Computer Physics Communications 210 (2017) 79–91

Contents lists available at ScienceDirect

Computer Physics Communications

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

A new Particle-in-Cell method for modeling magnetized fluids

Fabio Bacchini *, Vyacheslav Olshevsky, Stefaan Poedts, Giovanni Lapenta

Centre for Mathematical Plasma-Astrophysics, Department of Mathematics, KU Leuven, Celestijnenlaan 200B, 3000 Leuven, Belgium

ARTICLE INFO

Article history:
Received 9 March 2016
Received in revised form
28 September 2016
Accepted 7 October 2016
Available online 14 October 2016

Keywords:
MHD
PiC methods
Ringing instability
Vector potential
Moving mesh

ABSTRACT

We present a new Particle-in-Cell method for plasma simulations. This is based on the original algorithm of FLIP-MHD, which uses a Lagrangian formulation of the macroscopic equations. A finite-difference approximation of the equations of motion is solved on a fixed (non-moving) grid, while convection of the quantities is modeled with the support of Lagrangian particles. Interpolation with first-order b-splines is used to project the conserved quantities from particles to the grid and back. In this work, we introduce two modifications of the original scheme. A particle volume evolution procedure is adopted to reduce the computational error, based on the formulation used in the Material Point Method for computational mechanics. The additional step introduces little to none computational diffusion and turns out to efficiently suppress the so-called ringing instability, allowing the use of explicit time differencing. Furthermore, we eliminate the need for a Poisson solver in the magnetic field computation with the use of a vector potential in place of the particles' magnetic moment. The vector potential evolution is modeled with a moving grid and interpolated to the fixed grid points at each time step to obtain a solenoidal magnetic field. The method is tested with a number of standard hydrodynamic and magnetohydrodynamic tests to show the efficiency of the new approach. The results show good agreement with the reference solutions and rather fast time and space convergence.

1. Background

Computer models of various plasmas, including space and astrophysical plasmas, either employ computational meshes, or are mesh-free. The former models are widely used in astrophysical simulations [1] for historical and numerical reasons: they are, in general, less computational expensive than mesh-free ones. The latter have much higher memory and CPU requirements because they exploit particles (finite elements) of some sort. Modeling of the same blob of plasma would normally use much more particles than mesh cells, to reduce noise and maintain the stability of the solution. For many applications such as plasmas with high density gradients or where Lagrangian treatment is advantageous, purely mesh-based codes are inappropriate. These applications include simulations of galaxy dynamics, intergalactic gas and dust dynamics, stellar ejecta simulations, etc.

Among completely grid-less particle methods Smooth Particle Hydrodynamics (SPH, [2]) finds more and more applications in astrophysics and beyond [3,4]. Pure particle methods have an obvious disadvantage: the output of the simulation (e.g., flow field or magnetic field) is represented by a huge number of irregularly spaced data points. A comprehensive scientific analysis of such results needs cumbersome postprocessing and is demanding to the storage space.

Purely grid-based Lagrangian methods struggled to produce an adequate solution for problems involving large grid distortions until a compromise between Eulerian and Lagrangian representation, Arbitrary Lagrangian–Eulerian (ALE) was proposed by Hirt et al. [5]. In an ALE simulation, conservation equations are advanced on a Lagrangian grid which becomes distorted. The distorted Lagrangian grid is rezoned, and conservative quantities are interpolated (remapped) to the regular grid in a conservative way. Optimal rezoning strategy and remapping methods are great computational challenges, but despite these complexities and a number of issues [6,7], ALE methods have found their applications in simulations where delineations of fluid surfaces are important [8].

A Particle-in-Cell (PIC) method using both particles and grid was introduced by Harlow [9]. Particles in a PIC simulation are proxies for the grid cells' exchange of conserved quantities. The finite-size particles are moved through the grid with the area-weighted mean of the velocities from the neighboring grid elements. An evolution of the PIC method, cloud-in-cell (CIC) by Langdon and Birdsall [10], computes the movement of charged

* Corresponding author.
E-mail addresses: fabio.bacchini@wis.kuleuven.be (F. Bacchini), vyacheslav.olshevsky@wis.kuleuven.be (V. Olshevsky), stefaan.poedts@wis.kuleuven.be (S. Poedts), giovanni.lapenta@wis.kuleuven.be (G. Lapenta).

http://dx.doi.org/10.1016/j.cpc.2016.10.001
0010-4655/

plasma particles self-consistently, via the electromagnetic forces acting on them. The electromagnetic field equations are solved on the grid, and the forces are interpolated on the particles. This method, excellent for kinetic modeling of collisionless plasmas, has found its wide applications in many astrophysical simulations, including space weather modeling [11].

A Fluid Implicit particle-in-cell (FLIP) strategy proposed by Brackbill and Ruppel [12], and later extended to MHD [13] and beyond (e.g. fusion-related plasma–wall interactions [14]), aims at overcoming a number of issues of the traditional PIC methods. In particular, use of higher-order interpolation functions for projecting from the grid to particles, and implicit time stepping help in suppressing a ringing (finite-grid) instability [15]. Application of FLIP methodology to solid mechanics produced its most successful child, a Material Point Method (MPM) [16–18]. It is now widely applied in geophysics [19,20], engineering [21] and 3D graphics [22].

Despite the popularity of CIC and MPM methods, the applications of FLIP to astrophysical plasma simulations are rather limited. In our opinion, two main reasons cause such misjudgment: (1) complexity of the magnetic field treatment, and (2) unavoidable ringing instability. We have developed a new algorithm that aims at overcoming these issues, it is robust, easy to implement and is very flexible in applications. In the following chapter we introduce the FLIP method in more detail, and further explain our modifications of the original strategy. Two main additions: particle volume evolution and magnetic field evolution are discussed in great details in Sections 3 and 4, correspondingly. The algorithm has been tested on standard problems: sound wave propagation, Kelvin–Helmholtz instability, Orszag–Tang problem, and magnetic loop problem.

2. The FLIP and FLIP-MHD methods

The original FLIP-MHD is designed to solve the set of Lagrangian, visco-resistive MHD equations. In this work, we neglect all dissipative terms, focusing on the ideal MHD framework given by the continuity, momentum and internal energy equations

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{u}, \tag{1}$$

$$\rho\frac{d\mathbf{u}}{dt} = -\nabla\left(p+\frac{\mathbf{B}^2}{2}\right)+\nabla\cdot\mathbf{BB}, \tag{2}$$

$$\rho\frac{de}{dt} = -p\nabla\cdot\mathbf{u}, \tag{3}$$

and the induction equation

$$\rho\frac{d}{dt}\left(\frac{\mathbf{B}}{\rho}\right) = \mathbf{B}\cdot\nabla\mathbf{u}. \tag{4}$$

Here, ρ is the mass density, $\mathbf{u}$ is the bulk velocity, p is the pressure, $\mathbf{B}$ is the magnetic field, and e is the specific internal energy. All tests described in this work use non-dimensional units, thus may be scaled to any relevant physical systems. Lengths, magnetic field and plasma density are normalized to characteristic scales of the corresponding system L, B_0, and ρ_0, respectively. Time is normalized by the Alfvén time, defined as L/V_A, with $V_A = B_0/\sqrt{\mu_0\rho_0}$. The ratio of gas/magnetic pressure, the so-called plasma beta $\beta_0 = 2\mu_0 p_0/B_0^2$, relates the typical values of magnetic field and pressure, p_0. If the non-dimensionless pressure is normalized as $(\beta_0/2)p/p_0$, then the above form of the MHD equations is retrieved, without explicit μ_0 terms, present in the dimensional form of the equations.

Artificial viscosity can be added to the equations in order to smear shock waves, should it be necessary. However, in this work we focus on tests involving rather smooth flows and therefore we set any viscous term to 0. The system is closed with the ideal gas law $p = e(\gamma-1)\rho$, where γ is the specific heat ratio.

The algorithm solves the macroscopic equations by relying on a computational grid made of N_C cells. The variables are advanced on the grid by solving a finite difference discretization of the set of equations. It is common choice to employ a staggered discretization of the partial derivatives in space. In FLIP, the computational grid includes grid elements of type cell (labeled C) and node (labeled N), with the latter being the corners surrounding each cell center. The grid variables are centered on either of the two: mass, specific internal energy, and magnetic field to cells, velocity to nodes. In this way, the spatial derivatives of the node quantities (centered on cells) drive the time evolution of the cell quantities; conversely, the derivatives of cell quantities (centered on nodes) drive the evolution of node quantities. The explicit discretization used in the algorithm is illustrated in Appendix A.

The properties of the grid are initialized, at each time step, by interpolating from $N_p \gg N_C$ particles which model the advective part of the evolution of the system. Each particle carries a fraction of the conserved quantities, mass m, momentum $m\mathbf{u}$, and internal energy $\epsilon = me$. The interpolation projects the particle contribution (subscript p) to the grid (subscripts C and N) according to

$$m_C = \sum_p m_p W_{Cp}, \tag{5}$$

$$(m\mathbf{u})_N = \sum_p m_p\mathbf{u}_p W_{Np}, \tag{6}$$

$$(me)_C = \sum_p \epsilon_p W_{Cp}, \tag{7}$$

where W_{Cp} and W_{Np} are freely chosen interpolation functions relating the grid and the particles. A simple choice for the interpolation function is the first-order b-spline, which in one dimension is given by

$$W_x = \begin{cases} 1+\dfrac{x_g-x_p}{\Delta x} & \text{if} \quad -\Delta x < x_g-x_p < 0 \\ 1-\dfrac{x_g-x_p}{\Delta x} & \text{if} \quad 0 \le x_g-x_p < \Delta x \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where the subscript g indicates a grid element, either a node or a cell center. In two dimensions, the interpolation function is the product of the one-dimensional b-splines, such that $W_{xy} = W_x W_y$.

After the Lagrangian step, the particle properties are updated according to the inverse interpolations

$$\frac{d\mathbf{u}_p}{dt} = \sum_N \frac{d\mathbf{u}_N}{dt} W_{pN}, \tag{9}$$

$$\frac{d\epsilon_p}{dt} = m_p \sum_C \frac{de_C}{dt} W_{pC}. \tag{10}$$

Finally, the position of the particles is updated by solving $d\mathbf{x}_p/dt = \mathbf{u}_p$. One immediate advantage of using a large number of particles is the ability to model the advection of the macroscopic quantities much more precisely, while introducing very little numerical diffusion due to the interpolation steps. At the same time, solving the system of equations in Lagrangian form relaxes the Courant–Friedrichs–Lewy (CFL) condition to a wider range of values for the time step Δt.

