Journal of Computational Physics 255 (2013) 590–611

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Symmetry- and essentially-bound-preserving flux-corrected remapping of momentum in staggered ALE hydrodynamics

J. Velechovský [a,*], M. Kuchařík [a], R. Liska [a], M. Shashkov [b], P. Váchal [a]

[a] Czech Technical University in Prague, FNSPE, Břehová 7, Praha 1, 115 19, Czech Republic
[b] Los Alamos National Laboratory, XCP-4, MS-F644, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 20 March 2013
Received in revised form 5 August 2013
Accepted 18 August 2013
Available online 30 August 2013

Keywords:
Staggered hydrodynamics
ALE
Remap of vectors
Symmetric reconstruction

ABSTRACT

We present a new flux-corrected approach for remapping of velocity in the framework of staggered arbitrary Lagrangian–Eulerian methods. The main focus of the paper is the definition and preservation of coordinate invariant local bounds for velocity vector and development of momentum remapping method such that the radial symmetry of the radially symmetric flows is preserved when remapping from one equiangular polar mesh to another. The properties of this new method are demonstrated on a set of selected numerical cyclic remapping tests and a full hydrodynamic example.

1. Introduction

In the simulations of fluid flows, various numerical methods can be used. These methods have been traditionally referred to either as Lagrangian or Eulerian, depending on the treatment of the computational mesh in time, each with its own advantages and disadvantages. In a pioneering paper [10], Hirt et al. developed the formalism for a mesh, motion of which can be determined as an independent degree of freedom, and showed that this general framework could be used to combine the best properties of the Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian–Eulerian or ALE. This approach is currently very popular and many researchers have contributed to this topic [2,27,31,1,29,19,9].

The ALE scheme is typically subdivided into three distinct phases: (1) a Lagrangian phase, in which the solution and the computational mesh are advanced in time; (2) a rezoning phase, in which the nodes of the computational mesh are moved to improve the mesh quality; and (3) a remapping stage, in which the Lagrangian solution is conservatively interpolated (remapped) onto the rezoned mesh. Here we focus on the last stage of the ALE algorithm – remapping. In this paper we use the staggered compatible discretization on logically rectangular meshes, where discrete values of density, internal energy and pressure are assigned to the computational cells (cell-centered quantities), while velocity $\boldsymbol{w}$ belongs to the nodes [5]. Remapping of cell-centered conservative quantities (mass, for example) in this context is described in detail elsewhere, [8,28,15]. Usually, the remapping of mass is done in such way that the physically justified bounds for density are satisfied – for example, remapped density in a particular cell is supposed to be in the bounds defined by maximum and minimum values of the density in the neighboring cells before remap. When the remapped density is not in bounds, one can put it into bounds by the repair technique [35,20]. In this paper, we are specifically interested in the remap of nodal velocity. An overview of traditional methods can be found in the seminal review paper by Benson [2].

* Corresponding author.
E-mail address: velecjan@fjfi.cvut.cz (J. Velechovský).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.08.037

There are several requirements for the remapping of velocity. First of all, it is the conservation of total momentum. It implies that nodal momentum is remapped and then nodal velocity is recovered as a ratio of remapped nodal momentum and remapped nodal mass, that is, the remapped velocity is a derived quantity. This also brings the question how to remap nodal mass in a way consistent with remapping of cell mass [14,12]. Another requirement, which is often called DeBar condition [7], is the requirement that if velocity is constant, then it is supposed to be remapped exactly for arbitrary density distribution. The DeBar condition is usually satisfied by constructing the momentum flux as a product of corresponding mass flux by some reconstructed velocity. Therefore, for given inter-nodal mass fluxes, the momentum (and velocity) remap is completely defined by the velocity in the momentum flux.

Another important question is how to define bound preservation for velocity. The traditional way is to define bounds for Cartesian components of the velocity vector. However, it was recognized that such definition is not coordinate invariant and, in particular, gives non-symmetric bounds (for example, different bounds for radial velocity vector on an equiangular polar mesh for nodes with the same radius but different angles).

