1.00, −0.08)	(0.92, 0.08)	(−1.08, −0.08)	(1.56, 0.08)	(3.50, −0.08)
$\mathbf{\Psi}^{\mu,\max}_{a,b}$	(−0.84, 0.08)	(0.76, −0.08)	(−0.92, 0.08)	(0.92, −0.08)	(4.14, 0.08)

Cell a	l		m
Neighb. b	k	m	l
$\mathbf{\Psi}^{\mu,\min}_{a,b}$	$(-3.42, 0.08)$	$(\ 4.86, -0.08)$	$(-4.78, 0.08)$
$\mathbf{\Psi}^{\mu,\max}_{a,b}$	$(-4.14, -0.08)$	$(\ 5.58, 0.08)$	$(-4.94, -0.08)$

which gives for cells $j,\ k$ and l:

Cell	j	k	l
$\boldsymbol{\mu}^{\text{old}}$	$(31.25, 43.75)$	$(47.25, 47.25)$	$(-65.25, 50.75)$
$\boldsymbol{\mu}^{L}$	$(28.45, 43.05)$	$(44.05, 46.55)$	$(-42.75, 50.05)$
$\mathbf{Q}^{\mu,\min}$	$(10.0, 12.3)$	$(103.9, 13.3)$	$(35.9, 14.3)$
$\mathbf{Q}^{\mu,\max}$	$(14.6, 12.3)$	$(2.5, 13.3)$	$(92.8, 14.3)$
$\mathbf{P}^{\mu,\min,-}$	$(1.08, 0.08)$	$(0, 0.08)$	$(3.42, 0.08)$
$\mathbf{P}^{\mu,\max,+}$	$(0.76, 0.08)$	$(5.06, 0.08)$	$(5.58, 0.08)$
$\mathbf{D}^{\mu,\min,-}$	$(9.259, 153.75)$	$(\text{“}\infty\text{”}, 166.25)$	$(10.497, 178.75)$
$\mathbf{D}^{\mu,\max,+}$	$(19.211, 153.75)$	$(0.49407, 166.25)$	$(16.631, 178.75)$

Now we must construct the constraints according to formulas (6.3a) and (6.3b). For interface (j,k), after insertion in the formulas

$$\left(C^{\mu}_{j,k} d\mathbf{F}^{\mu}_{j,k} - \mathbf{u}^{\min}_{j} C^{m}_{j,k} dF^{m}_{j,k}\right)_{\zeta} \geqslant (\mathbf{D}^{\mu,\min,-}_{j})_{\zeta} \min\left(\left(\mathbf{\Psi}^{\mu,\min}_{j,k}\right)_{\zeta}, 0\right),$$

$$\left(C^{\mu}_{j,k} d\mathbf{F}^{\mu}_{j,k} - \mathbf{u}^{\max}_{j} C^{m}_{j,k} dF^{m}_{j,k}\right)_{\zeta} \leqslant (\mathbf{D}^{\mu,\max,+}_{j})_{\zeta} \max\left(\left(\mathbf{\Psi}^{\mu,\max}_{j,k}\right)_{\zeta}, 0\right)$$

we have for the x-component of velocity

$$-1.20 C^{\mu}_{j,k} + 0.12 C^{m}_{j,k} \geqslant -10.0, \quad -1.20 C^{\mu}_{j,k} + 0.28 C^{m}_{j,k} \leqslant 0$$

and for the y-component of velocity

$$-0.28 C^{\mu}_{j,k} + 0.20 C^{m}_{j,k} \geqslant -12.3, \quad -0.28 C^{\mu}_{j,k} + 0.36 C^{m}_{j,k} \leqslant 12.3.$$

Looking at the interface from cell k, i.e. switching k and i in the formulas and using the fact that $C^{m}_{k,j} = C^{m}_{j,k}$ and $C^{\mu}_{k,j} = C^{\mu}_{j,k}$, we get for the x-component of velocity

$$1.20 C^{\mu}_{j,k} + 0.36 C^{m}_{j,k} \geqslant 0, \quad 1.20 C^{\mu}_{j,k} - 0.28 C^{m}_{j,k} \leqslant 0.45454$$

and for the y-component of velocity

$$0.28 C^{\mu}_{j,k} - 0.20\ C^{m}_{j,k} \geqslant 0, \quad 0.28 C^{\mu}_{j,k} - 0.36 C^{m}_{j,k} \leqslant 0.$$