The FLIP algorithm is designed in a way that preserves the shape of the computational grid while relying on the particles to compute the movement of the fluid parcels. Without taking into account the grid motion, it is assumed that the grid cells maintain the initial volume throughout the computation, by interpolating the particle

contribution over the same grid at each time step. However, in Lagrangian schemes, this is not true in general: the change in the volume V_C of each cell is determined by

$$\frac{dV_C}{dt} = V_C(\nabla \cdot \mathbf{u})_C. \tag{11}$$

This is neglected, at each time step, by computing the density as

$$\rho_C = \frac{\sum_p m_p W_{Cp}}{V_C^0}, \tag{12}$$

where $V_C^0 = \Delta x \Delta y$ is the volume of each grid cell at the beginning of the computation.

In the MHD extension of the original algorithm, the magnetic field effects are modeled by assigning a magnetic moment property $\boldsymbol{\mu}_p$ to the particles. This is projected to the grid to compute the magnetization, $\mathbf{M}$ (a cell quantity), from

$$\mathbf{M}_C V_C = \sum_p \boldsymbol{\mu}_p W_{Cp}. \tag{13}$$

The magnetization is related to the magnetic field via a scalar potential ϕ (a node quantity) according to

$$\mathbf{B} = \mathbf{M} - \nabla\phi. \tag{14}$$

This relation allows one to compute, at each time step, a divergence-free magnetic field by solving the Poisson equation $\nabla^2\phi = \nabla \cdot \mathbf{M}$. After this step, the magnetic field can be updated according to the induction equation and then used to advance the magnetization via

$$\frac{d\mathbf{M}}{dt} = \frac{d\mathbf{B}}{dt} - \nabla\mathbf{u} \cdot \nabla\phi, \tag{15}$$

where the scalar potential has been chosen so that it respects $d\phi/dt = 0$. Finally, the magnetic moment of the particles is updated via the inverse interpolation

$$\frac{d\boldsymbol{\mu}_p}{dt} = m_p \sum_C \frac{d}{dt}\left(\frac{\mathbf{M}}{\rho}\right)_C W_{pC}. \tag{16}$$

Because of the use of a scalar potential, the resulting magnetic field is solenoidal and therefore no divergence cleaning procedure is necessary to respect the divergence free condition. However, at the same time the algorithm introduces the need for solving a Poisson equation at each time step. This is a well-studied problem for which several numerical schemes are available, but in general this step can be computationally expensive, to the point where it represents the highest portion of the simulation execution time. Therefore, it is tempting to find an alternative strategy and avoid such expense, to significantly improve the performance of the code.

The FLIP algorithm has been validated with a number of HD, MHD and non fluid-related tests (see e.g., [23–25]) and refined to achieve better accuracy [26]. It has been shown that the method well preserves conserved quantities (momentum, energy [12]; magnetic flux and energy [13]; angular momentum [27]) and can be applied to model complex fluid–fluid interactions (e.g. surface tension-driven problems [28]). In this work, we try to further improve the method performance by both implementing a strategy to track the volume evolution of the grid and suggesting alternatives to the need for a Poisson solver in modeling the magnetic field effects. For this purpose, through all of the next sections we focus on the 2D version of the ideal MHD equations.

3. Particle volume evolution

We attempt to give a more accurate description of the Lagrangian fluid and reduce the errors on all the macroscopic quantities by introducing a dedicated volume evolution procedure. A modification of the original algorithm is needed to account for the variation of the cell volume throughout the computation. Eq. (12) can be substituted with

$$\rho_C = \frac{\sum_p m_p W_{Cp}}{\sum_p V_p W_{Cp}} \tag{17}$$

where V_p is a new particle property, a "volume" which represents the physical space occupied by a particle on the domain. At the beginning of each computational cycle, the particle information carried by V_p is introduced in the value of the cell density.

It is straightforward that, at the end of the Lagrangian phase, the particle volume has to be updated, together with all the other particle properties, which adds a certain complexity to the algorithm.

The problem of volume evolution has been extensively addressed in computational mechanic simulations and solid body dynamics. These studies have proven the Material Point Method (MPM) to be a rather simple but efficient way to evolve particle volumes in the simulation. A formal development of the MPM method has been given by Love and Sulsky [29]. This has been applied with several extensions to a number of computational mechanics problems [30,31].

The strategy is as follows: at the beginning of the computation, each particle is assigned a volume $V_p^0 = \Delta x_p^0 \Delta y_p^0$, whose dimensions are Δx_p^0 and Δy_p^0. Assume all particle volume to have a shape of square initially; for each cell, it is valid $V_p^0 = V_C^0/N_{ppC}$, where N_{ppC} is the number of particles each cell C. Hence, each particle initially has the dimensions $\Delta x_p^0 = \Delta y_p^0 = \sqrt{V_C^0/N_{ppC}}$. At each computational cycle (after the Lagrangian phase), a particle velocity gradient tensor is built, according to

$$\nabla\mathbf{u}_p = \sum_N \nabla W_{pN}\mathbf{u}_N \tag{18}$$

where, recalling that in two dimensions $W_{pN} = W_{pN}^x W_{pN}^y$, the gradient of the interpolation function is readily

$$\nabla W_{pN} = \begin{bmatrix} W_{pN}^y \dfrac{\partial W_{pN}^x}{\partial x} \\ W_{pN}^x \dfrac{\partial W_{pN}^y}{\partial y} \end{bmatrix}. \tag{19}$$

Therefore, the particle velocity gradient tensor is

$$\nabla\mathbf{u}_p = \sum_N \begin{bmatrix} u_N W_{pN}^y \dfrac{\partial W_{pN}^x}{\partial x} & v_N W_{pN}^y \dfrac{\partial W_{pN}^x}{\partial x} \\ u_N W_{pN}^x \dfrac{\partial W_{pN}^y}{\partial y} & v_N W_{pN}^x \dfrac{\partial W_{pN}^y}{\partial y} \end{bmatrix}, \tag{20}$$

which can be used to multiplicatively update the particle deformation gradient

$$J_p^{n+1} = (\mathbf{I} + \Delta t \nabla\mathbf{u}_p^{n+1}) J_p^n, \tag{21}$$

which is initially set as $J_p^0 = \mathbf{I}$. The new values of the particle volume are found by simply computing the determinant, $J_p = |J_p|$, according to

$$V_p^{n+1} = V_p^0 J_p^{n+1}. \tag{22}$$

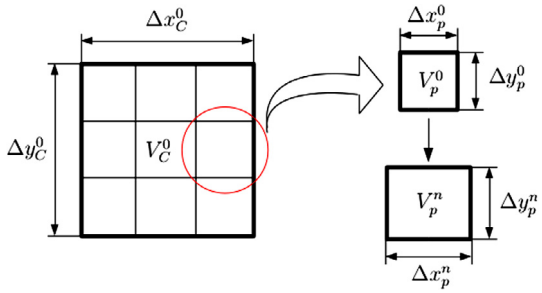


Fig. 1. Schematic representation of the particle volume evolution, from the initial value V_p^0 to the updated V_p^n.

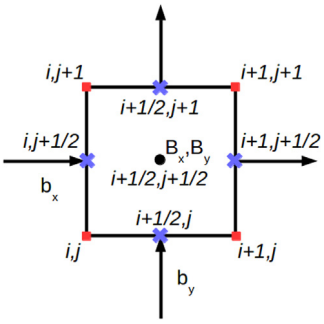


Fig. 2. Reference scheme for the edge-centered quantities.

Particle dimensions can be updated as

$$\begin{Bmatrix} \Delta x_p^{n+1} \\ \Delta y_p^{n+1} \end{Bmatrix} = \begin{bmatrix} j_{p,xx}^{n+1} & 0 \\ 0 & j_{p,yy}^{n+1} \end{bmatrix} \begin{Bmatrix} \Delta x_p^0 \\ \Delta y_p^0 \end{Bmatrix}. \tag{23}$$

The process is sketched in Fig. 1.

The proposed volume evolution strategy can involve both diagonal and off-diagonal components of the deformation gradient. Widely employed MPM, later evolved to Generalized Interpolation MPM (GIMP) schemes, however, neglect the off-diagonal elements thus retaining rectangular shapes of computational particles. Manipulation of non-rectangular particles significantly complicates the algorithm and slows down the execution. Specific investigations [30] have shown that many applications do not suffer at all from the assumption that particles are kept rectangular. In this work, we limit ourselves to this approach. However, this is not a limitation of our algorithm and will be studied in subsequent works.

As will be shown below (Section 5.1) this scheme drastically improves the overall accuracy of the method and suppresses the growth of the finite grid instability, while keeping the algorithm highly efficient.

4. Strategies for magnetic field evolution and transport

We already mentioned that delegating magnetic field property from particles to grid is a significant modification to the Flip method, where particles used to carry the so-called magnetic moment. This means that the magnetic field can be treated as it would be in any standard, non particle-related code for MHD, as long as its evolution remains consistent with the description of the other macroscopic quantities provided by the particles.

The main issue with this step is the conservation of the divergence-free condition $\nabla \cdot \mathbf{B} = 0$. A number of techniques suitable for this purpose are available, and adopted in many codes for MHD. Here, we choose to implement a strategy that allows to respect the divergence-free constraint as much as possible, that is, keeping the numerical value of $\nabla \cdot \mathbf{B}$ down to the limit of machine precision. For this purpose, we compare constrained transport (CT), evolution of vector potential (VP), and the FLIP-MHD original strategy relying on the particle magnetic moment.

4.1. Constrained transport on a fixed Eulerian frame

The constrained transport scheme was first introduced in [32]. As discussed in [33], the strategy relies on a staggered representation of the magnetic and electric fields in the numerical scheme. The induction equation

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times \mathbf{E}, \tag{24}$$

in two dimensions, relates the two evolving components of $\mathbf{B}$ only to the z component of the electric field $\mathbf{E}$, such that $B_x = -\partial E_z/\partial y$ and $B_y = \partial E_z/\partial x$. Such component can be obtained from the relation $\mathbf{E} = -\mathbf{u} \times \mathbf{B}$; because of the choice of cell-centering for the magnetic field discretization, the electric field is discretized as a node quantity, like the velocity $\mathbf{u}$.

