Computer Physics Communications 291 (2023) 108838

Contents lists available at ScienceDirect

Computer Physics Communications

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

Computational Physics

A finite volume flux coordinate independent approach ☆

Matthias Wiesenberger [a,*], Markus Held [b,c]

[a] Department of Physics, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark
[b] Department for Mathematics and Statistics, UiT The Arctic University of Norway, Tromsø, N-9037, Norway
[c] Institute for Ion Physics and Applied Physics, Universität Innsbruck, Innsbruck, A-6020, Austria

ARTICLE INFO

Article history:
Received 1 December 2022
Received in revised form 14 June 2023
Accepted 20 June 2023
Available online 3 July 2023

Keywords:
FCI
Finite volume
Parallel advection
FELTOR

ABSTRACT

We present a novel family of schemes as the merging between a one-dimensional advection scheme with the flux coordinate independent approach. The scheme can be used to discretize the field-aligned Navier-Stokes equations in three dimensions. Our approach consists of three major steps: (i) the formulation of the one-dimensional scheme in a locally field-aligned coordinate system, (ii) a numerical evaluation of the surface integrals over the field-aligned finite volumes and (iii) the introduction of smoothing into the numerical transformation operators to ensure stability of the resulting scheme.

We study this approach at the example of a staggered finite volume scheme with a field-aligned cylinder as initial condition. We show superior stability and conservative properties over previous direct discretizations. In particular, the relative mass conservation is improved by several orders of magnitude. Without smoothing in the transformation operators the scheme is prone to oscillations in both parallel and perpendicular directions. In the presence of strong perpendicular gradients, additional parallel diffusion is needed to control spurious oscillations in the perpendicular planes.

We provide parallel implementations for various platforms including GPUs freely in the C++ library FELTOR

Contents

☆ The review of this paper was arranged by Prof. David W. Walker.

* Corresponding author.
E-mail address: mattwi@fysik.dtu.dk (M. Wiesenberger).

https://doi.org/10.1016/j.cpc.2023.108838

1. Introduction

The purpose of this paper is to find a discretization of the Navier-Stokes equation along the streamlines of a given unit vector field $\hat{\boldsymbol{b}}(\boldsymbol{x})$ in three dimensions:

$$\frac{\partial}{\partial t} n + \boldsymbol{\nabla} \cdot \left(n u \hat{\boldsymbol{b}} \right) = 0 \tag{1a}$$

$$\frac{\partial}{\partial t} (nu) + \boldsymbol{\nabla} \cdot \left(n u^2 \hat{\boldsymbol{b}} \right) = -\nabla_\parallel n + \nu_u \Delta_\parallel u \tag{1b}$$

with

$$\nabla_\parallel f := \hat{\boldsymbol{b}} \cdot \boldsymbol{\nabla} f \tag{2}$$

$$\Delta_\parallel u := \nabla \cdot (\hat{\boldsymbol{b}} \hat{\boldsymbol{b}} \cdot \nabla u) \tag{3}$$

where n is density, u is velocity and ν_u is the viscosity coefficient. Here, we use an isothermal setting with pressure $p = nT$ with constant temperature $T = 1$. We consider schemes for both the case with shocks ($\nu_u = 0$) as well as without ($\nu_u \neq 0$). The direction along $\hat{\boldsymbol{b}}$ is in this work called the "parallel" direction, while "perpendicular" refers to planes $\perp \hat{\boldsymbol{b}}$.

Our motivation to study such a system stems from three-dimensional plasma turbulence simulations, where $\hat{\boldsymbol{b}}$ is the magnetic field unit vector [1–6]. In a magnetized plasma, the dynamics parallel to $\hat{\boldsymbol{b}}$ fundamentally differs from the dynamics perpendicular to $\hat{\boldsymbol{b}}$. Typically, the length scale of density and velocity fluctuations perpendicular to $\hat{\boldsymbol{b}}$, $L_\perp$, is much smaller than the scale of fluctuations parallel to $\hat{\boldsymbol{b}}$, $L_\parallel$: $L_\perp \ll L_\parallel$. The resolution in a numerical simulation of such a system thus is fundamentally lower in the parallel direction than perpendicular to it. In order to develop schemes just for the parallel direction, we neglect perpendicular dynamics, resistivity and the electric potential, which leaves Eqs. (1) as a subset of the density and velocity equations.

The straightforward idea of discretizing Eqs. (1) is to construct a field-aligned coordinate system where one coordinate is given by the streamlines of $\hat{\boldsymbol{b}}$, i.e. solution to the equation $\dot{\boldsymbol{x}} = \hat{\boldsymbol{b}}$. This effectively reduces Eqs. (1) to a one-dimensional problem highlighting that the dynamics on different field lines is entirely decoupled. Do note the appearance of the volume element $\sqrt{g}$ in the divergences of Eqs. (1) in curvilinear coordinate systems, e.g. $\boldsymbol{\nabla} \cdot (nu\hat{\boldsymbol{b}}) = \partial_s \left(\sqrt{g} n u b^s \right) / \sqrt{g}$, where s is the field-line following coordinate. This approach was and is used in so-called flux-tube coordinates [7–9]. Aligning one coordinate to $\hat{\boldsymbol{b}}$ allows to use few grid points in this parallel coordinate while using a high resolution in the remaining two. The downside of a flux-tube is that (i) for strong magnetic shear the grid cells strongly distort and (ii) the field-lines do in general not close on themselves making boundary conditions hard to treat correctly [10,9]. However, the major issue is the inability to treat X-points or O-points (points with vanishing gradient) in the magnetic flux function. This is because field-lines cannot cross stationary points. In fact, it can be shown that even the less stringent flux-aligned coordinate systems are unable to treat X-points correctly [11]. The solutions to elliptic equations discretized on flux-aligned coordinates with X-point do not converge in general and only with first order if the flux function meets certain criteria at the X-point. A correct modelling of magnetic field equilibria that include one or even several X-points is however crucial for the development and study of current and future magnetic fusion devices [12–14].

Under the constraint to discretize Eq. (1) on a grid that is not aligned to and independent of $\hat{\boldsymbol{b}}$ the most straightforward idea would be to simply use a direct discretization on a standard Cartesian or Cylindrical grid. For example the term $\nabla_\parallel n = b^R \partial_R n + b^Z \partial_Z n + b^\varphi \partial_\varphi n$ can be simply discretized using standard finite difference methods. Such an approach is used for example in [15] with the immediate advantage to model arbitrary magnetic field equilibria. However, the main downside of such an approach is its inability to take advantage of the reduced numerical resolution requirement along the magnetic field. This comes with an overly stringent CFL condition and strong numerical diffusion due to the fast parallel dynamics. As [16,17] point out the result is a significantly higher computational cost for a non-aligned scheme (several orders of magnitude for larger machines in fact) than for an aligned scheme.

The motivation to combine the flexibility of non-aligned grids with the optimized computational cost of aligned discretizations leads to the development of the so-called flux-coordinate independent approach (FCI) [18,19,16]. The main idea of the FCI scheme is to construct a field-line map that maps each point in the numerical grid to its nearest toroidal plane along the streamlines of $\hat{\boldsymbol{b}}$. At the next toroidal plane, the resulting value can be interpolated from its surrounding grid points. In such a way, simple finite difference formulas for $\nabla_\parallel f$, $\boldsymbol{\nabla} \cdot (\hat{\boldsymbol{b}} f)$ and $\Delta_\parallel f$ can be constructed that do not require an aligned underlying grid and at the same time make use of the field-aligned character of plasma turbulence. This approach was successfully applied to fluid plasma turbulence [1–6].

Past work on the FCI scheme has mainly focused on the discretization of the parallel operators $\nabla_\parallel f$, $\nabla\cdot(\hat{\boldsymbol{b}}f)$ and $\Delta_\parallel f$ individually, without relation to the nature of Eqs. (1) as hyperbolic conservation laws. Unfortunately, such an approach fails to incorporate the success of specialized shock-capturing and conservative schemes that are available in the literature. In fact, methods for the discretization of the one-dimensional form of Eqs. (1) (taking straight field-lines $\hat{\boldsymbol{b}}\equiv e_x$) are plentiful and the foundations can be found in textbooks, for example [20]. Shocks, or discontinuities in the density and momentum density, may appear in the case $\nu_u=0$, or numerically, if steep gradients appear that the numerical scheme cannot distinguish from an actual shock. Here, Godunov type schemes combined with an (approximate) Riemann solver are notable. Others include staggered finite volume schemes [21–23] and discontinuous Galerkin schemes [24,25]. These schemes have excellent conservative properties and are generally more stable than simple finite difference discretizations.

The goal of this work is to bridge the gap between the FCI approach on the one side and the literature on one-dimensional conservation laws on the other side. We first provide a formalism that re-interprets the FCI approach as a discretization in a locally field-aligned coordinate system. We are then able to elevate one-dimensional approaches to the three-dimensional case via their formulation in the field-aligned coordinate system. A crucial ingredient in this approach is to allow smoothing within the transformation from non-aligned to aligned coordinates and back. Numerically, this is realized by interpolation and projection matrices on constant or linear finite elements. We provide parallel implementations of the resulting finite volume FCI approach (FV-FCI) for various platforms including GPUs freely in the C++ library Feltor [26,27]. We then show and compare various discretizations at the example of a field-aligned structure in a non-aligned grid. We remark that we do not include the temperature equation in Eqs. (1) to keep the derivation simple but that the extension is immediate. Finally, by replacing the pressure gradient $\nabla_\parallel n$ in Eq. (1b) with $g\nabla_\parallel n^2/2$ we have the shallow-water equations, which may be equally interesting.

The paper is organized as follows: in Section 2 we introduce and compare various finite difference and finite volume schemes on staggered grids. We discuss their behaviour for a shock problem and one with finite viscosity. In Section 3 we introduce the FCI approach and present our formulation in a locally field-aligned coordinate system. We present convergence tables for the discretization of $\nabla_\parallel f$, $\nabla\cdot(\hat{\boldsymbol{b}}f)$ and $\Delta_\parallel f$ with a particular focus on the conservative properties of the latter two. In Section 4 we then elevate the schemes introduced in Section 2 and show their behaviour with example simulations of a field-aligned structure in Section 5. Finally, we present conclusive remarks in Section 6. The topics of curvilinear coordinates and the complete Jacobian of the coordinate transformation are discussed separately in Appendix A and Appendix B. All figures and data presented in this paper are reproducible as outlined in Appendix C.