For interface (k,l), looking from cell k, we have for the x-component of velocity

$$3.86 C^{\mu}_{k,l} - 0.36 C^{m}_{k,l} \geqslant 0, \quad 3.86 C^{\mu}_{k,l} + 0.28 C^{m}_{k,l} \leqslant 2.04545$$

and for the y-component of velocity

$$-0.28 C^{\mu}_{k,l} + 0.20 C^{m}_{k,l} \geqslant -13.3, \quad -0.28 C^{\mu}_{k,l} + 0.36 C^{m}_{k,l} \leqslant 13.3,$$

while looking from cell l and using $C^{m}_{l,k} = C^{m}_{k,l}$ and $C^{\mu}_{l,k} = C^{\mu}_{k,l}$, we get for the x-component of velocity

$$-3.86 C^{\mu}_{k,l} + 0.44 C^{m}_{k,l} \geqslant -35.9, \quad -3.86 C^{\mu}_{k,l} - 0.28 C^{m}_{k,l} \leqslant 0$$

and for the y-component of velocity

$$0.28 C^{\mu}_{k,l} - 0.20\ C^{m}_{k,l} \geqslant 0, \quad 0.28 C^{\mu}_{k,l} - 0.36\ C^{m}_{k,l} \leqslant 0.$$

A.3. Complete set of constraints

For interface (j,k) we have the constraints

$$0 \leqslant C^m_{j,k} \leqslant 1, \quad 0 \leqslant C^\mu_{j,k} \leqslant 1 \tag{A.1}$$

(since density bounds imposed no constraints – recall that all $C^{m,\rho} = 1$) and further

$$C^\mu_{j,k} \leqslant 0.1C^m_{j,k} + 8.3333, \quad C^\mu_{j,k} \geqslant 0.2333\, C^m_{j,k}, \tag{A.2a}$$
$$C^\mu_{j,k} \geqslant -0.3\, C^m_{j,k}, \quad C^\mu_{j,k} \leqslant 0.2333C^m_{j,k} + 0.3788, \tag{A.2b}$$
$$C^\mu_{j,k} \leqslant 0.7143C^m_{j,k} + 43.9286, \quad C^\mu_{j,k} \geqslant 1.2857C^m_{j,k} - 43.9286, \tag{A.2c}$$
$$C^\mu_{j,k} \geqslant 0.7143C^m_{j,k}, \quad C^\mu_{j,k} \leqslant 1.2857C^m_{j,k}, \tag{A.2d}$$

interface of which is shown in the left part of Fig. 12. The maximizing pair is

$$C^m_{k,j} = C^m_{j,k} = 0.7876, \quad C^\mu_{k,j} = C^\mu_{j,k} = 0.5626.$$

For interface (k,l) we have the constraints

$$0 \leqslant C^m_{k,l} \leqslant 1, \quad 0 \leqslant C^\mu_{k,l} \leqslant 1 \tag{A.3}$$

(since density bounds imposed no constraints) and further

$$C^\mu_{k,l} \geqslant 0.09326C^m_{k,l}, \quad C^\mu_{k,l} \leqslant -0.07254C^m_{k,l} + 0.5299, \tag{A.4a}$$
$$C^\mu_{k,l} \leqslant 0.1140\, C^m_{k,l} + 9.3005, \quad C^\mu_{k,l} \geqslant -0.07254\, C^m_{k,l}, \tag{A.4b}$$
$$C^\mu_{k,l} \leqslant 0.7143C^m_{k,l} + 47.5, \quad C^\mu_{k,l} \geqslant 1.2857C^m_{k,l} - 47.5, \tag{A.4c}$$
$$C^\mu_{k,l} \geqslant 0.7143\, C^m_{k,l}, \quad C^\mu_{k,l} \leqslant 1.2857C^m_{k,l}, \tag{A.4d}$$

which are shown in the right portion of Fig. 12 and give the result

$$C^m_{k,l} = C^m_{l,k} = 0.6735, \quad C^\mu_{k,l} = C^\mu_{l,k} = 0.4810.$$

Let us remark, that constraints (A.2c) at left and (A.4c) at right are satisfied but are not visible in the plots because they are too far from the origin $(C^m, C^\mu) = (0, 0)$.