Let $(i + 1/2, j + 1/2)$ be the indices of a generic cell center of the 2D domain; then, the nodes surrounding it are given by the possible combinations of the indices $(i+k, j+k)$ with $k = 0, 1$. With this discretization, the location of the edge-centered, x-component of $\mathbf{B}$ will be represented by $(i + k, j + 1/2)$ (see Fig. 2).

The initial values for the edge-centered magnetic field components can be computed as the average of the two adjacent cells sharing the chosen edge. During the computation, the edge-centered magnetic fields are consistently evolved; the cell-centered magnetic field is not evolved, but obtained by averaging the edge-centered values. The evolution of the edge-centered components is given by

$$b_{x,i+1/2,j+k}^{n+1} = b_{x,i+k,j+1/2}^{n} - \Delta t \frac{E_{z,i+k,j+1}^{n} - E_{z,i+k,j}^{n}}{\Delta y}, \tag{25}$$

$$b_{y,i+k,j+1/2}^{n+1} = b_{y,i+1/2,j+k}^{n} + \Delta t \frac{E_{z,i+1,j+k}^{n} - E_{z,i,j+k}^{n}}{\Delta x}. \tag{26}$$

It can been shown that CT schemes keep $\nabla \cdot \mathbf{B}$ to machine precision (see [33]). Therefore, it is sufficient to define a divergence-free magnetic field at the beginning of the computation to automatically satisfy the constraint for the rest of the current run. However, the technique is designed to be applied on a fixed, Eulerian frame (extension to moving grids have been attempted by Mocz et al. [34]). On a Lagrangian frame, an advective term is required in the induction equation, nullifying the usefulness of the staggered scheme (since the terms in the expression of the numerical $\nabla \cdot \mathbf{B}$ do not cancel out exactly any longer). Apparently, such a scheme cannot be compatible with the FLIP strategy, which relies on a Lagrangian grid.

The advection of the fluid quantities in the FLIP approach is provided by the moving particles. However, this is not the case with the magnetic field in our formulation. The magnetic field (evolved purely on the grid) is not advected. In some sense, direct application of the CT method evolves the magnetic field in a Eulerian way, missing the particle phase. Such misalignment is prominent in specific test cases involving predominant magnetic diffusion (see the magnetic loop advection test in Section 5). The differences between the FLIP scheme and a fixed Eulerian viewpoint are shown schematically in Fig. 3.

4.2. Vector potential on a moving Lagrangian frame

An alternative formulation of the CT scheme is based on a vector potential $\mathbf{A}$ such that $\mathbf{B} = \nabla \times \mathbf{A}$; it is straightforward that $\nabla \cdot \mathbf{B} =$

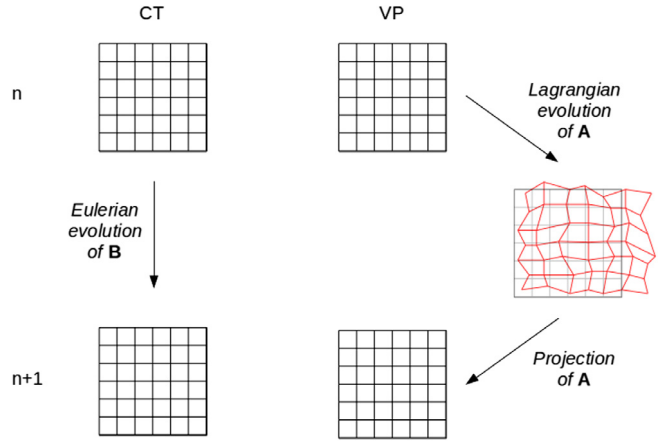


Fig. 3. Comparison between the Eulerian and Lagrangian frames for the evolution of the magnetic field, from the time level n to $n+1$. Left: in the CT scheme, the magnetic field is directly evolved on a fixed Eulerian grid. Right: in the VP scheme, the vector potential is first evolved on a Lagrangian moving grid, and then interpolated on the fixed frame at the initial position.

$\nabla \cdot (\nabla \times \mathbf{A}) = 0$, thus the divergence-free condition is always satisfied.

The Lagrangian evolution of the vector potential is given by

$$\frac{d\mathbf{A}}{dt} = -\mathbf{E} + (\mathbf{u}\cdot\nabla)\mathbf{A} = \mathbf{u}\times\mathbf{B} + (\mathbf{u}\cdot\nabla)\mathbf{A}, \tag{27}$$

where in pure 2D the only evolving components of the magnetic field are computed as

$$B_x = \frac{\partial A_z}{\partial y}, \tag{28}$$

$$B_y = -\frac{\partial A_z}{\partial x}, \tag{29}$$

meaning that the only relevant component of the vector potential is A_z. Additionally, by definition, $dA_z/dt = 0$ which implies that the values of A_z defined on a Lagrangian grid do not change over time. The VP formulation is, on a Eulerian frame, analogous to the CT scheme described above. Here, however, it is more consistent with the nature of FLIP, and adds the required advection to the Eulerian evolution of the magnetic field. While the other macroscopic quantities are evolved on the fixed grid after the particle interpolation phase, the vector potential can be evolved on a separate Lagrangian grid, and the particle phase of FLIP can be substituted by a procedure to obtain the values of the vector potential on the initial fixed grid. Differentiating such values then gives the necessary magnetic field on the fixed grid to be used at the next computational cycle. The procedure is represented in Fig. 3. It can be shown that, with our choice of discretization, this ensures $\nabla\cdot\mathbf{B} = 0$ to machine precision (see Appendix B).

In the context of ideal 2D ideal MHD, the additional advantage of a non-evolving vector potential is fully exploited. This, however, should not be looked at as a limitation of the algorithm. In a full 3D simulation, the vector potential will evolve on the moving grid as $d\mathbf{A}/dt = (\nabla\mathbf{A})\cdot\mathbf{u}$. Hence, at each time step gradients of $\mathbf{A}$ have to be evaluated on the distorted grid. This is the only additional complication. Evaluation of gradients on unstructured grids is an elaborate task, which, however, has been addressed in dedicated literature (see e.g., [35–37]), and can be implemented in the code.

The change of frame of reference, from the evolved Lagrangian grid to the fixed initial grid, is the step that substitutes the particle phase of the FLIP method, and therefore provides the necessary description of the magnetic energy transport. Thus, it has to be as accurate as possible while minimizing the amount of dissipation it brings in the system. There are indeed several options to perform this operation, and it is necessary to identify which one gives the best results both in terms of energy conservation and computational efficiency.

4.2.1. Initial vector potential definition

First of all, it is necessary to define the initial vector potential at the beginning of the computation. While for many standard tests such value is immediately available, this is not true in general. Usually, the simulation settings involve a definition of the initial magnetic field, but not of the relative vector potential. Rambo [38] presented a formulation of the vector potential relying on the assumption that, within each cell, the magnetic field varies linearly with the distance from the cell center. The resulting vector potential on the four nodes N surrounding the cell center C is given by

$$\mathbf{A}_N = \mathbf{B}_C \times \mathbf{x}_N + \frac{1}{2}[(\mathbf{x}_N - \mathbf{x}_C)\cdot(\nabla\mathbf{B})_C]\times(\mathbf{x}_N - \mathbf{x}_C). \tag{30}$$

4.2.2. Continuous grid regeneration

The first approach for the vector potential evolution is somehow to mimic the particle phase with an analogous procedure. Consider the computational grid, fixed at the initial position. After the Lagrangian phase, the grid has moved according to the equation of motion

$$\frac{d\mathbf{x}_N}{dt} = \mathbf{u}_N, \tag{31}$$

whose discretization yields

$$x_{N'}^{n+1} = x_N^n + \Delta t u_N^n, \tag{32}$$

$$y_{N'}^{n+1} = y_N^n + \Delta t v_N^n, \tag{33}$$

where the subscript N' indicates the Lagrangian nodes moved from the fixed initial positions labeled as N. The values on the fixed grid are obtained by simple interpolation as

$$A_{z,N}^{n+1} = \sum_{N'} A_{z,N'}^{n+1} W_{NN'}, \tag{34}$$

where $A_{z,N'}^{n+1} = A_{z,N'}^n$. The magnetic field on the fixed grid is given by the discretized version of Eqs. (28) and (29) above (see Appendix B for details). Finally, the Lagrangian grid can be regenerated at the initial position by simply assigning $\mathbf{x}_{N'}^{n+1} = \mathbf{x}_N^{n+1}$ and $A_{z,N'}^{n+1} = A_{z,N}^{n+1}$.

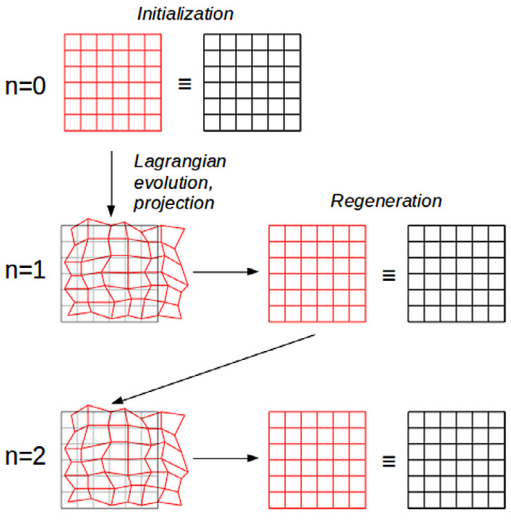
The whole procedure is described in Fig. 4(a). Clearly, the main disadvantage of this scheme is represented by the interpolation of the vector potential on the fixed grid, which introduces numerical noise. This noise is accumulated at each time step, and can lead to instabilities (see the magnetic loop advection test in Section 5).

4.2.3. Continuous grid displacement

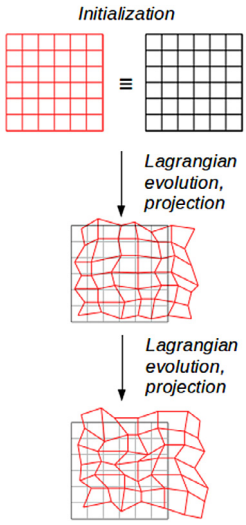
To reduce the number of interpolations (and noise accumulation) one can evolve the Lagrangian grid independently. This means that, after the first time step, $\mathbf{x}_{N'}^{n+1} \neq \mathbf{x}_N^{n+1}$. Retrieving the value of the vector potential on the fixed grid introduces some diffusion in the system, but this time there is no accumulation, as the reassignment $A_{z,N'}^{n+1} = A_{z,N}^{n+1}$ is skipped. Thus, at each cycle, the error on the interpolated values will not increase, certainly giving better results. However, the evolution of the Lagrangian grid according to the equation of motion,

$$\frac{d\mathbf{x}_{N'}}{dt} = \mathbf{u}_{N'}, \tag{35}$$

implies that the value of the velocity components on the Lagrangian grid, $\mathbf{u}_{N'}$, must be interpolated from the fixed one (as



(a) Regeneration at each time step.