2. Finite volume schemes in one-dimension

2.1. Staggered grids in 1d

When $\hat{\boldsymbol{b}}=\hat{\boldsymbol{e}}_x$ the system (1) reduces to the one-dimensional Navier-Stokes equations for which there is a large amount of literature on numerical schemes available (see for example [20] and references therein).

$$\frac{\partial}{\partial t}n=-\frac{\partial}{\partial x}(nu) \tag{4}$$

$$\frac{\partial}{\partial t}nu=-\frac{\partial}{\partial x}\left(nu^2+n\right)+\nu_u\frac{\partial^2}{\partial x^2}u \tag{5}$$

We here choose a class of schemes known as staggered grid discretization. The main goal is to discretize the variables n and nu on two grids that are adjoint to each other. On a domain $[a,b]$ we choose $N+1$ grid points with grid distance $\Delta x=(b-a)/N$ and let $x_0:=a$, $x_k:=x_{k-1}+\Delta x$ for $k=1,\dots,N$. We denote the points of the adjoint grid as $x_{k+1/2}:=x_k+\Delta x/2$. If n is discretized on x_k, then nu is discretized on $x_{k+1/2}$. We will indicate this by writing $n_k:=n(x_k)$ and $(nu)_{k+1/2}:=(nu)(x_{k+1/2})$. We base our discretization on the finite volume scheme presented by [21–23]:

$$\frac{\mathrm{d}}{\mathrm{d}t}n_k=-\frac{1}{\Delta x}(\hat{q}_{k+1/2}-\hat{q}_{k-1/2}) \tag{6a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(nu)_{k+1/2}=-\frac{1}{\Delta x}\left(\hat{f}_{k+1}-\hat{f}_k\right)-\frac{1}{\Delta x}\left[n_{k+1}-n_k\right]+\frac{\nu_u}{(\Delta x)^2}\left[u_{k+3/2}-2u_{k+1/2}+u_{k-1/2}\right] \tag{6b}$$

where

$$\hat{q}_{k+1/2}:=u_{k+1/2}\begin{cases}n_k+\frac{1}{2}\Lambda(\Delta n_{k+1/2},\Delta n_{k-1/2}) & \text{if } u_{k+1/2}\geq 0\\ n_{k+1}-\frac{1}{2}\Lambda(\Delta n_{k+3/2},\Delta n_{k+1/2}) & \text{if } u_{k+1/2}<0\end{cases} \tag{7a}$$

$$\hat{f}_k:=q_k\begin{cases}u_{k-1/2}+\frac{1}{2}\Lambda(\Delta u_k,\Delta u_{k-1}) & \text{if } q_k\geq 0\\ u_{k+1/2}-\frac{1}{2}\Lambda(\Delta u_{k+1},\Delta u_k) & \text{if } q_k<0\end{cases} \tag{7b}$$

with $\Delta n_{k+1/2}:=n_{k+1}-n_k$, $\Delta u_k:=u_{k+1/2}-u_{k-1/2}$, $n_{k+1/2}=\left(n_{k+1}+n_k\right)/2$, $u_{k+1/2}=(nu)_{k+1/2}/n_{k+1/2}$ and $u_k=\left(u_{k+1/2}+u_{k-1/2}\right)/2$ (local shock speed in Burger's equation) and analogously $q_k:=\left(q_{k+1/2}+q_{k-1/2}\right)/2$. We here choose a minmod slope-limiter

$$\Lambda_{\text{minmod}}(x,y):=\begin{cases}\min(x,y) & \text{if } x,y\geq 0\\ \max(x,y) & \text{if } x,y\leq 0\\ 0 & \text{else}\end{cases} \tag{8}$$

noting that there exists a host of other possible choices (see [20]).

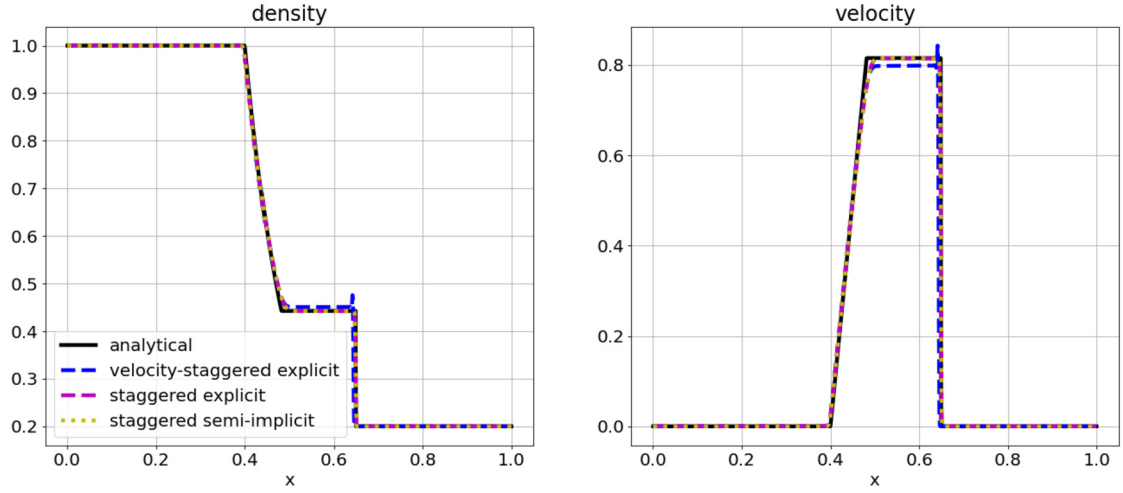

Fig. 1. Plot of the one dimensional Riemann initial condition (12) at $T = 0.1$ and $N_x = 1000$ with the semi-implicit staggered finite volume scheme "staggered semi-implicit" and its completely explicit variant "staggered explicit" (6). Both solutions coincide with the analytical solution. The "velocity-staggered" variant (10) converges to the wrong shock speed and slightly overshoots at the jump.

Furthermore, we look into discretizing the velocity equation over the momentum formulation, which may be interesting for convenience of implementation, but is physically allowed only for $\nu_u \neq 0$ (no shocks) [20].

$$\frac{\partial}{\partial t}u = -\frac{\partial}{\partial x}\left(\frac{u^2}{2} + \ln n\right) + \frac{\nu_u}{n}\frac{\partial^2}{\partial x^2}u \tag{9}$$

This formulation is close to Burger's equation. The best scheme that we found in this formulation is

$$\frac{\mathrm{d}}{\mathrm{d}t}n_k = -\frac{1}{\Delta x}(\hat{q}_{k+1/2} - \hat{q}_{k-1/2}) \tag{10a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{k+1/2} = -\frac{1}{\Delta x}\left(\hat{f}_{k+1} - \hat{f}_k\right) - \frac{1}{\Delta x}\left[\left(n_{k+1} - n_k\right)\frac{1}{2}\left(\frac{1}{n_{k+1}} + \frac{1}{n_k}\right)\right] + \frac{\nu_u}{n_{k+1/2}(\Delta x)^2}\left(u_{k+3/2} - 2u_{k+1/2} + u_{k-1/2}\right) \tag{10b}$$

with

$$\hat{q}_{k+1/2} := u_{k+1/2}\begin{cases} n_k + \frac{1}{2}\Lambda(\Delta n_{k+1/2}, \Delta n_{k-1/2}) & \text{if } u_{k+1/2} \geq 0 \\ n_{k+1} - \frac{1}{2}\Lambda(\Delta n_{k+3/2}, \Delta n_{k+1/2}) & \text{if } u_{k+1/2} < 0 \end{cases} \tag{11a}$$

$$\hat{f}_k := \frac{1}{2}u_k\begin{cases} u_{k-1/2} + \frac{1}{2}\Lambda(\Delta u_k, \Delta u_{k-1}) & \text{if } u_k \geq 0 \\ u_{k+1/2} - \frac{1}{2}\Lambda(\Delta u_{k+1}, \Delta u_k) & \text{if } u_k < 0 \end{cases} \tag{11b}$$

We divide by the harmonic mean in the pressure gradient term, which from empirical tests yields better convergence than other types like the arithmetic or geometric mean.

In the remainder of this section we will call discretization Eqs. (6) the "staggered" scheme and Eqs. (10) the "velocity-staggered" scheme.

2.2. A shock and a wave test problem

Since we use the above schemes as a basis on which we build our three-dimensional scheme it is judicious to further investigate the scheme and to reproduce previous numerical results.

There are various methods to discretize the time in Eqs. (6) and Eqs. (10) as both schemes are represented in the method of lines. The original staggered scheme [22,23,21] was presented with a semi-implicit Euler step, where the pressure gradient in equation (6b) $-[n_{k+1} - n_k]^{n+1}/\Delta x$ was treated implicitly. The resulting implicit equation trivially decouples and can be solved analytically. The procedure can be straightforwardly elevated to higher order semi-implicit timesteppers.