References

[1] J.W. Banks, W.D. Henshaw, J.N. Shadid, An evaluation of the FCT method for high-speed flows on structured overlapping grids, J. Comput. Phys. 128 (2009). DOI: 10.1016/j.jcp.2009.04.03.

[2] D.J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comput. Methods Appl. Mech. Eng. 72 (3) (1989) 305–350.

[3] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (2–3) (1992) 235–394.

[4] J. Boris, D. Book, Flux-corrected transport I: SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11 (1) (1973) (Reprinted in vol. 135(2) (1997) 172–186).

[5] R.B. DeBar, Fundamentals of the KRAKEN code, Technical Report UCIR-760, Lawrence Livermore Laboratory, 1974.

[6] E.D. Dendy, N.T. Padial-Collins, W.B. VanderHeyden, A general-purpose finite-volume advection scheme for continuous and discontinuous fields on unstructured grids, J. Comput. Phys. 180 (2002) 559–583.

[7] J. Donea, S. Giuliani, J.P. Halleux, An arbitrary Lagrangian–Eulerian finite-element method for transient dynamic fluid structure interaction, Comput. Methods Appl. Mech. Eng. 33 (1–3) (1982) 689–723.

[8] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) (Reprinted in vol. 135(2) (1997) 203–216).

[9] J.R. Kamm, F.X. Timmes, On efficient generation of numerically robust Sedov solutions, Technical Report LA-UR-07-2849, Los Alamos National Laboratory, Los Alamos, NM, USA, 2007.

[10] D.S. Kershaw, M.K. Prasad, M.J. Shaw, J.L. Milovich, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method, Comput. Methods Appl. Mech. Eng. 158 (1–2) (1998) 81–116.

[11] P. Kjellgren, J. Hyvarinen, An arbitrary Lagrangian–Eulerian finite element method, Comput. Mech. 21 (1) (1998) 81–90.

[12] P.M. Knupp, L.G. Margolin, M.J. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 176 (1) (2002) 93–128.

[13] M. Kuchařík, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, J. Comput. Phys. 188 (2) (2003) 462–471.

[14] D. Kuzmin, R. Löhner, S. Turek (Eds.), Flux-Corrected Transport. Principles, Algorithms and Applications, Springer Verlag, Berlin, Heidelberg, 2005.

[15] R. Löhner, Applied CFD Techniques. An Introduction Based on Finite Element Methods, John Wiley & Sons, 2001 (Chapter 9).

[16] R. Loubère, M.J. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 209 (1) (2005) 105–138.

[17] H.U. Mair, Review: hydrocodes for structural response to underwater explosions, Shock Vib. 6 (2) (1999) 81–96.

[18] P.-H. Maire, A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (7) (2009) 2391–2425.

[19] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, SIAM J. Sci. Comput. 29 (4) (2007) 1781–1824.

[20] P.-H. Maire, J. Breil, S. Galera, A cell-centered arbitrary Lagrangian–Eulerian (ALE) method, Int. J. Numer. Methods Fluids 56 (8) (2008) 1161–1166. DOI 10.1002/fld.1557.

[21] L.G. Margolin, Introduction to An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 135 (2) (1997) 198–202.

[22] L.G. Margolin, M. Shashkov, Remapping, recovery and repair on a staggered grid, Comput. Methods Appl. Mech. Eng. 193 (2004) 4139–4155.
[23] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298.
[24] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Methods Appl. Mech. Eng. 187 (3–4) (2000) 591–619.
[25] C. Schär, P.K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations, J. Comput. Phys. 128 (1) (1996) 101–120.
[26] J. Thuburn, Multidimensional flux-limited advection schemes, J. Comput. Phys. 123 (1) (1996) 74–83.
[27] Q.H. Tran, Second-order slope limiters for the simultaneous linear advection of (not so) independent variables, Commun. Math. Sci. 6 (2008) 569–593.
[28] P. Váchal, R. Liska, Sequential flux-corrected remapping for ALE methods, in: A. Bermúdez de Castro, D. Gómez, P. Quintela, P. Salgado (Eds.), Numerical Mathematics and Advanced Applications, ENUMATH 2005, Springer Berlin Heidelberg New York, 2006, pp. 671–679.
[29] W.B. VanderHeyden, B.A. Kashiwa, Compatible fluxes for van Leer advection, J. Comput. Phys. 146 (1) (1998) 1–28.
[30] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.