(b) Subsequent Lagrangian evolutions.

Fig. 4. Representation of the different approaches for the evolution of the vector potential. The Lagrangian reference frame is either regenerated at each cycle or let evolve throughout the computation.

opposite to the previous strategy). This is still a one-way interpolation to be performed at each cycle, and would not cause numerical diffusion accumulation. The procedure is represented in Fig. 4(b).

The introduction of the secondary Lagrangian grid complicates the procedure to some extent, mainly because the secondary grid is not rectilinear in the non-moving frame. Firstly, severe distortion of the Lagrangian grid can create concave cells in which the grid–grid interpolation can fail, or produce the crossing between cell borders. Secondly, cells may concentrate in the specific regions of the computational domain, and at a certain time the solution may become meaningless due to interpolation issues. In [34], these issues are addressed with complex procedures for cell regeneration.

It has to be clarified, however, that the distortion of the grid depends on the specific simulation parameters and thus it is possible that none of the issues listed above ever appears.

4.2.4. Mixed approach

An obvious solution is to find a compromise between the two described algorithms: keep track of a secondary Lagrangian grid, but regenerate it when it becomes too distorted. The process is represented in Fig. 5.

The strategies described above are applied to a number of tests in Section 5.2. A comparison is made between the proposed VP and CT schemes.

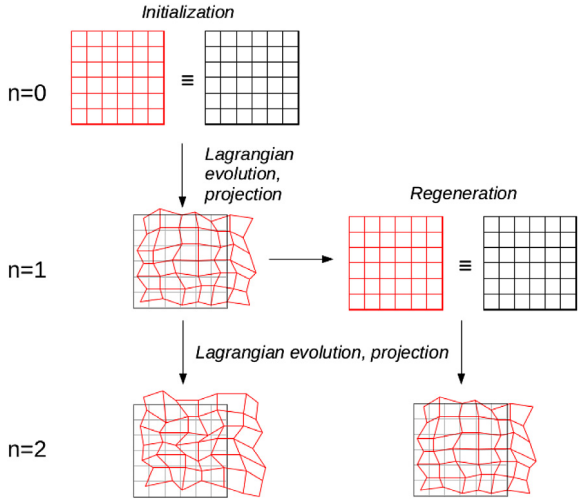


Fig. 5. "Mixed" approach for the evolution of the vector potential. At each cycle, the Lagrangian grid is either regenerated, according to some criterion (e.g., if it is too distorted) or evolved further.

5. Code validation tests

In this section, we present a number of standard HD and MHD tests showing the code performance. The HD tests are meant to show the effect of the volume evolution strategy, while the MHD tests will compare the different approaches for the magnetic field evolution.

5.1. Hydrodynamic tests

5.1.1. Sound wave

The propagation of a sound wave is modeled in 1D by imposing symmetry along the y-direction (that is, $\partial/\partial y = 0$). The importance of this test lies in having an analytical solution to be compared to the simulation results. In this case, the correctness of the algorithm is represented by the perturbation wave traveling at the correct sound speed as predicted from theory.

A background state is set on the grid variables by imposing the unperturbed values

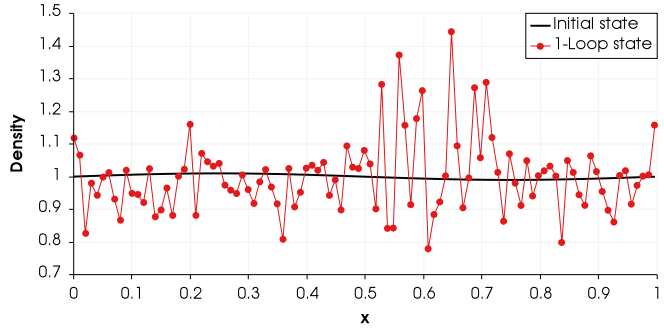
$$U_0 = (\rho_0, e_0, u_{x,0}, p_0)^T = (1, 1, 0, \rho_0 e_0(\gamma - 1))^T,$$

where the specific heat ratio is $\gamma = 5/3$. From linear theory, a perturbation is added to each quantity in the form

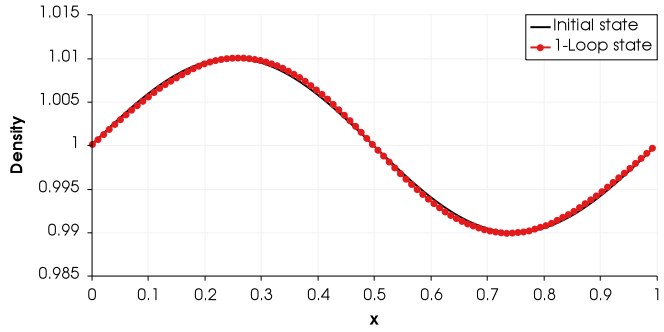
$$U_1 = (\rho_1, e_1, u_{x,1}, p_1)^T = (1, p_0, c, c^2)^T K \sin(2\pi x/L_x),$$

where $c = \sqrt{\gamma(\gamma - 1)e_0}$ is the sound speed of the unperturbed state, and K is the perturbation amplitude. The specific heat ratio is set to $\gamma = 5/3$ and we choose $K = 0.01$.

The simulation is run on a domain of length $L_x = 1$, with periodic boundary conditions along the x-direction, up to $t = 1.0$. The propagation of the sound wave is monitored by checking the speed of the perturbation as it moves along the domain, and comparing it with the exact value of the background sound speed c. Note that, in this case and for all hydrodynamic tests, normalized



(a) Solution without volume evolution.



(b) Solution with volume evolution.

Fig. 6. Comparison between the results of a sound wave propagation after one loop across the domain, with and without the particle volume evolution strategy.

timescales are related to the sound propagation time, L/c, with L the typical length scale of the system. The meaningless (because of the absence of magnetic field) Alfvén velocity is dropped in favor of the sound speed in the normalization of the macroscopic equations.

To obtain a reference result, a first run is set up with and without volume evolution procedure, at the low resolution of 128 cells and 9 particles per cell. Given the initial state, the sound speed of the unperturbed medium is given by $c_s = \sqrt{\gamma p_0/\rho_0} = 1.054$. Taking this value in computing the CFL condition, for the chosen resolution we have $\Delta t < 0.0082$. Here, we set the time step to the conservative value 0.0001. Note that "pure" particle methods, relying solely on particles for modeling fluid flows, are not constrained by CFL conditions. In our case, the presence of a grid imposes a limitation on the time step provided that the time differentiation of the macroscopic equation is explicit. Running the simulation up to the final time provides the solution shown in Fig. 6. In the simulation without volume evolution, the plot clearly shows the arising of the so-called ringing instability, which gives a meaningless solution.

We proceed then to check the convergence of the algorithm by keeping the volume evolution and varying the resolution parameters Δt, Δx, and N_{ppC} two at a time. The results are reported in Table 1. Here the relative error is computed by taking the relative absolute difference between the computed sound speed and the analytical value above. The computed value is retrieved by measuring the time needed by the wave to loop around the domain.

The results show that without volume evolution, the scheme fails in describing correctly the evolution of the system. The ringing instability arises in the density values, creating unphysical oscillations that make the solution meaningless. When the volume evolution is activated, space–time convergence is achieved when varying the number of cells and the time step. The anti-aliasing effect of the particle volume evolution procedure is rather surprising, and deserves further investigation. However, a formal description of the instability and the mechanism behind its elimination is beyond the scope of this work, and will be addressed in future analyses.

It is also remarkable that the particle resolution, after increasing over a certain threshold, does not affect significantly the precision of the algorithm. This is because the presence of particles adds several degrees of freedom to the representation of the system. When interpolating to the grid, however, such additional information is partly lost. In this specific case, it is shown that above 16 particles per cell there is no sensible difference in the results, meaning that a further increase in the particle resolution cannot bring in any refinement to the algorithm accuracy. It is also straightforward, however, that an insufficient particle resolution does not represent the system adequately, and often results in a break-up of the run.

5.1.2. Kelvin–Helmholtz instability

The Kelvin–Helmholtz instability [39] is a standard test for multi-dimensional HD codes. Here, we refer to the same setup studied by McNally et al. [40] and compare our results to those presented in the authors' study. Being the first phase of the system evolution linear, we have an analytical solution to be used as a reference to check the correctness of the results; at later times, the solution provided by other tested codes running the same setup can be used for comparison.

The system is initialized according to

$$\rho = \begin{cases} \rho_1 - \rho_m e^{\frac{y-1/4}{L}} & \text{if } 0 \le y < 1/4 \\ \rho_2 + \rho_m e^{\frac{-y+1/4}{L}} & \text{if } 1/4 \le y < 1/2 \\ \rho_2 + \rho_m e^{\frac{y-3/4}{L}} & \text{if } 1/2 \le y < 3/4 \\ \rho_1 - \rho_m e^{\frac{-y+3/4}{L}} & \text{if } 3/4 \le y < 1, \end{cases}$$

where $\rho_m = (\rho_1 - \rho_2)/2$, $\rho_1 = 1.0$, $\rho_2 = 2.0$ and $L = 0.025$. The x-component of the fluid velocity is given by

$$u = \begin{cases} u_{x,1} - u_m e^{\frac{y-1/4}{L}} & \text{if } 0 \le y < 1/4 \\ u_{x,2} + u_m e^{\frac{-y+1/4}{L}} & \text{if } 1/4 \le y < 1/2 \\ u_{x,2} + u_m e^{\frac{y-3/4}{L}} & \text{if } 1/2 \le y < 3/4 \\ u_{x,1} - u_m e^{\frac{-y+3/4}{L}} & \text{if } 3/4 \le y < 1, \end{cases}$$

where again $u_m = (u_{x,1} - u_{x,2})/2$, with $u_{x,1} = 0.5$ and $u_{x,2} = -0.5$. The initial perturbation is given by the y-component of velocity being set to $u_y = 0.01 \sin(4\pi x)$. Finally, the pressure is set initially at the uniform value 2.5. The specific heat ratio is set to $\gamma = 5/3$ and the test is run up to $t = 1.5$ with double periodic boundary conditions. The unit square domain is discretized in 256^2 cells with $N_{ppC} = 9$, and the time step is set to $\Delta t = 10^{-4}$.