We show a Riemann test problem in Fig. 1. The initial state is given by

$$n_0(x) = \begin{cases} 1 \text{ if } x < 0.5 \\ 0.2 \text{ else} \end{cases} \qquad u_0(x) = 0 \tag{12}$$

We use $N_x = 1000$ points and simulate until $T = 0.1$. We use an adaptive additive Runge-Kutte scheme of third order [28] for the semi-implicit version and a Bogacki-Shampine explicit adaptive Runge-Kutta of third order [29] for the explicit version. Both versions show the same results. The "velocity-staggered" discretization of Eq. (10) does not correctly capture the shock, which is not curable by increasing the resolution further. The authors of the scheme warn that if the pressure term is treated explicitly the staggered scheme loses its shock conserving properties (see Figures 10, 11 of [23]) and is prone to spurious oscillations. Our own extensive investigations and private correspondence with the authors reveal that this in fact only happens if an explicit Euler step is used and the timestep is very close to

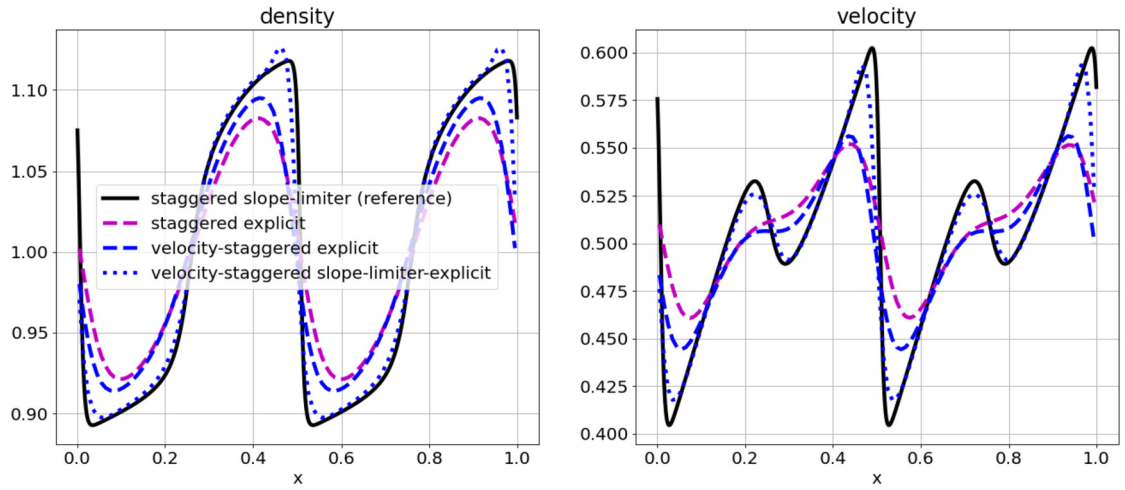

Fig. 2. Plot of the one dimensional Navier-Stokes equation with viscosity $\nu_u = 0.001$ and the sine wave initial condition (13) at $T = 1.5$. The solid black line is a reference solution with $N_x = 1000$ points while the other solutions are computed with $N_x = 100$. The solution with slope-limiter slightly overshoots. The solutions without slope-limiter show numerical diffusion.

the CFL condition. If the timestep is small enough or if any higher order timestepper is used the explicit schemes behave the same as the semi-implicit ones in all our tests. We will therefore only consider the third order in time, explicit Bogacki-Shampine scheme for the remainder of the paper.

In a second test we investigate the performance of the schemes for a non-vanishing viscosity $\nu_u = 0.001$. We initialize a plane wave and impose periodic boundary conditions.

$$n_0(x) = 1 + 0.2\sin(4\pi x) \tag{13a}$$

$$u_0(x) = 0.5 + 0.1\sin(4\pi x) \tag{13b}$$

The result is shown in Fig. 2 at $T = 1.5$. We compute a reference solution with $N_x = 1000$ points while the remaining schemes use $N_x = 100$ points to highlight differences. We find that the "staggered" schemes have slightly more numerical diffusion than the "velocity-staggered" schemes. The variant with minmod slope-limiter is closest to the reference solution but shows a slight overshoot in the density profile. For higher resolution all curves coincide with the reference solution.

Finally, we note that in the above numerical tests the total mass $M := \sum_k n_k \Delta x$ is conserved in time up to machine precision. This is a main feature of the finite-volume discretization. The total momentum and energy are not conserved in the first example due to boundary conditions. In the second example momentum is conserved up to machine precision in the staggered schemes but not in the velocity-staggered schemes due to the non-conservative formulation. Energy is not conserved in any scheme in the second example due to the viscosity.

3. The flux coordinate independent approach

We here develop the major ideas of the flux coordinate independent approach, interpreted as a finite difference scheme in a locally field-aligned coordinate system.

3.1. Constructing the coordinate map and the volume form

Given is a vector field $\hat{\boldsymbol{b}}(R, Z, \varphi)$ in cylindrical coordinates (R, Z, φ). The vector field $\hat{\boldsymbol{b}}$ might be the magnetic unit vector field but the algorithm works for any vector field $\hat{\boldsymbol{b}}$ with $b^\varphi \neq 0$, in particular $\hat{\boldsymbol{b}}$ does not necessarily need to have unit length. Notice here that cylindrical coordinates are no prerequisite to the method. In Appendix A we show how to start from curvilinear coordinates.

As a first step, our idea is to construct a field-aligned coordinate system ρ, ζ, Φ that originates at the plane given by $\varphi = \varphi_k$ for some value φ_k. With originate we mean that for $\Phi = \varphi_k$ we have $\rho(R, Z, \varphi_k) = R$, $\zeta(R, Z, \varphi_k) = Z$. The central motivation for the field-aligned coordinate system is that $\hat{\boldsymbol{b}}$ only has one non-vanishing contravariant component: $b^\Phi(\rho, \zeta, \Phi) = b^\varphi(R(\rho, \zeta, \Phi), Z(\rho, \zeta, \Phi), \varphi(\Phi))$. The field-aligned gradient, divergence and Laplacian in the field-aligned coordinate system thus read

$$\nabla_\parallel F(\rho, \zeta, \Phi) = b^\Phi \partial_\Phi F \tag{14}$$

$$\nabla \cdot (\hat{\boldsymbol{b}} F)(\rho, \zeta, \Phi) = \frac{1}{\sqrt{G}} \partial_\Phi \left(\sqrt{G} b^\Phi F \right) \tag{15}$$

$$\Delta_\parallel F(\rho, \zeta, \Phi) = \frac{1}{\sqrt{G}} \partial_\Phi \left(\sqrt{G} b^\Phi b^\Phi \partial_\Phi F \right) \tag{16}$$

with $F(\rho, \zeta, \Phi) = f(R(\rho, \zeta, \Phi), Z(\rho, \zeta, \Phi), \varphi(\Phi))$. In order to compute the above derivatives (14), (15) and (16) we thus need the coordinate map $(R(\rho, \zeta, \Phi), Z(\rho, \zeta, \Phi), \varphi(\Phi))$ as well as the volume form in the field-aligned system $\sqrt{G}(\rho, \zeta, \Phi)$.

The construction is straightforward. First, we choose $\Phi = \varphi$. All we then have to do is to integrate the field-line equation parameterized by φ (see Reference [30] for details on streamline integration) together with the volume form

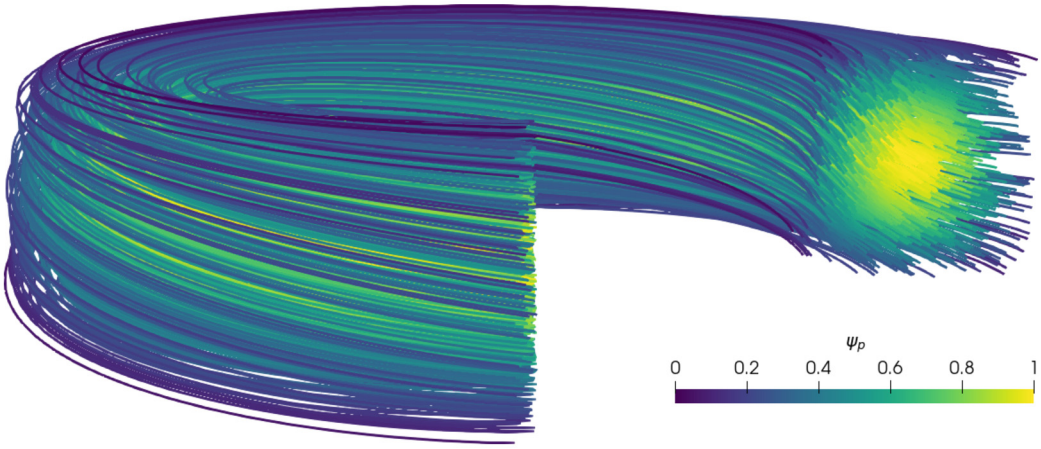

Fig. 3. Plot of the streamlines of the example magnetic field $\hat{\boldsymbol{b}}$ used in this paper (defined by Eq. (53)).

$$\frac{\mathrm{d}R}{\mathrm{d}\varphi} = \frac{b^R}{b^\varphi} \tag{17a}$$

$$\frac{\mathrm{d}Z}{\mathrm{d}\varphi} = \frac{b^Z}{b^\varphi} \tag{17b}$$

$$\frac{\mathrm{d}\sqrt{G}}{\mathrm{d}\varphi} = \nabla\cdot\left(\frac{\hat{\boldsymbol{b}}}{b^\varphi}\right)\sqrt{G} \tag{17c}$$

As initial conditions at φ_k we choose $(R(\varphi_k), Z(\varphi_k)) = (\rho, \zeta)$ and integrate from $\varphi = \varphi_k$ to $\varphi = \Phi$. The resulting $(R(\rho,\zeta,\Phi),\ Z(\rho,\zeta,\Phi),\ \varphi(\Phi))$ completely define the coordinate transformation. Note that we choose $\Phi = \varphi$ because then equidistant planes in Φ coincide with equidistant planes in φ. Another choice would for example be the field-line length s as proposed by [19,16], where $b^s = 1$ simplifies. We note that Eq. (15) with $F = 1/b^\Phi$ yields Eq. (17c). In Appendix B we will show how to construct the full Jacobian and metric tensors for (ρ,ζ,Φ) and rederive Eq. (17c) with the full metric tensor. As initial condition we take $\sqrt{G} = \sqrt{g}$, where $\sqrt{g} = R$ is the volume form in the cylindrical coordinate system.

In Fig. 3 we plot an example field-aligned coordinate system for a magnetic field that we precisely define only later in the text in Eq. (53). The domain is periodic in φ but we show only three quarters to enable a view on the inside. The field-lines, if followed in φ clearly change their R and Z position and if followed for one turn appear at the almost opposite side of the square domain in R-Z.