A very interesting new idea, which was suggested in the context of vector field reconstruction, defines the bounds by using the Vector Image Polygon (VIP) [21] constructed as the convex hull of velocity vectors in the plane, where for radial flow on the axis one has the Cartesian components of the velocities and velocities in the neighboring vertices are represented by points in this plane. Such convex polygon can be quite complicated in shape. This approach employs a constrained-optimization procedure which allows to satisfy this type of constraints in the context of reconstruction. We are exploring this approach in the symmetric remapping context in [37]. The VIP approach has been already applied to remapping in [23,22].

In the framework of cell-centered Lagrangian discretization and vector field reconstruction inside the cell it was suggested to use bounds related to the projection of the velocity vector to principal axes of deformation tensor direction [24–26], that is, to use directions related to the flow.

In our paper we a use similar idea to define coordinate invariant bounds at mesh points. We project all velocity vectors in neighboring points into the direction of velocity at the point under consideration and the direction orthogonal to it – this is a linear transformation, achieved by applying rotation matrix to the velocity vectors; then we define bounds using these projections. In particular, it allows us to obtain the same bounds for radial velocity vector on an equiangular polar mesh for nodes with the same radius but different angles. This is a necessary condition for preservation of the spatial symmetry when remapping from one equiangular polar mesh to another. The preservation of this type of spatial symmetry is crucial for certain types of applications, such as those related to inertial confinement fusion (ICF). Most of these symmetry requiring applications are defined in the axisymmetric cylindrical geometry. In this paper we however stay in Cartesian geometry, making the first step concerning symmetry preservation. The generalization of our approach to cylindrical geometry will be the next step.

To enforce the defined bound preservation we use a non-trivial modification of the Flux-Corrected Remapping (FCR) (see, for example, [17]), which combines low- and high-order momentum fluxes obtained from the corresponding low- and high-order reconstruction of the velocity vector.

The remaining text is organized in the following way. The problem treated in this paper is defined in Section 2. The bounds for the velocity vector are defined in Section 3. The velocity reconstruction methods used are reviewed in Section 4. The main contribution of this paper, the symmetry preserving flux-corrected remapping algorithm, is presented in Section 5. Numerical examples are shown in Section 6. Three appendices contain the proof of symmetry of the piecewise linear velocity reconstruction, the proof of velocity bounds preservation for low-order velocity remap and the proof of symmetry of flux-corrected velocity remap.

2. Problem statement

Assume that we have two computational meshes – old (Lagrangian) mesh $\{c\}$ and new (rezoned) mesh $\{\tilde{c}\}$. Further also assume that both meshes have the same connectivity and are close to each other (meaning that each new cell $\tilde{c}$ stays inside the region defined by old cell c and its immediate neighbors, including corner neighbors, in the old mesh). The remap of cell mass can be written in the following flux form

$$\widetilde{m}_c = m_c + \sum_{c' \in C'(c)} F^m_{c,c'}, \tag{1}$$

where $C'(c)$ represents the set of all cells neighboring to c (including corners) and $F^m_{c,c'}$ stands for the (outwards oriented) mass flux from c to c', so $F^m_{c,c'} = -F^m_{c',c}$ as shown in Fig. 1(a). Depending on the particular remapping method, the $F^m_{c,c'}$ fluxes can be constructed in various ways. For example, they can be zero for corner neighbors, while for the edge neighbors the swept-region approach [15] is used. Another possibility is the integration of the density function over all intersections of the original cell c with all its neighbors in the new mesh and vice versa, see [8,28] for more details. This approach is more computationally expensive, however, unlike the swept region approach, it can be directly generalized for multi-material flows. Eventually, one can combine both approaches in one of the hybrid ways described in [3,11,13], requiring to construct the expensive intersections only in the vicinity of material interfaces, while computationally cheap swept fluxes are computed in pure material regions covering typically most of the computational domain. For the rest of the paper it does not