As for the previous test, the simulation is run both with and without volume evolution. The results are compared in Fig. 7 with the reference solution [40].

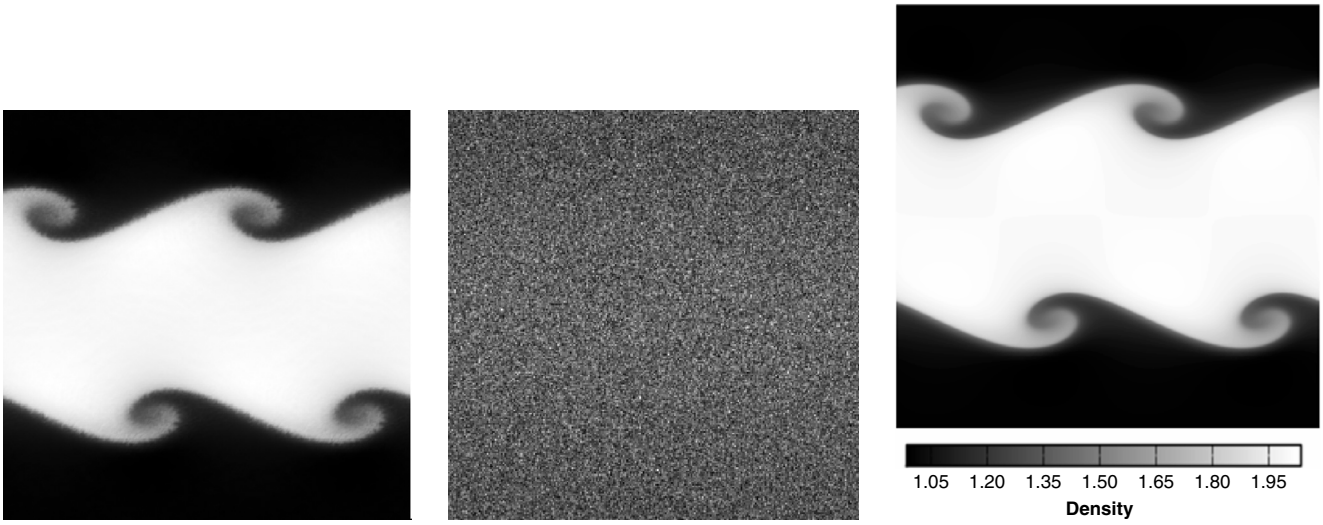
Once again, it is evident that the solution computed without the volume evolution is completely meaningless. In this case, it becomes mandatory to suppress the ringing instability, thus the strategy provides the necessary solution to the issue.

To better describe the evolution of the perturbation, we perform a quantitative analysis on the system parameters. The amplitude, M, and the relative growth rate, $\mathrm{d}M/\mathrm{d}t$, of the y-velocity mode of the instability are monitored throughout the computation, along with the maximum y-direction energy density $1/2\rho u_y^2$. The amplitude can be extracted at each time step by computing

$$M = 2\sqrt{\left(\frac{\sum_i^{N_y}\sum_j^{N_x} s_{i,j}}{\sum_i^{N_y}\sum_j^{N_x} d_{i,j}}\right)^2 + \left(\frac{\sum_i^{N_y}\sum_j^{N_x} c_{i,j}}{\sum_i^{N_y}\sum_j^{N_x} d_{i,j}}\right)^2}, \tag{36}$$

Table 1
Numerical convergence of the scheme for the propagation of a sound wave. Two parameters among grid resolution, particle resolution and time step are kept fixed in each test. From left to right: test with increasing resolution, increasing number of particles and decreasing time step.

$\Delta t = 1.e{-}3, N_{ppC} = 9$		$N_C = 128^2, \Delta t = 1.e{-}3$		$N_C = 128^2, N_{ppC} = 9$	
N_C	*Error*	N_{ppC}	*Error*	Δt	*Error*
32	2.399e−2	9	7.65e−3	1.e−3	7.65e−3
64	1.487e−2	16	6.61e−3	1.e−4	5.57e−3
128	7.65e−3	25	6.61e−3	1.e−5	3.48e−3
256	5.57e−3	36	6.61e−3	1.e−6	1.39e−3



(a) Solution with volume evolution. (b) Solution without volume evolution. (c) Solution from [40].

Fig. 7. Comparison between the results of the Kelvin–Helmholtz instability, with (a) and without (b) the particle volume evolution strategy, at time $t = 1.5$. The density is shown in the three pictures according to the same scale of the reference solution (c).

where

$$s_{i,j} = \begin{cases} u_y \sin(4\pi x) \exp(-4\pi |y - 1/4|) & \text{if } y < 1/2 \\ u_y \sin(4\pi x) \exp(-4\pi | - y + 3/4|) & \text{if } y \geq 1/2, \end{cases} \quad (37)$$

$$c_{i,j} = \begin{cases} u_y \cos(4\pi x) \exp(-4\pi |y - 1/4|) & \text{if } y < 1/2 \\ u_y \cos(4\pi x) \exp(-4\pi | - y + 3/4|) & \text{if } y \geq 1/2, \end{cases} \quad (38)$$

$$d_{i,j} = \begin{cases} \exp(-4\pi |y - 1/4|) & \text{if } y < 1/2 \\ \exp(-4\pi | - y + 3/4|) & \text{if } y \geq 1/2. \end{cases} \quad (39)$$

The results obtained from the simulation with volume evolution are presented in Fig. 8. The visual comparison with the reference data shows very good agreement throughout the computation, both during the linear and the nonlinear growth phases.

5.2. Magnetohydrodynamic tests

5.2.1. Magnetic loop advection

This test involves the solution of the whole MHD set of equations to transport a field loop across the domain. A non-dynamic version of the test (solving only the induction equation) has also been experimented [41]. Here, we set a flow velocity of $\sqrt{2}$ at an angle of $\theta = 45°$ with respect to the x-direction, so that $u_x = u_y = 1$. The domain is a unit square with double periodic boundary conditions. The pressure and density are both set to 1, with $\gamma = 5/3$.

The magnetic loop is placed, initially, at the center of the domain by computing a vector potential as

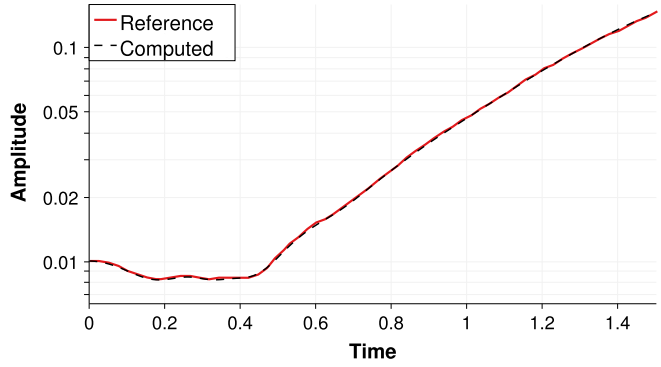
$$A_z = \text{MAX}(a(R_0 - r), 0),$$

where $a = 0.001$ to ensure that the resulting magnetic field is small enough not to cause oscillations in the hydrodynamical properties; $R_0 = 0.15$; $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the radius computed around the point (x_0, y_0), the center of the domain.

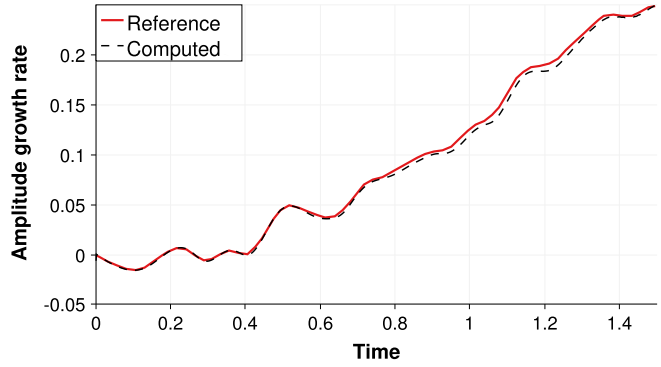
The goal of the test is ensuring that the initial loop be advected with the smallest distortion in the field lines, along with keeping $\nabla \cdot \mathbf{B} = 0$ as much as possible. The simulation is run up to $t = 5$ with a resolution of 128^2 cells and $\Delta t = 0.0001$. We compare the results for the various magnetic field evolution strategies in Fig. 9. Here, the VP scheme is set without regeneration, being the velocity uniform and constant throughout the computation, hence causing no grid distortion.

The results show that FLIP-MHD's method can handle the magnetic loop advection almost perfectly, only showing a tiny deformation of the initial magnetic field lines. The CT method, on the other hand, causes the arising of unphysical oscillations that completely deform the initial loop. As expected, a key aspect of the advection is missing, due to the mismatch between the particle motion computed in a Lagrangian frame and the magnetic field evolution kept on a fixed Eulerian frame. The VP method shows the best result, keeping the loop completely unchanged from the initial condition. This is due to the formulation of the algorithm, which truly respects the condition $dA_z/dt = 0$ while retaining the capability of modeling the motion of the magnetic field. Furthermore, the need for a Poisson solver is eliminated.

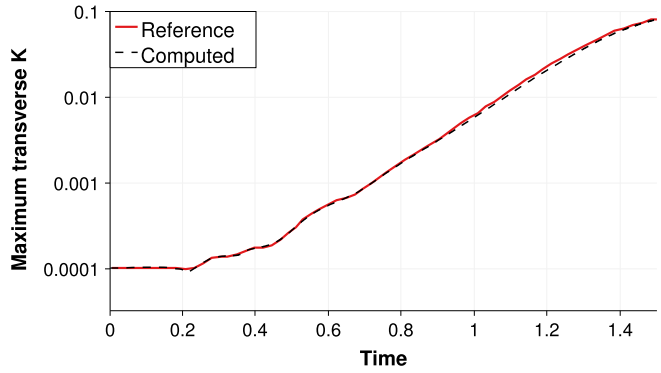
As an additional comparison, the time evolution of $\|\nabla \cdot \mathbf{B}\|$ for the three strategies is plotted in Fig. 10. Here, it is evident how the CT and VP strategies avoid the numerical build-up of divergence error, with respect to the FLIP-MHD's scheme. The VP scheme is confirmed to be the best choice as it introduces the lowest numerical error while keeping the simulation physically consistent.