Finally, the coordinate map defines a pullback operator $\mathcal{P}_k$. Any function $f(R,Z,\varphi)$ defined on cylindrical coordinates can be transformed to the field-aligned system originating at φ_k via

$$F(\rho,\zeta,\Phi) := f(R(\rho,\zeta,\Phi), Z(\rho,\zeta,\Phi), \Phi) \equiv (\mathcal{P}_k f)(\rho,\zeta,\Phi) \tag{18}$$

3.2. Discrete operators in the field-aligned coordinate system

Next, we assume that the φ direction is discretized into N_φ equidistant points φ_k with $k = 0, \ldots, N_\varphi - 1$. We leave the R, Z coordinates undiscretized for now. We further assume that for each of the φ_k values we can construct a field-aligned coordinate system as defined in Section 3.1 separately.

Assume then that we have a function $f(R,Z,\varphi)$ given on the discrete planes $\varphi = \varphi_k$ and set $f_k(R,Z) \equiv f(R,Z,\varphi_k)$. We define

$$\begin{aligned} F_k(\rho,\zeta) :=& F(\rho,\zeta,\Phi_k) = f(R(\rho,\zeta,\Phi_k), Z(\rho,\zeta,\Phi_k), \varphi_k) = f_k(\rho,\zeta) \\ =:& (\mathcal{P}_{k,0} f_k)(\rho,\zeta) \end{aligned} \tag{19}$$

$$\begin{aligned} F_{k\pm1}(\rho,\zeta) :=& F(\rho,\zeta,\Phi_{k\pm1}) = f(R(\rho,\zeta,\Phi_{k\pm1}), Z(\rho,\zeta,\Phi_{k\pm1}), \varphi_{k\pm1}) = f_{k\pm1}(R(\rho,\zeta,\Phi_{k\pm1}), Z(\rho,\zeta,\Phi_{k\pm1})) \\ =:& (\mathcal{P}_{k,\pm\Delta\varphi} f_{k\pm1})(\rho,\zeta) \end{aligned} \tag{20}$$

which means the transformation from the cylindrical to the transformed coordinate system at neighbouring planes is a two-dimensional operation. This is a result of our choice $\Phi = \varphi$. In section 3.3 we will look more closely at how to represent the operations $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,\pm\Delta\varphi}$ numerically, given some discrete form of $f_k(R,Z)$. Note here that even though $\mathcal{P}_{k,0} = \mathbf{1}$ is the identity we formally introduce it here to unify the notation in the next section 3.3.

In fact, the transformation operators $\mathcal{P}_{k,\pm\Delta\varphi}$ and $\mathcal{P}_{k,0}$ represent the flow generated by $\hat{\boldsymbol{b}}/b^\varphi$ at various distances from φ_k, the plane where the coordinate system originates. In other words they are separate instances of the same underlying pullback operator $\mathcal{P}_k$ defined in Eq. (18).

In Fig. 4 we plot the effects of $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,+\Delta\varphi}$ for the example vector field (53). The initial rectangular grid is transported by the field-lines onto the next plane, which clearly deforms and distorts the grid.

Once transformed, it is straightforward to discretize field-aligned derivatives of f. We simply find expressions for the forward (FW), backward (BW) and centred (CC) finite differences from Eq. (14)

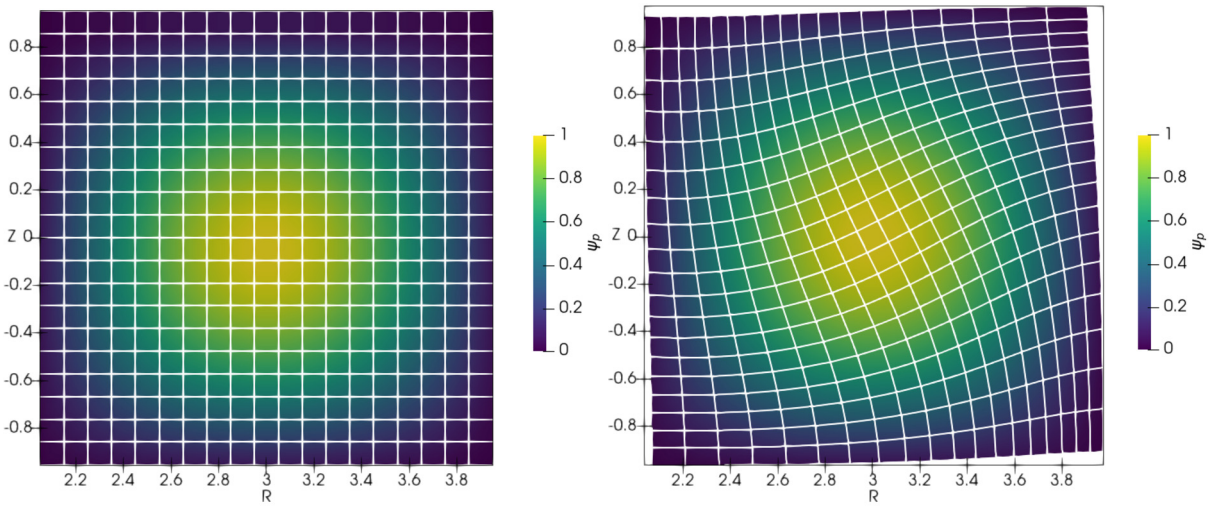

Fig. 4. Plot of the non-aligned coordinate system $\mathcal{P}_{k,0}(R,Z)$ on the left and the transformed coordinate system $\mathcal{P}_{k,+\Delta\varphi}(R,Z)$ on the right. We use the field defined by Eq. (53).

$$\nabla_\parallel^{FW} F_k := b_k^\Phi \frac{F_{k+1} - F_k}{\Delta\Phi} \tag{21a}$$

$$\nabla_\parallel^{BW} F_k := b_k^\Phi \frac{F_k - F_{k-1}}{\Delta\Phi} \tag{21b}$$

$$\nabla_\parallel^{CC} F_k := b_k^\Phi \frac{F_{k+1} - F_{k-1}}{2\Delta\Phi} \tag{21c}$$

For the divergences we find from Eq. (15)

$$\nabla\cdot(\hat{\boldsymbol{b}}F)_k^{FW} := \frac{\Gamma_{k,1} b_{k+1}^\Phi F_{k+1} - \Gamma_{k,0} b_k^\Phi F_k}{\Delta\Phi} \tag{22a}$$

$$\nabla\cdot(\hat{\boldsymbol{b}}F)_k^{BW} := \frac{\Gamma_{k,0} b_k^\Phi f_k - \Gamma_{k,-1} b_{k-1}^\Phi f_{k-1}}{\Delta\Phi} \tag{22b}$$

$$\nabla\cdot(\hat{\boldsymbol{b}}F)_k^{CC} := \frac{\Gamma_{k,1} b_{k+1}^\Phi F_{k+1} - \Gamma_{k,-1} b_{k-1}^\Phi F_{k-1}}{2\Delta\Phi} \tag{22c}$$

Finally, for the parallel Laplacian defined by Eq. (16) we find

$$\Delta_\parallel^{CC} F_k := \frac{1}{\Delta\Phi^2}\left[\Gamma_{k,1/2} b_{k+1/2}^\Phi b_{k+1/2}^\Phi (F_{k+1} - F_k) - \Gamma_{k,-1/2} b_{k-1/2}^\Phi b_{k-1/2}^\Phi (F_k - F_{k-1})\right] \tag{23}$$

where we define

$$\Gamma_{k,l} := \frac{\sqrt{G^{k+l}}}{\sqrt{G^k}} \tag{24}$$

$$\sqrt{G^{k\pm1/2}} := 0.5\left(\sqrt{G^{k\pm1}} + \sqrt{G^k}\right) \tag{25}$$

$$b_{k\pm1/2}^\Phi := 0.5\left(b_{k\pm1}^\Phi + b_k^\Phi\right) \tag{26}$$

Notice that only the relative volume ratios $\Gamma_{k,+1}$ and $\Gamma_{k,-1}$ need to be stored ($\Gamma_{k,0} = 1$) in favour of the three elements $\sqrt{G^k}$, $\sqrt{G^{k+1}}$ and $\sqrt{G^{k-1}}$. The results are given in the transformed coordinate system (ρ,ζ), however, with $\mathcal{P}_{k,0} = \mathbf{1}$ it is trivial to transform back to the original (R,Z). We remark that the suggested expressions for the parallel gradient Eq. (21) were suggested before [31], while the specific expressions for the parallel divergence Eq. (15) and the parallel Laplacian Eq. (16) are here presented for the first time. The discretization of the Laplacian in Eq. (23) is self-adjoint with respect to the scalar product given by $\sqrt{G}$, while the divergences (22) are the negative adjoints to their respective parallel derivative (21). This is true if we can assume periodicity in Φ. For example we have

$$\langle H, \nabla\cdot(\hat{\boldsymbol{b}}F)^{FW}\rangle = \Delta\Phi\sum_k H_k \nabla\cdot(\hat{\boldsymbol{b}}F)_k^{FW}\sqrt{G^k} = -\Delta\Phi\sum_k \nabla_\parallel^{BW} H_k\sqrt{G^k} = -\langle\nabla_\parallel^{BW} H, F\rangle \tag{27}$$

However, this property does not automatically translate to the adjoint with respect to the cylindrical volume element in cylindrical coordinates. This is due to the numerical coordinate transformations that we discuss in the next Section 3.3.

3.3. Coordinate transformations

It now remains to devise a suitable discretization of the (R, Z), respectively (ρ, ζ) planes and in particular find discrete expressions for the pullback operators $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,\pm\Delta\varphi}$. In order to simplify the notation of this section we present our ideas in a one-dimensional setting (i.e. for x and transformed coordinate systems X^-, X^0, X^+ corresponding to the three operators $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,\pm\Delta\varphi}$) noting that the generalization to two dimensions is immediate and directly applicable. The important realization is that there is more than one way to discretize these operators. Here, we will present two methods in two variations each to discretize $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,\pm\Delta\varphi}$ that were suggested in the literature and that we here present in a generalized setting. The two methods are the "interpolation-method" in Section 3.3.2 and the "projection-method" in Section 3.3.3. The variations consist in starting with either a discontinuous Galerkin (dG) or a finite element (FE) expansion. To familiarize the reader with these expansions we begin to introduce notation and some preliminary results in Section 3.3.1.

3.3.1. Discontinuous Galerkin and finite element methods

In a discontinuous Galerkin (dG) scheme functions $f(x)$ are represented by [25]

$$f_h(x) = \sum_{n=1}^{N}\sum_{k=0}^{P-1} \bar{f}^{nk} p_{nk}(x) = \sum_{n=1}^{N}\sum_{k=0}^{P-1} f_{nk} l^{nk}(x), \tag{28}$$

where the polynomials p_{nk} form an orthogonal basis of the space of polynomials of degree $P-1$ in cell n, while l^{nk} is the k-th Lagrange interpolation polynomial in cell n. The *modal* values $\bar{f}^{nk}$ are uniquely related to the *nodal* values $f_{nq} := f(x_{nq})$ via the $P \times P$ invertible Vandermonde matrix V [25] $f_{nq} = V_{qk}\bar{f}^{nk}$ with elements $V_{qk} = p_k(x_{nq})$. We here choose p_{nk} as the Legendre polynomials and x_{nq} as the corresponding Legendre-Gauss-Lobatto quadrature points. We have N cells in total with P polynomial coefficients in each cell. The Legendre and Lagrange polynomials are related by

$$l^{nk}(x) = \sum_{q=0}^{P-1} p_{nq}(x) \left(V^{-1}\right)^{qk} \tag{29}$$

from where we directly have $l^{nk}(x_{mq}) = \delta_m^n \delta_q^k$, with the Kronecker δ.

Legendre polynomials are orthogonal in cell C_n: $\int_{C_n} p_{nq}(x)p_{nk}(x)\mathrm{d}x = \delta_{kq}h/(2k+1)$. We denote two $P \times P$ matrices W and S by

$$W_{ij} := w_j\delta_{ij}, \quad S_{ij} := \frac{h}{2i+1}\delta_{ij} \tag{30}$$

With W and S we can write the scalar product between two dG approximated functions

$$\int f_h(x)g_h(x)\mathrm{d}x = \boldsymbol{f}^{\mathrm{T}}(\mathbf{1}\otimes W)\boldsymbol{g} = \bar{\boldsymbol{f}}^{\mathrm{T}}(\mathbf{1}\otimes S)\bar{\boldsymbol{g}} \tag{31}$$

where $\mathbf{1}$ is an $N \times N$ identity matrix and we integrate over the entire volume. Thus, S and W are expressions of the metric in modal respectively nodal space. It can be shown that

$$V^{-1} = S^{-1}V^{\mathrm{T}}W = V^{\dagger} \tag{32}$$

which shows that the Vandermonde matrix is unitary with the adjoint defined with respect to the metric in nodal and modal space.

Suppose we have a dG discretization as defined in Eq. (28), i.e. a function given on N cells with P polynomial coefficients. We now want to divide each cell C_n of the original grid into K equidistant subcells. We use the letter c to denote the original grid as a coarse grid with N cells and the letter f to denote the fine grid with KN cells. We denote $q_{ml}(x)$ the polynomials on the fine grid and x^F_{mj} the corresponding Gaussian abscissas. For a discretization $f_h(x)$ represented by the vector $\boldsymbol{f}_c$ given on the coarse grid, we can simply interpolate onto the fine grid via

$$f^F_{mj} := f_h(x^F_{mj}) = \sum_{n=1}^{N}\sum_{k,i=0}^{P-1} p_{nk}(x^F_{mj})\left(V^{-1}\right)^{ki}_{C} f^C_{ni} =: \sum_{n=1}^{N}\sum_{i=0}^{P-1} (Q_{DG})_{mj}{}^{ni} f^C_{ni} \tag{33}$$

where we implicitly define the special interpolation matrix Q_{DG}, which has KNP lines and NP columns. It transforms vectors given on the coarse grid to the fine grid. No information is lost if the number of cells in the fine grid is an integer multiple K of the coarse grid and the number of polynomial coefficients in the fine grid is the same or higher: f^F and f^C represent exactly the same expansion $f_h(x)$.

$$f^C(x) = \sum_{n=1}^{N}\sum_{k=0}^{P-1} \bar{f}^{nk}_C p_{nk}(x) = f^F(x) = \sum_{m=1}^{KN}\sum_{l=0}^{P-1} \bar{f}^{ml}_F q_{ml}(x) \tag{34}$$

This can be seen since the polynomial in each cell is unique.

Vice versa, given an expansion $f^F(x)$ on the fine grid, we can compute the projection integrals from the fine grid to the coarse grid.

$$\bar{f}^{nk}_C := \sum_{s=0}^{P-1}\left(S_C^{-1}\right)^{ks}\int_{C_n} f^F(x)p_{ns}(x)\,\mathrm{d}x \tag{35}$$

We can show that

$$
\begin{aligned}
f_{nt}^C &= \sum_{k=0}^{P-1} (V_C)_{tk}\, \bar{f}_C^{nk} = \sum_{m=1}^{K}\sum_{ksij=0}^{P-1} (V_C)_{tk} \left(S_C^{-1}\right)^{ks} (W_F)^{ij}\, p_{ns}\left(x_{mj}^F\right) f_{mi}^F = \sum_{m=1}^{K}\sum_{ksij=0}^{P-1} \left(W_C^{-1}\right)_{tk} \left(V_C^{-1}\right)^{sk} (W_F)^{ij}\, p_{ns}\left(x_{mj}^F\right) f_{mi}^F \\
&= \sum_{m=1}^{K}\sum_{kij=0}^{P-1} \left(W_C^{-1}\right)_{tk} (Q_{DG})_{mj}{}^{nk} (W_F)^{ji}\, f_{mi}^F = \sum_{m=1}^{K}\sum_{kij=0}^{P-1} \left(W_C^{-1}\right)_{tk} \left(Q_{DG}^T\right)^{nk}{}_{mj} (W_F)^{ji}\, f_{mi}^F \\
&=: \sum_{m=1}^{K}\sum_{i=0}^{P-1} (P_{DG})_{nt}{}^{mi}\, f_{mi}^F
\end{aligned}
$$

The first identity is simply evaluating the projection integral on the fine grid using Gauss-Legendre integration and the fact that we have the fine expansion f_{mi}^F given. Note here that the polynomial order on the coarse grid may not exceed that of the fine grid as otherwise the integration is no longer exact. The second identity uses Eqs. (32) and (33). Note that these relations hold even if the number of polynomial coefficients is lower in the coarse and the fine grid. From here we directly conclude that

$$
P_{DG} = Q_{DG}^\dagger = W_C^{-1} Q_{DG}^T W_F \tag{36}
$$

i.e. the projection matrix is the adjoint of the interpolation matrix. We can also prove that

$$
P_{DG} \circ Q_{DG} = 1_C \tag{37}
$$

which is a reformulation of Eq. (34), however, $Q_{DG} \circ P_{DG} \neq 1$. The projection is not loss-free but it conserves the integral value of the function on the fine grid.

Instead of a discontinuous Galerkin approach another natural approach is a continuous finite element (FE) discretization, where

$$
f_h(x) = \sum_{n=1}^{N} f_n v^n(x) \tag{38}
$$

where $v_n(x)$ is the finite element in cell n. For example we can choose triangular elements

$$
v^i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} \text{ if } x_{i-1} \leq x < x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} \text{ if } x_i \leq x < x_{i+1} \\ 0 \text{ else} \end{cases} \tag{39}
$$

A similar result to Eq. (37) can be derived for the finite element expansion. First we note that $(W_{FE})_{ij} := \delta_{ij} \int \mathrm{d}x v_i(x) = h\delta_{ij}$ and $(S_{FE})ij := \frac{1}{w_j} \int \mathrm{d}x v_i(x) v_j(x) = \frac{1}{6}[1,4,1]_{ij}$ where the notation $[1,4,1]$ denotes a tridiagonal matrix with 4 on the main diagonal. Note that we do not show boundary entries or non-equidistant grids for ease of notation here but the definitions hold in general. With similar arguments as before we can show that on a "fine" grid resulting from subdividing each cell in a given coarse grid into K equidistant subcells we can define

$$
(Q_{FE})_m^n := v_m\left(x_n^F\right) \tag{40}
$$

$$
P_{FE} := W_{FE,C}^{-1} (Q_{FE})^{\mathrm{T}} W_{FE,F} \tag{41}
$$

and show that

$$
P_{FE} \cdot Q_{FE} = S_{FE} \tag{42}
$$

Notice here the appearance of S_{FE} in contrast to 1_C in Eq. (37), which fundamentally is a result of the fact that the finite elements are not orthogonal to each other while Legendre polynomials are.

3.3.2. A simple interpolation scheme

A first method to discretize $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,\pm\Delta\varphi}$ is to directly use the dG expansion in Eq. (28). We have

$$
F_h(X^\pm) = F_{nk} l^{nk}\left(X^\pm\right) = \sum_{n=1}^{N}\sum_{k=0}^{P-1} f_h\left(x\left(X_{nk}^\pm\right)\right) l^{nk}\left(X^\pm\right) = \sum_{m,n=1}^{N}\sum_{q,k=0}^{P-1} f_{mq} l^{mq}\left(x\left(X_{nk}^\pm\right)\right) l^{nk}\left(X^\pm\right)
$$

We now define the interpolation matrix

$$
\left(I_{DG}^\pm\right)_{nk}^{mq} := l^{mq}\left(x\left(X_{nk}^\pm\right)\right) \tag{43}
$$

which reflects that the transformation from x to $X^\pm$ coordinates is a linear operation as expected. $I^\pm$ is the matrix that corresponds to $\mathcal{P}_{k,\pm\Delta\varphi}$. Furthermore, we have that $F_h(X^0) = f_h(x(X^0)) = f_h(X^0)$, i.e. $I^0 = \mathbf{1}$ is the identity corresponding to $\mathcal{P}_{k,0}$. Note that the order of the interpolation is given by the order P of the polynomials in use.