(a) Time evolution of the instability amplitude.



(b) Time evolution of the instability amplitude growth rate.



(c) Time evolution of the maximum transverse kinetic energy.

Fig. 8. Evolution of the Kelvin–Helmholtz instability parameters.

5.2.2. Orszag–Tang vortex

The Orszag–Tang vortex problem [42] has been widely studied with a number of MHD codes. The nature of the problem is prevalently hydrodynamical, with the magnetic field introducing additional forces that distort the solution. Here, we take the results from the AMRVAC code [43,44], as a reference, to compare the different approaches for the magnetic field evolution.

The simulation is run on a unit square domain with double periodic boundary conditions. The initial properties of the grid are set according to

$$\begin{pmatrix} \rho \\ p \\ u_x \\ u_y \\ u_z \\ B_x \\ B_y \\ B_z \end{pmatrix} = \begin{pmatrix} 25/(36\pi) \\ 5/(12\pi) \\ -\sin(2\pi y) \\ \sin(2\pi x) \\ 0 \\ -B_0 \sin(2\pi y) \\ B_0 \sin(4\pi x) \\ 0 \end{pmatrix},$$

where $B_0 = (4\pi)^{-1/2}$. As an initial test, the system is let evolve up to $t = 1$ at the low resolution of 64^2 cells and time step 0.001. The three strategies for magnetic field evolution are compared in Fig. 11 by evaluating the time history of the mean magnetic energy. By default, the volume evolution strategy is kept activated throughout the simulation.

The plot clearly shows the differences between the strategies for magnetic field evolution. The reference solution (green line) is obtained with the AMRVAC code at a resolution of 256^2 cells. When the FLIP strategy is applied along with the volume evolution (dashed black line), the simulation breaks after diverging from the reference solution at some early time; if the volume evolution is switched off (solid black line), the run can be completed but the effect of the ringing instability violently kicks in causing an extreme dissipation in the magnetic energy, thus producing a meaningless result (see Fig. 7(a)).

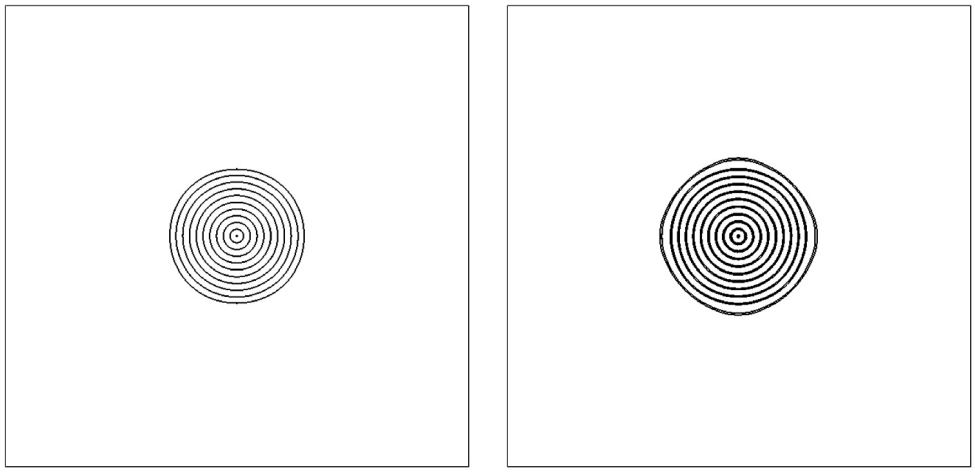
The CT strategy (blue line) follows the reference solution quite well initially. However, the result starts diverging at an early time, due to the missing modeling of the magnetic field advection; in the end, the run can be completed but there is discrepancy with the reference line. However, the mostly hydrodynamical nature of this test prevents the computed solution to show extreme differences with respect to the reference one, as occurring instead in the case of the previous test.

When applying the VP strategy with no regeneration (dashed red line), the result follows the same line produced by the CT scheme; however, the simulation breaks at about half the final time. This is due to the grid distortion, which in this case provokes the failing of the interpolation between the moving and fixed meshes. When setting a regeneration of the grid every five time steps (solid red line), the simulation can be completed, but the strategy introduces dramatic magnetic energy dissipation, which makes the results diverge significantly from the reference solution. However, the scheme allows the volume evolution strategy to be activated and correctly suppresses the ringing instability, thus still producing a better result with respect to FLIP's standard procedure. Moreover, the dissipation is mostly due to the one-to-one interpolation included in the remapping step. At low resolutions this certainly causes a much higher error, which can be expected to decrease if the number of cells increases.

Having verified the need for an alternative to FLIP's scheme, we proceed with further tests on the CT and VP strategies. Fig. 12 shows the quantitative comparison between the reference magnetic energy evolution and the results from the simulation with the two strategies, as the resolution increases. Here, the progressive refinement of the grid shows that the VP scheme converges to the reference line. The application of the CT scheme, however, does not significantly improve the quality of the results as the resolution increases. Finally, in Fig. 13 we plot the 1D spatial distribution of the pressure at $t = 0.5$ along $y = 0.4277$ and $y = 0.3125$, with progressive grid refinement for the VP scheme. This can be compared with the work of Londrillo and Del Zanna [45] (Figure 11) and the results from the ATHENA code, which uses a third order reconstruction algorithm [46,47]. The plot shows once again the convergence of the results towards the reference solution obtained with AMRVAC.

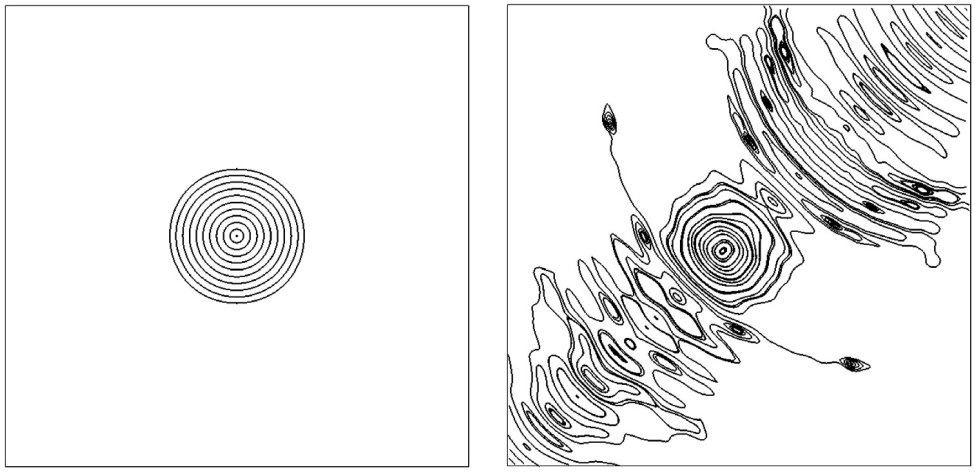
6. Discussion and conclusions

We have presented two improvements of the existing MHD fluid particle-in-cell original schemes (of which FLIP [13] is the most widely used one): the volume evolution procedure and a vector potential-based magnetic field evolution strategy. We have tested our developments on a number of classical problems using a 2D Fortran prototype of the new code.



(a) Initial condition.

(b) Solution with the FLIP-MHD scheme.



(c) Solution with VP scheme.

(d) Solution with the CT scheme.

Fig. 9. Results of the advection test after 5 loops across the domain. The figures show the magnetic field lines of the advected loop.

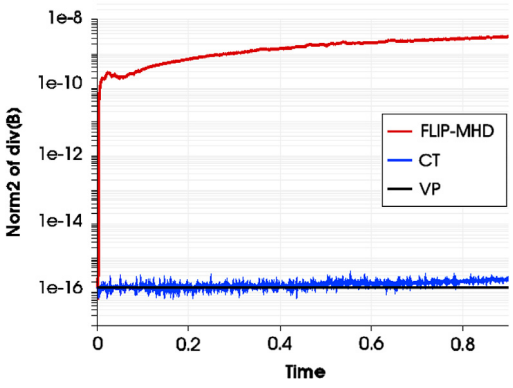


Fig. 10. Time evolution of the magnetic field divergence norm for the three strategies.

- The proposed volume evolution procedure improves the accuracy of the scheme and stabilizes the solution by effectively suppressing the ringing instability even if explicit time integration is used. The new procedure introduces little to no numerical diffusion and computational expense. The effectiveness of the procedure is confirmed even during nonlinear phases of the system evolution such as in the Kelvin–Helmholtz instability test.

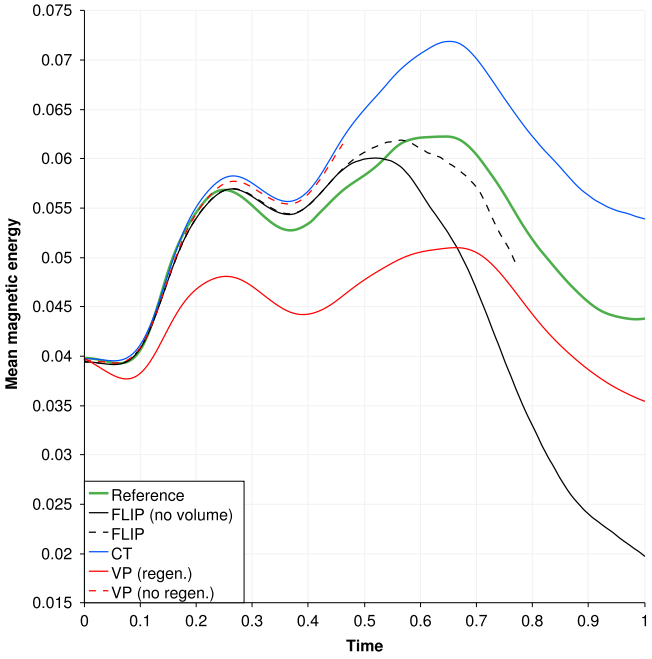


Fig. 11. Time history of the mean magnetic energy for the Orszag–Tang test.