In the case of a finite element expansion Eq. (38) the transformation operator reads

$$
\left(I_{FE}^\pm\right)_n^m := v^m\left(x\left(X_n^\pm\right)\right) \tag{44}
$$

If we choose triangular elements Eq. (39) we have a linear interpolation while for constant elements we have a first order nearest neighbour interpolation. The piecewise constant elements are identical to a dG approximation with $P = 1$. We also implemented a cubic interpolation scheme for comparison. In all above cases we have

$$\boldsymbol{F}^{\pm} = I^{\pm}\boldsymbol{f} \tag{45a}$$

$$\boldsymbol{F}^{0} = \boldsymbol{f} \tag{45b}$$

The dG interpolation matrix (43) coincides with the one proposed in [16] while the finite element interpolation in Eq. (44) reproduces the results in [18,19].

3.3.3. A projection scheme

There is a second method of numerical coordinate transformation that comes naturally in a finite element setting, which is to compute the projection integrals over the finite elements in the transformed coordinate system. We can write in the transformed coordinate system

$$\bar{F}^{nk} := \sum_{s=0}^{P-1} \left(S^{-1}\right)^{ks} \int_{C_n} F_h\left(X^{\pm}\right) p_{ns}\left(X^{\pm}\right) \mathrm{d}X^{\pm} = \sum_{s=0}^{P-1} \left(S^{-1}\right)^{ks} \int_{C_n} f_h\left(x\left(X^{\pm}\right)\right) p_{ns}\left(X^{\pm}\right) \mathrm{d}X^{\pm} \tag{46}$$

Since we are only able to compute the coordinate transformation numerically we have to resort to a numerical integration scheme. First, we realize that the above expression is completely analogous to Eq. (35) if $F_h(X^{\pm})$ was an expansion on a fine grid. To obtain such an expansion we again divide each cell C_n into K equidistant subcells and approximate

$$F_h(X^{\pm}) = \sum_{n=1}^{KN}\sum_{k=0}^{P-1} F^F_{nk} l^{nk}_F\left(X^{\pm}\right) = \sum_{n=1}^{KN}\sum_{k=0}^{P-1} f_h\left(X^{\pm}_{F,nk}\right) l^{nk}_F\left(X^{\pm}\right) = \sum_{n=1}^{KN}\sum_{m=1}^{N}\sum_{qk=0}^{P-1} f_{mq} l^{mq}\left(x\left(X^{\pm}_{F,nk}\right)\right) l^{nl}_F\left(X^{\pm}\right) \tag{47}$$

We can define

$$\left(I^{\pm}_{DG,F}\right)^{mq}_{nk} := l^{mq}\left(x\left(X^{\pm}_{F,nk}\right)\right) \tag{48}$$

as the interpolation matrix that interpolates a vector given on a coarse grid in x to a fine grid on $X^{\pm}$. $X^{\pm}_{F,nk}$ are the Legendre-Gauss-Lobatto quadrature points in the fine grid. The fine interpolation matrix $I^{\pm}_{DG,F}$ has PNK rows and PN columns whereas $I^{\pm}_{DG}$ has PN rows and columns.

Inserting Eq. (47) back into Eq. (46) we thus arrive at the transformation

$$\boldsymbol{F}^{\pm} = P_{DG} I^{\pm}_{DG,F} \boldsymbol{f} \tag{49a}$$

$$\boldsymbol{F}^{0} = P_{DG} Q_{DG} \boldsymbol{f} = \boldsymbol{f} \tag{49b}$$

For the finite element discretization we have a similar result

$$\boldsymbol{F}^{\pm} = S^{-1}_{FE} P_{FE} I^{\pm}_{FE,F} \boldsymbol{f} \tag{50a}$$

$$\boldsymbol{F}^{0} = S^{-1}_{FE} P_{FE} Q_{FE} \boldsymbol{f} = \boldsymbol{f} \tag{50b}$$

The necessary inversion S^{-1}_{FE} is the main downside of the FE method compared to the dG method as it requires the implementation of an efficient and dedicated inversion algorithm.

However, a major realization in the development of the three-dimensional advection schemes is to approximate $S^{-1}_{FE} \approx \mathbf{1}$ and use

$$\boldsymbol{F}^{\pm} := P_{FE} I^{\pm}_{FE,F} \boldsymbol{f} \tag{51a}$$

$$\boldsymbol{F}^{0} := P_{FE} Q_{FE} \boldsymbol{f} = S_{FE} \boldsymbol{f} \tag{51b}$$

Here, we realize that S_{FE} effectively is a smoothing operator (it computes the convolution of $f_h(x)$ with the "window functions" $v_i(x)$). It turns out that this smoothing leads to a slight perpendicular diffusion that helps to improve stability in the resulting scheme that we present in Section 4. Eq. (51) is thus preferrable to use over Eq. (50).

3.4. Discussion

In summary we have derived discretizations for $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,\pm\Delta\varphi}$ in Eqs. (43) and (49) for the dG expansion as well as Eqs. (44) and (51) for the FE expansion. Each of those expansions can yield various order methods, for example by varying P in the dG expansion we arrive at a P-th order discretization or by changing linear finite elements to cubic finite elements we arrive at a third order method for the FE expansion.

Still, it is possible to construct further discretizations. We could, for example, mix various interpolation and projection schemes in Eq. (51) and use a linear interpolation to the fine grid and project the result on dG polynomials on the coarse grid. We explored those options but found no fundamental changes in the behaviour of the results compared to the original suggestions. The only option we use in practice is to project on piecewise constant elements when we use linear interpolation, which we denote by (cFE). The effect this has is $(S_{cFE})_{ij} \equiv (P_{cFE} Q_{FE})_{ij} = [1/8, 6/8, 1/8]$, which is slightly less diffusive than S_{FE}. Replacing the linear FE projection P_{FE} with constant projection P_{cFE} and S_{FE} with S_{cFE} in the transformation (51) we recover the projection method proposed in [32]. Eqs. (49) and Eqs. (50) are newly derived here.

In the remaining text we will first convince the reader that the various discretization schemes discussed so far do converge in Section 3.5, while we will study which discretization is optimal in a parallel advection scheme in Section 5.

Table 1
Convergence Table for $K = 1$ and $N_R = N_Z = N$ with testfunction (54) and error definition (55), using the dG interpolation Eq. (43).

		$\nabla_\parallel^{CC}$ Eq. (21c)		$\nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$ Eq. (22c)		$\Delta_\parallel^{CC}$ Eq. (23)		$\int dV \nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$		$\int dV \Delta_\parallel^{CC} f$	
		error	order	error	order	error	order	error	order	error	order
N	N_φ										
10	**5**	8.01e-01	n/a	8.07e-01	n/a	4.95e-01	n/a	-7.01e-07	n/a	9.93e-07	n/a
16	**10**	2.74e-01	1.55	2.79e-01	1.53	1.56e-01	1.67	6.03e-08	3.54	2.06e-08	5.59
26	**20**	7.45e-02	1.88	7.59e-02	1.88	4.15e-02	1.91	-1.70e-07	-1.49	-2.13e-07	-3.37
40	**40**	1.90e-02	1.97	1.94e-02	1.97	1.05e-02	1.98	7.85e-08	1.11	-2.79e-07	-0.39
64	**80**	4.78e-03	1.99	4.88e-03	1.99	2.64e-03	1.99	-1.56e-08	2.33	3.74e-08	2.90

Table 2
Convergence Table for $K = 12$ and $N_R = N_Z = N$ with testfunction (54) and error definition (55), using the dG projection Eq. (49).

		$\nabla_\parallel^{CC}$ Eq. (21c)		$\nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$ Eq. (22c)		$\Delta_\parallel^{CC}$ Eq. (23)		$\int dV \nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$		$\int dV \Delta_\parallel^{CC} f$	
		error	order	error	order	error	order	error	order	error	order
N	N_φ										
10	**5**	8.01e-01	n/a	8.07e-01	n/a	4.95e-01	n/a	3.03e-07	n/a	8.86e-08	n/a
16	**10**	2.74e-01	1.55	2.79e-01	1.53	1.56e-01	1.67	5.10e-09	5.89	-4.57e-09	4.28
26	**20**	7.45e-02	1.88	7.59e-02	1.88	4.15e-02	1.91	-5.00e-10	3.35	1.48e-09	1.62
40	**40**	1.90e-02	1.97	1.94e-02	1.97	1.05e-02	1.98	-3.26e-10	0.62	-7.57e-10	0.97
64	**80**	4.78e-03	1.99	4.88e-03	1.99	2.64e-03	1.99	2.49e-09	-2.94	8.40e-10	-0.15

3.5. Numerical verification of convergence in an example magnetic field

Having developed all ingredients of the FCI scheme we can now proceed to a first numerical test. We choose an axisymmetric magnetic field given by

$$\boldsymbol{B} = \frac{R_0}{R}(I(\psi_p)\hat{e}_\varphi + \nabla\psi_p \times \hat{e}_\varphi) \tag{52}$$

which gives rise to a unit vector field $\hat{\boldsymbol{b}} = \boldsymbol{B}/B$ with $B = \frac{R_0}{R}\sqrt{I^2 + \left(\nabla\psi_p\right)^2}$. Note that the symmetry makes the transformation operators $\mathcal{P}_{k,0}$ and $\mathcal{P}_{k,\pm\Delta\varphi}$ independent of k but is no necessity for the scheme to work.