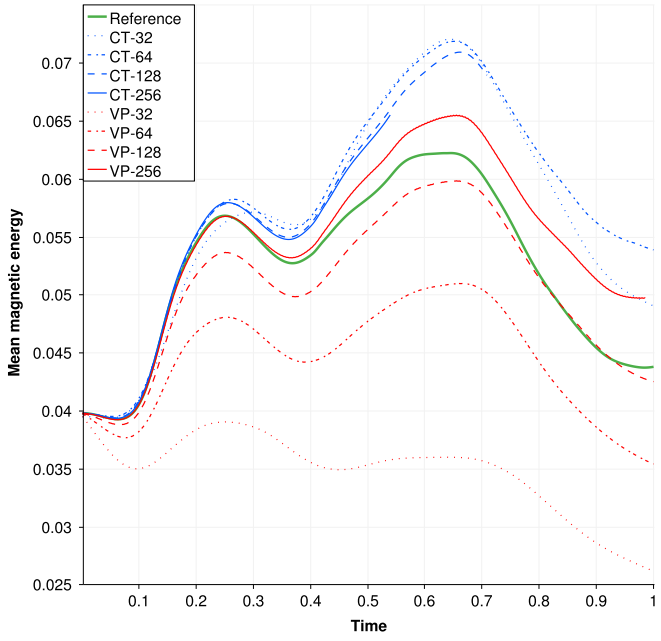
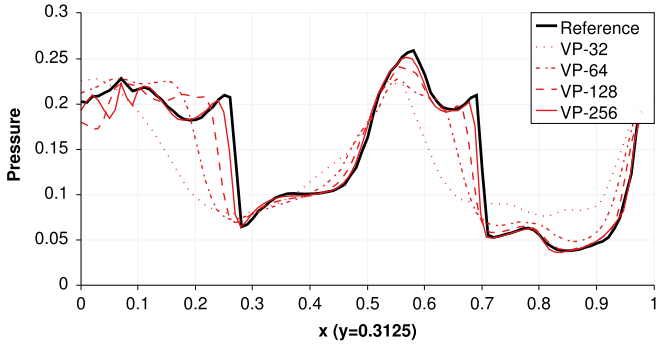
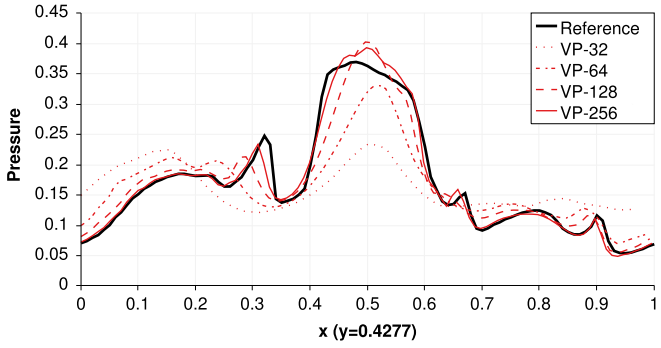


Fig. 12. Convergence test for the VP and CT strategies applied to the Orszag–Tang test. The magnetic energy evolution is plotted up to the final time.



(a) Cut along $y = 0.3125$.



(b) Cut along $y = 0.4277$.

Fig. 13. Convergence test for the VP strategy applied to the Orszag–Tang test. The pressure at $t = 0.5$ is plotted along horizontal cuts of the domain.

- New magnetic field evolution strategies eliminate the need in using the Poisson solver (required by FLIP), while preserving $\nabla \cdot \mathbf{B} = 0$ to machine precision. The constrained transport scheme is rather simple and requires negligible computational effort, compared to the solution of a Poisson equation. However, the advective component of the magnetic field evolution equation is left out of the computation. This can affect the results significantly in specific cases (e.g., when the system dynamics is dominated by magnetic diffusion phenomena). However, if the hydrodynamic effects are predominant, the results are rather satisfactory.
- The second method is based on advancing the electromagnetic vector potential on a dedicated Lagrangian grid, and its interpolation back onto fixed Eulerian grid. When the moving grid becomes too distorted, it is rezoned to its original position (fixed Eulerian grid), and vector potential is remapped accordingly. This step obviously introduces some diffusion; however, the amount of dissipated magnetic energy decreases with increasing grid resolution, converging to the expected result.

The proposed magnetic field solver based on a moving grid deviates somewhat from FLIP's particle-linked advection strategy. Considering the main idea behind FLIP's algorithm, each cell of the domain is basically divided in a number of sub-elements when interpolating the conserved quantities from the grid to the particles. These are used, at each time step, to model the transport of the macroscopic variables without the need to address the complexity of computing gradients on a moving grid. The grid is therefore kept fixed and particles wander through the domain as the system evolves. The nodes of the distorted grid, in our strategy, can be regarded as a set of particles: each one corresponds to a node of the fixed grid, moving around the values of the vector potential.

Clearly, the strength of the procedure relies on simple expression of the Lagrangian evolution of the vector potential. It is especially handy in 2D, where the value of A_z does not even change on the Lagrangian grid. Finally, the strategy turns out to be compatible with the volume evolution procedure without issues even at high resolutions.

While there is certainly room for further improvements, the proposed approach is a major step towards the new generation of particle-based methods that incorporate excellent parallelization capabilities together with simple post-processing and low storage requirements.

Acknowledgments

F.B. thanks B. Ripperda for providing the reference data of OT test; B. Ripperda, D. Millas, and N. Magyar for useful discussions throughout the development of this work. F.B. and V.O. are supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF under Award No. FA9550-14-1-0375. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) via the DEEP-ER project under Grant Agreement no. 610476. This work is supported in part by the Belgian Space Policy IUAP (IAP P7/08 network CHARM).

Appendix A. Discretization of the governing equations and review of the computational cycle

We present here the chosen discretization of the 2D fluid equations, (1)–(4). As stated previously, the spatial derivatives of cell center quantities (e.g. the specific internal energy) drive the evolution of node quantities (e.g. the velocity) and vice versa. It follows that the spatial derivative of a cell quantity is located on the nodes, and vice versa. For convenience, we introduce two operators expressing the discrete directional derivatives, so that

$$\frac{\partial}{\partial x} = \Delta^x, \qquad \frac{\partial}{\partial y} = \Delta^y.$$

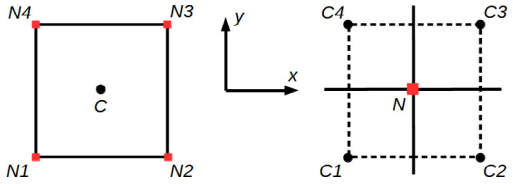


Fig. A.14. Reference scheme for the spatial discretization of the macroscopic quantities. The spatial derivatives are computed on a cell center (left) or a node (right) using the four surrounding elements.

With reference to Fig. A.14, applying these operators to either a node quantity U_N or a cell quantity U_C produces the following finite difference approximations:

$$\Delta_C^x U_N = \frac{U_{N2} - U_{N1} + U_{N3} - U_{N4}}{2\Delta x},$$
$$\Delta_C^y U_N = \frac{U_{N4} - U_{N1} + U_{N3} - U_{N2}}{2\Delta y},$$
$$\Delta_N^x U_C = \frac{U_{C2} - U_{C1} + U_{C3} - U_{C4}}{2\Delta x},$$
$$\Delta_N^y U_C = \frac{U_{C4} - U_{C1} + U_{C3} - U_{C2}}{2\Delta y}.$$

With the definitions above, the discretized momentum equation, in component form, reads

$$\rho_N^n \frac{u_{x,N}^{n+1} - u_{x,N}^n}{\Delta t} = -\Delta_N^x \left(p + \frac{B_y^2 + B_z^2}{2} \right)_C^n + B_{y,N}^n \Delta_N^y B_{x,C}^n, \tag{A.1}$$

$$\rho_N^n \frac{u_{y,N}^{n+1} - u_{y,N}^n}{\Delta t} = -\Delta_N^y \left(p + \frac{B_x^2 + B_z^2}{2} \right)_C^n + B_{x,N}^n \Delta_N^x B_{y,C}^n, \tag{A.2}$$

$$\rho_N^n \frac{u_{z,N}^{n+1} - u_{z,N}^n}{\Delta t} = B_{x,N}^n \Delta_N^x B_{z,C}^n + B_{y,N}^n \Delta_N^y B_{z,C}^n. \tag{A.3}$$

The discrete continuity and internal energy equations read

$$\frac{\rho_C^{n+1} - \rho_C^n}{\Delta t} = -\rho_C^n (\Delta_C^x u_{x,N}^{n+1} + \Delta_C^y u_{y_N}^{n+1}), \tag{A.4}$$

$$\rho_C^n \frac{e_C^{n+1} - e_C^n}{\Delta t} = -p_C^n (\Delta_C^x u_{x,N}^{n+1} + \Delta_C^y u_{y_N}^{n+1}). \tag{A.5}$$

Finally, the discrete form of the induction equation is written, in component form, as

$$\rho_C^{n+1} \frac{B_{x,C}^{n+1} - B_{x,C}^n}{\Delta t} = B_{x,C}^n \Delta_C^x u_{x,N}^{n+1} + B_{y,C}^n \Delta_C^y u_{x,N}^{n+1}, \tag{A.6}$$

$$\rho_C^{n+1} \frac{B_{y,C}^{n+1} - B_{y,C}^n}{\Delta t} = B_{x,C}^n \Delta_C^x u_{y,N}^{n+1} + B_{y,C}^n \Delta_C^y u_{y,N}^{n+1}, \tag{A.7}$$

$$\rho_C^{n+1} \frac{B_{z,C}^{n+1} - B_{z,C}^n}{\Delta t} = B_{x,C}^n \Delta_C^x u_{z,N}^{n+1} + B_{y,C}^n \Delta_C^y u_{z,N}^{n+1}. \tag{A.8}$$

In the above equations, quantities needed at points in the domain on which they are not naturally discretized (such as the magnetic field in the momentum equation) can be obtained by simple averaging. Note that the updated velocity is used to advance the cell center quantities.

The particle equations of motion are explicitly discretized in time as

$$\frac{\mathbf{u}_p^{n+1} - \mathbf{u}_p^n}{\Delta t} = \sum_N \frac{\mathbf{u}_N^{n+1} - \mathbf{u}_N^n}{\Delta t} W_{pN}, \tag{A.9}$$

$$\frac{\epsilon_p^{n+1} - \epsilon_p^n}{\Delta t} = m_p \sum_C \frac{e_C^{n+1} - e_C^n}{\Delta t} W_{pC}, \tag{A.10}$$

$$\frac{\mathbf{x}_p^{n+1} - \mathbf{x}_p^n}{\Delta t} = \mathbf{u}_p^n. \tag{A.11}$$

Finally, the discrete versions of the coupling equations relating particles and grid read

$$m_C^n = \sum_p m_p W_{Cp}, \tag{A.12}$$

$$(m\mathbf{u})_N^n = \sum_p m_p \mathbf{u}_p^n W_{Np}, \tag{A.13}$$

$$(me)_C^n = \sum_p \epsilon_p^n W_{Cp}. \tag{A.14}$$

In summary, the computational cycle is as follows:

- The particle properties are projected on the grid to initialize density, specific internal energy, and momentum.
- The magnetic field is initialized according to the preferred strategy. In the VP approach, the vector potential is interpolated on the regular grid from the evolved mesh.
- The grid is updated according to the discrete equations of motion above.
- The changes on the conserved quantities are projected on the particles by inverse interpolation.
- The particles are moved through the domain and boundary conditions are applied.

Appendix B. Derivation of the discretized $\nabla \cdot \mathbf{B} = 0$ condition for the VP strategy

With the definitions given in Appendix A, the derivation of the discrete divergence-free condition is straightforward. Computing $\nabla \cdot \mathbf{B}$ on a node requires knowing the values of the magnetic field on the four surrounding cell centers. This operation involves the values of A_z on the four nodes surrounding each cell center. In total, the vector potential must be known on nine nodes to compute $\nabla \cdot \mathbf{B}$ on the central one. This appears evident when considering the discrete equations

$$B_{x,C} = \Delta_C^y A_{z,N}, \qquad B_{y,C} = -\Delta_C^x A_{z,N},$$

and the discrete divergence-free condition

$$(\nabla \cdot \mathbf{B}_C)_N = \Delta_N^x B_{x,C} + \Delta_N^y B_{y,C}.$$

Taking as a reference the scheme shown in Fig. B.15, the magnetic field on the four cell centers is given by

$$B_{x,C1} = \frac{A_{z,N4} - A_{z,N1} + A_{z,N5} - A_{z,N2}}{2\Delta y},$$
$$B_{y,C1} = \frac{A_{z,N4} - A_{z,N5} + A_{z,N1} - A_{z,N2}}{2\Delta x},$$
$$B_{x,C2} = \frac{A_{z,N5} - A_{z,N2} + A_{z,N6} - A_{z,N3}}{2\Delta y},$$
$$B_{y,C2} = \frac{A_{z,N5} - A_{z,N6} + A_{z,N2} - A_{z,N3}}{2\Delta x},$$
$$B_{x,C3} = \frac{A_{z,N7} - A_{z,N4} + A_{z,N8} - A_{z,N5}}{2\Delta y},$$
$$B_{y,C3} = \frac{A_{z,N7} - A_{z,N8} + A_{z,N4} - A_{z,N5}}{2\Delta x},$$
$$B_{x,C4} = \frac{A_{z,N8} - A_{z,N5} + A_{z,N9} - A_{z,N6}}{2\Delta y},$$
$$B_{y,C4} = \frac{A_{z,N8} - A_{z,N9} + A_{z,N5} - A_{z,N6}}{2\Delta x}.$$

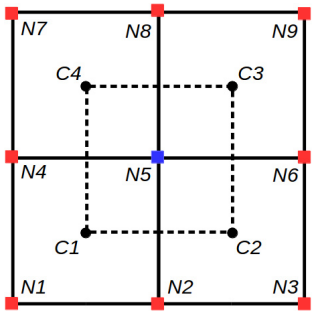


Fig. B.15. Reference scheme for the computation of the discrete $\nabla \cdot \mathbf{B}$. Computing the value on the central node $N5$ involves the values of the vector potential on the nine nodes represented in the picture.

The resulting discrete divergence, computed on the central node, is thus

$$(\nabla \cdot \mathbf{B}_C)_{N5} = \frac{B_{x,C4} - B_{x,C3} + B_{x,C2} - B_{x,C1}}{2\Delta x} + \frac{B_{y,C3} - B_{y,C1} + B_{y,C4} - B_{y,C2}}{2\Delta y}.$$

By substituting the expressions of B_x and B_y above, it follows immediately that $(\nabla \cdot \mathbf{B}_C)_{N5} = 0$ exactly. This applies to any node in the domain, ensuring that the divergence-free condition is respected at all time steps.

References

[1] R. Teyssier, Ann. Rev. Astronom. Astrophys. 53 (2015) 325–364.
[2] L.B. Lucy, Astrophys. J. 82 (1977) 1013–1024.
[3] V. Springel, Ann. Rev. Astronom Astrophy. 48 (2010) 391–430.
[4] J.J. Monaghan, Annu. Rev. Fluid Mech. 44 (2012) 323–346.
[5] C.W. Hirt, A.A. Amsden, J.L. Cook, J. Comput. Phys. 14 (1974) 227–253.
[6] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodriguez-Ferran, Arbitrary Lagrangian–Eulerian Methods, 2004.
[7] L. Margolin, Published Online on ResearchGate, Vo. 1, 2013, pp. 1–28.
[8] G.G. Gray, J.K. Morgan, P.F. Sanz, J. Struct. Geol. 59 (2014) 19–36.
[9] F. Harlow, Methods Comput. Phys. 3 (1964) 319–343.
[10] A.B. Langdon, C.K. Birdsall, Phys. Fluids 13 (1970) 2115–2122.
[11] G. Lapenta, J. Comput. Phys. 231 (2012) 795–821.
[12] J.U. Brackbill, H.M. Ruppel, J. Comput. Phys. 65 (1986) 314–343.
[13] J.U. Brackbill, J. Comput. Phys. 96 (1991) 163–192.
[14] D.B. Kothe, J.U. Brackbill, C.K. Choi, Phys. Fluids B 2 (8) (1990).
[15] J.U. Brackbill, Internat. J. Numer. Methods Fluids 47 (2005) 693–705.
[16] D. Sulsky, S.-J. Zhou, H.L. Schreyer, Comput. Phys. Comm. 87 (1995) 236–252.
[17] S. Bardenhagen, J. Brackbill, D. Sulsky, Comput. Methods Appl. Mech. Engrg. 187 (3–4) (2000) 529–541.
[18] S. Cummins, J. Brackbill, J. Comput. Phys. 180 (2) (2002) 506–548.
[19] D. Sulsky, H. Schreyer, K. Peterson, R. Kwok, M. Coon, J. Geophys. Res. (Oceans) 112 (2007) C02S90.
[20] F. Fatemizadeh, C. Moormann, IOP Conf. Ser.: Earth Environ. Sci. 26 (1) (2015) 012019.
[21] A.R. York, D. Sulsky, H.L. Schreyer, Internat. J. Numer. Methods Engrg. 48 (6) (2000) 901–924.
[22] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, A. Selle, ACM Trans. Graph. 32 (4) (2013) 102:1–102:10.
[23] G. Lapenta, J.U. Brackbill, J. Comput. Phys. 115 (1) (1994) 213–227.
[24] J. Birn, F. Iinoya, J.U. Brackbill, M. Hesse, Geophys. Res. Lett. 23 (4) (1996) 323–326.
[25] D. Sulsky, Z. Chen, H. Schreyer, Comput. Methods Appl. Mech. Engrg. 118 (1–2) (1994) 179–196.
[26] D. Burgess, D. Sulsky, J. Brackbill, J. Comput. Phys. 103 (1) (1992) 1–15.
[27] J. Brackbill, Comput. Phys. Comm. 47 (1) (1987) 1–16.
[28] J. Brackbill, D. Kothe, C. Zemach, J. Comput. Phys. 100 (2) (1992) 335–354.
[29] E. Love, D.L. Sulsky, Internat. J. Numer. Methods Engrg. 65 (10) (2006) 1608–1638.
[30] A. Sadeghirad, R.M. Brannon, J. Burghardt, Internat. J. Numer. Methods Engrg. 86 (12) (2011) 1435–1456.
[31] S. Bardenhagen, E. Kober, Comput. Model. Eng. Sci. 5 (6) (2004) 477–495.
[32] C. Evans, J. Hawley, Astrophys. J. 332 (2) (1988) 659–677.
[33] G. Tóth, J. Comput. Phys. 161 (2) (2000) 605–652.
[34] P. Mocz, M. Vogelsberger, L. Hernquist, Mon. Not. R. Astron. Soc. 442 (1) (2014) 43–55.
[35] L. Davidson, Internat. J. Numer. Methods Fluids 22 (4) (1996) 265–281.
[36] A. D. G.H. Jasak, Numer. Heat Transfer B 38 (3) (2000) 257–271.
[37] S. Muzaferija, D. Gosman, J. Comput. Phys. 138 (2) (1997) 766–787.
[38] P. Rambo, P.W. Rambo, Lawrence Livermore National Laboratory Report UCRL-ID-132123.
[39] S. Chandrasekhar, Hydrodynamic and Hydromagnetic Stability, 1961.
[40] C.P. McNally, W. Lyra, J.-C. Passy, Astrophys. J. Suppl. 201 (2012) 18.
[41] G. Tóth, D. Odstrčil, J. Comput. Phys. 128 (1) (1996) 82–100.
[42] S.A. Orszag, C.-M. Tang, J. Fluid Mech. 90 (1979) 129–143.
[43] R. Keppens, Z. Meliani, A. van Marle, P. Delmont, A. Vlasis, B. van der Holst, J. Comput. Phys. 231 (3) (2012) 718–744. Special Issue: Computational Plasma PhysicsSpecial Issue: Computational Plasma Physics.
[44] O. Porth, C. Xia, T. Hendrix, S.P. Moschou, R. Keppens, Astrophys. J. Suppl. 214 (2014) 4.
[45] P. Londrillo, L. Del Zanna, Astrophys. J. 530 (2000) 508–524.
[46] T.A. Gardiner, J.M. Stone, J. Comput. Phys. 227 (8) (2008) 4123–4141.
[47] J.M. Stone, T.A. Gardiner, P. Teuben, J.F. Hawley, J.B. Simon, Astrophys. J. Suppl. 178 (2008) 137–177.