A simple choice for the flux function that makes field-lines stay inside a square box (which allows us to avoid boundary conditions for now) is [33]

$$\psi_p = \cos(\pi(R - R_0)/2)\cos(\pi Z/2) \quad I = I_0 \tag{53}$$

with $I_0 = 10$ and $R_0 = 3$. We set up a domain $[R_0 - 1, R_0 + 1] \times [-1, 1] \times [0, 2\pi]$ and construct a testfunction

$$f(R, Z, \varphi) = \exp(R - R_0)\exp(Z)\cos^2(\varphi) \tag{54}$$

With the given identities we can analytically compute $\nabla_\parallel f$, $\Delta_\parallel f$ and $\nabla\cdot(\hat{\boldsymbol{b}}f)$ and thus test the proposed discretizations. We compute relative errors using

$$\varepsilon := \frac{||g_{\text{ana}} - g_{\text{num}}||}{||g_{\text{ana}}||} \tag{55}$$

for any analytical result function g_{ana} and numerical result g_{num}. The norm is given by the L_2-norm $||g||^2 := \int \mathrm{d}V\, g^2$

In Tables 1, 2 and 3 we show resulting errors ε for increasing numbers of $N_\varphi = N_\Phi$. From two consecutive errors we can compute an order of the method via

$$\mathcal{O} := -\frac{\ln\left(\varepsilon(N_{\varphi,1})/\varepsilon(N_{\varphi,2})\right)}{\ln\left(N_{\varphi,1}/N_{\varphi,2}\right)} \tag{56}$$

The first Table 1 is based on the simple interpolation matrix using the discontinuous Galerkin scheme presented in Eq. (43) inserted in centered differences. The order of the discretizations for $\nabla_\parallel f$, $\Delta_\parallel f$ and $\nabla\cdot(\hat{\boldsymbol{b}}f)$ is almost exactly 2 for all resolutions. In the last two columns we show the volume integrals for $\nabla\cdot(\hat{\boldsymbol{b}}f)$ and $\Delta_\parallel f$. Analytically the volume integrals should be zero as the boundary is aligned with the vector field. We thus normalize the error ε with $||f||$ instead of $||g_{\text{ana}}||$ to avoid division by zero. As can be seen, the volume error is between 10^{-7} and 10^{-8} for all resolutions and there is no convergence order discernible.

The next Table 2 shows the same tests for the projection method given in Eq. (49). Comparing Table 1 with 2 shows no difference between the interpolation and the projection method regarding the errors and orders of $\nabla_\parallel f$, $\Delta_\parallel f$ and $\nabla\cdot(\hat{\boldsymbol{b}}f)$. However, the volume conservation is almost two orders of magnitude better for the projection method than for the interpolation method.

In the last Table 3 we show results for the finite element projection method given by Eq. (51) again with $K = 12$. Convergence of order two is again almost perfectly reached for $\nabla_\parallel f$ and $\nabla\cdot(\hat{\boldsymbol{b}}f)$ and $\Delta_\parallel f$. In fact, the errors and orders are almost entirely equal to the dG test, which may be because our test function (54) does not vary quickly across the R and Z coordinates such that both the dG and the linear interpolation errors are negligible. The volume conservation is generally worse than for the dG scheme especially for $\Delta_\parallel$.

Table 3
Convergence Table for $K = 12$, $N_R = N_Z = N$ with testfunction (54) and error definition (55) using the finite element projection method Eq. (51).

		$\nabla_\parallel^{CC}$ Eq. (21c)		$\nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$ Eq. (22c)		$\Delta_\parallel^{CC}$ Eq. (23)		$\int dV\nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$		$\int dV\Delta_\parallel^{CC}f$	
		error	order	error	order	error	order	error	order	error	order
N	N_φ										
8	**5**	8.01e-01	n/a	8.07e-01	n/a	4.96e-01	n/a	2.03e-05	n/a	-1.16e-05	n/a
15	**10**	2.74e-01	1.55	2.79e-01	1.53	1.56e-01	1.67	1.15e-06	4.14	-5.74e-07	4.34
30	**20**	7.45e-02	1.88	7.60e-02	1.88	4.15e-02	1.91	-3.99e-08	4.85	1.55e-06	-1.43
60	**40**	1.90e-02	1.97	1.94e-02	1.97	1.05e-02	1.98	1.99e-08	1.00	-1.76e-06	-0.19
120	**80**	4.78e-03	1.99	4.88e-03	1.99	2.65e-03	1.99	-3.57e-09	2.48	1.24e-06	0.50

Further tests reveal that care must be taken for sufficient resolution in N. Compared to the dG results we use an increased number of cells N. This is because linear interpolation is only order two while the dG interpolation converges with third order. Note that a linear interpolation (44) (table shown in accompanying dataset) shows similar order 2 convergence but worse volume conservation than the projection scheme as expected.

4. Parallel advection schemes

We have now derived all ingredients necessary to elevate our one-dimensional finite-volume discretization to three dimensions. There are two ways of achieving this: the "value-centred" approach and the "flux-centred" approach. We will in this section only show the formulation of the "velocity-staggered" scheme (10). The formulation for the "staggered" scheme (6) is analogous.

4.1. Field-aligned finite volumes

Before we begin to show the two approaches we first explicitly make the connection between the three-dimensional finite volumes and the projection scheme we derived in Section 3.3. We begin by constructing a locally aligned finite element $P_{lk}(\rho,\zeta,\Phi) := p_l(\rho,\zeta)\Theta_k(\Phi)$ as the finite element (either dG or linear finite element) in the k-th field-aligned coordinate system. $p_l(\rho,\zeta)$ is constant along the field-line, while $\Theta_k(\Phi) = 1$ for $\varphi_k - \Delta\varphi/2 \le \Phi \le \varphi_k + \Delta\varphi/2$ and zero elsewhere. We denote $S_{lkmq} = \int_{C_k} P_{lk}P_{mq}\mathrm{d}\rho\mathrm{d}\zeta\mathrm{d}\Phi = \Delta\varphi\delta_{kq}\int p_l p_m \mathrm{d}\rho\mathrm{d}\zeta$. We start with the continuity equation $\partial_t n = -\nabla\cdot(\hat{\boldsymbol{b}}un)$, multiply by P_{lk} and integrate over the volume to get

$$\int \frac{\partial n}{\partial t}(\rho,\zeta,\Phi,t)P_{lk}\mathrm{dV} = -\int P_{lk}\nabla\cdot(\hat{\boldsymbol{b}}un)\mathrm{dV} = -\int \nabla\cdot(\hat{\boldsymbol{b}}P_{lk}un)\mathrm{dV} = \left[\int P_{lk}\hat{q}\hat{\boldsymbol{b}}\cdot\mathbf{dA}\right]_{\Phi=\varphi_k-\Delta\varphi/2}^{\Phi=\varphi_k+\Delta\varphi/2}$$
$$= \left[\int \sqrt{G}b^\Phi p_l\hat{q}\mathrm{d}\rho\mathrm{d}\zeta\right]_{\Phi=\varphi_k-\Delta\varphi/2}^{\Phi=\varphi_k+\Delta\varphi/2}$$

where $[f]_a^b := f(b) - f(a)$. We use that P_{lk} is field-aligned and its parallel derivative vanishes inside the cell. Further, $\hat{q}$ is the chosen flux on the cell boundary. Now, we assume that $n = \sum_{k=1}^{N_\varphi}\sum_l \bar{n}^{mq}P_{mq}$ and further assume that the volume form $\sqrt{G}$ and b^Φ do not vary strongly inside the chosen volume such that we have

$$\sum_{k=1}^{N_\varphi}\sum_l S_{lkmq}\sqrt{G}(\Phi_k)\bar{n}^{mq} = \left[\sqrt{G}b^\Phi\int p_l\hat{q}\mathrm{d}\rho\mathrm{d}\zeta\right]_{\Phi=\varphi_k-\Delta\varphi/2}^{\Phi=\varphi_k+\Delta\varphi/2} \tag{57}$$

Now, there are two possibilities on how $\hat{q}$ is given. Either, $\hat{q}$ is computed inside the field-aligned system, which yields $\hat{q} = \hat{q}^{kl}p_{kl}$ and thus Eq. (57) simplifies to the expression of the divergence in Eq. (22c). This will be the case for the "value-centred" approach discussed in Section 4.2. The other possibility is that $\hat{q}$ is given in cylindrical coordinates (R,Z) and thus has to be transformed to field-aligned coordinates. Then Eq. (57) is also an expression for Eq. (22c) but this time we have to compute the projection integral numerically with the help of the methods described in Section 3.3.3. We call this method the "flux-centred" approach in Section 4.3.

Note that moving the volume and magnetic field out of the integral in Eq. (57) simplifies our approach since the projection integrals can be easily computed. However, this introduces an error in the exact (mass) conservation of the scheme that we consider small from the Tables 2 and 3 but prevents us from reaching machine precision.

4.2. Value-centred approach

The main idea of the "value-centred" approach is to first transform all quantities to the locally field-aligned coordinate system and then compute the entire one-dimensional scheme in the transformed coordinates. We have to do this twice, once for the coordinate system centred on k associated with the density equation (10a) and once for the coordinate system centred on the plane $k+1/2$ associated with the velocity equation (10b). We call this approach "value-centred" because the values $n_k(R,Z)$ and $u_{k+1/2}(R,Z)$ are associated with these planes.

Care must be taken as we need to integrate the field-lines not only to construct $\mathcal{P}_{k,+\Delta\varphi}$ and $\mathcal{P}_{k,-\Delta\varphi}$, but also to transform between the half planes halfway between the original planes $\varphi_{k+1/2}$ with $\mathcal{P}_{k,+\Delta\varphi/2}$ and $\mathcal{P}_{k,-\Delta\varphi/2}$. We denote quantities transformed into the coordinate system centred on k as

$$\bar{n}_{k-2} = \mathcal{P}_{k,-2\Delta\varphi} n_{k-2}, \quad \bar{n}_{k-1} = \mathcal{P}_{k,-\Delta\varphi} n_{k-1}, \quad \bar{n}_k = \mathcal{P}_{k,0} n_k, \quad \bar{n}_{k+1} = \mathcal{P}_{k,+\Delta\varphi} n_{k+1}, \quad \bar{n}_{k+2} = \mathcal{P}_{k,+2\Delta\varphi} n_{k+2}$$
$$\bar{u}_{k-1/2} = \mathcal{P}_{k,-\Delta\varphi/2} u_{k-1/2}, \quad \bar{u}_{k+1/2} = \mathcal{P}_{k,+\Delta\varphi/2} u_{k+1/2} \tag{58}$$

With these definitions we can discretize the continuity equation by

$$\frac{\mathrm{d}}{\mathrm{d}t} n_k = -\frac{\bar{\Gamma}_{k,1/2} \bar{b}^{\varphi}_{k+1/2} \bar{q}_{k+1/2} - \bar{\Gamma}_{k,-1/2} \bar{b}^{\varphi}_{k-1/2} \bar{q}_{k-1/2}}{\Delta\varphi} \tag{59}$$

where the right hand side is a discretization of the divergence similar to Eq. (22c). We have $\bar{\Gamma}_{k,\pm 1/2}$ and $\bar{b}^{\varphi}_{k\pm 1/2}$ as the volume form ratio and the contravariant φ component of $\hat{\boldsymbol{b}}$ on the half planes and

$$\bar{q}_{k+1/2} := \bar{u}_{k+1/2} \begin{cases} \bar{n}_k + \frac{1}{2}\Lambda(\Delta\bar{n}_{k+1}, \Delta\bar{n}_k) & \text{if } \bar{u}_{k+1/2} \geq 0 \\ \bar{n}_{k+1} - \frac{1}{2}\Lambda(\Delta\bar{n}_{k+2}, \Delta\bar{n}_{k+1}) & \text{if } \bar{u}_{k+1/2} < 0 \end{cases} \tag{60}$$

Next, we denote quantities transformed into the coordinate system centred on the half plane $k+1/2$ as

$$\widetilde{u}_{k-3/2} = \mathcal{P}_{k+1/2,-2\Delta\varphi} u_{k-3/2}, \quad \widetilde{u}_{k-1/2} = \mathcal{P}_{k+1/2,-\Delta\varphi} u_{k-1/2},$$
$$\widetilde{u}_{k+1/2} = \mathcal{P}_{k+1/2,0} u_{k+1/2}, \quad \widetilde{u}_{k+3/2} = \mathcal{P}_{k+1/2,+\Delta\varphi} u_{k+3/2}, \quad \widetilde{u}_{k+5/2} = \mathcal{P}_{k+1/2,+2\Delta\varphi} u_{k+5/2}$$
$$\widetilde{n}_k = \mathcal{P}_{k+1/2,-\Delta\varphi/2} n_k, \quad \widetilde{n}_{k+1} = \mathcal{P}_{k+1/2,+\Delta\varphi/2} n_{k+1} \tag{61}$$

Note that the pullback $\mathcal{P}_{k+1/2}$ is different from $\mathcal{P}_k$ in general. Only in the special case of an axisymmetric $\hat{\boldsymbol{b}} = \hat{\boldsymbol{b}}(R, Z)$ we have $\mathcal{P}_{k+1/2} \equiv \mathcal{P}_k$. We arrive at the velocity equation

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} u_{k+1/2} = &- \frac{b^{\varphi}_{k+1/2}}{\Delta\varphi} \left(\widetilde{f}_{k+1} - \widetilde{f}_k\right) - \frac{b^{\varphi}_{k+1/2}}{\Delta\varphi} \left[\left(\widetilde{n}_{k+1} - \widetilde{n}_k\right) \frac{1}{2} \left(\frac{1}{\widetilde{n}_{k+1}} + \frac{1}{\widetilde{n}_k} \right) \right] \\ &+ \nu_u \frac{2}{\widetilde{n}_{k+1} + \widetilde{n}_k} \frac{\widetilde{\Gamma}_{k+1/2,1/2} \widetilde{b}^{\varphi}_{k+1} \widetilde{b}^{\varphi}_{k+1} (\widetilde{u}_{k+3/2} - \widetilde{u}_{k+1/2}) - \widetilde{\Gamma}_{k+1/2,-1/2} \widetilde{b}^{\varphi}_{k} \widetilde{b}^{\varphi}_{k} (\widetilde{u}_{k+1/2} - \widetilde{u}_{k-1/2})}{\Delta\varphi^2} \end{aligned} \tag{62}$$

Note that the first term on the right hand side is a centred discretization of the parallel gradient similar to Eq. (21c). Similarly, the second term on the right hand side is a centred discretization for $(\nabla_\parallel n)/n$, while the last term represents $\nu_u \Delta_\parallel u/n$ according to (23). With $\widetilde{u}_k := 0.5(\widetilde{u}_{k+1/2} + \widetilde{u}_{k-1/2})$ the flux is discretized according to

$$\widetilde{f}_k := \frac{1}{2}\widetilde{u}_k \begin{cases} \widetilde{u}_{k-1/2} + \frac{1}{2}\Lambda\left(\Delta\widetilde{u}_{k+1/2}, \Delta\widetilde{u}_{k-1/2}\right) & \text{if } \widetilde{u}_k \geq 0 \\ \widetilde{u}_{k+1/2} - \frac{1}{2}\Lambda\left(\Delta\widetilde{u}_{k+3/2}, \Delta\widetilde{u}_{k+1/2}\right) & \text{if } \widetilde{u}_k < 0 \end{cases} \tag{63}$$

4.3. Flux-centred approach

A second possibility is to first centre the coordinate system on the fluxes and then in a second step centre on the value for the time-derivative. First, with the definitions (61) we compute the fluxes on the flux-centred grids:

$$\widetilde{q}_{k+1/2} := \widetilde{u}_{k+1/2} \begin{cases} \widetilde{n}_k + \frac{1}{2}\Lambda\left(\Delta\widetilde{n}_{k+1}, \Delta\widetilde{n}_k\right) & \text{if } \widetilde{u}_{k+1/2} \geq 0 \\ \widetilde{n}_{k+1} - \frac{1}{2}\Lambda\left(\Delta\widetilde{n}_{k+2}, \Delta\widetilde{n}_{k+1}\right) & \text{if } \widetilde{u}_{k+1/2} < 0 \end{cases} \tag{64}$$

$$\bar{f}_k := \frac{1}{2}\bar{u}_k \begin{cases} \bar{u}_{k-1/2} + \frac{1}{2}\Lambda\left(\Delta\bar{u}_{k+1/2}, \Delta\bar{u}_{k-1/2}\right) & \text{if } \bar{u}_k \geq 0 \\ \bar{u}_{k+1/2} - \frac{1}{2}\Lambda\left(\Delta\bar{u}_{k+3/2}, \Delta\bar{u}_{k+1/2}\right) & \text{if } \bar{u}_k < 0 \end{cases} \tag{65}$$

with

$$\bar{u}_k = \frac{1}{2}\left(\bar{u}_{k-1/2} + \bar{u}_{k+1/2}\right) \tag{66}$$

Then, in a second step we transform to the "value-centred" grids via

$$\bar{q}_{k+1/2} := \mathcal{P}_{k,+\Delta\varphi/2} \widetilde{q}_{k+1/2}, \quad \bar{q}_{k-1/2} := \mathcal{P}_{k,-\Delta\varphi/2} \widetilde{q}_{k-1/2}, \quad \widetilde{f}_{k+1} := \mathcal{P}_{k+1/2,+\Delta\varphi/2} \bar{f}_{k+1}, \quad \widetilde{f}_k := \mathcal{P}_{k+1/2,-\Delta\varphi/2} \bar{f}_k \tag{67}$$

We can then insert these fluxes back into Eqs. (59) and (62).

5. Numerical solutions of the three-dimensional Navier-Stokes equations

In this section we present an extensive numerical study of the novel finite volume flux coordinate independent approach (FV-FCI) proposed in this work. Our parallel implementation of the FV-FCI approach resides in the Feltor library as outlined in Appendix C. The study is somewhat complicated by the fact that there are many combinations that we could use to construct a final scheme. In Fig. 5 we graphically represent the decision flow. We here restrict ourselves to (i) using either the "staggered" (6) or "velocity-staggered" (10) scheme in 1d that we elevate to 3d using (ii) either the "value-centred" 4.2 or "flux-centred" 4.3 approach combined with (iii) either the dG (49) or linear finite element (51) projection operators for instances of $\mathcal{P}_k$ and $\mathcal{P}_{k+1/2}$. We already here discard the interpolation methods (43) and (44) as they are generally unstable in the tests that we performed. In the projection method we use $K = 12$ cells generally. We tested $K = 6$ as well and found no significant differences. As time integrator we use a Bogacki-Shampine explicit adaptive Runge-Kutta of third order [29].

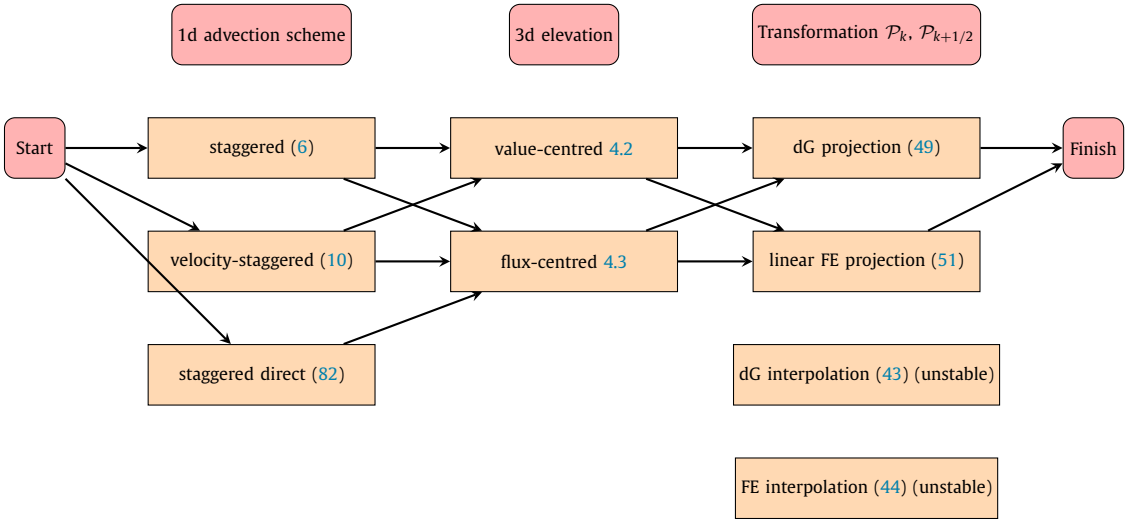

Fig. 5. Flow chart of the schemes constructed in this paper. Each path through the graph represents a scheme. The "staggered-direct" scheme is discussed later in the text.

5.1. Mass, momentum and energy conservation

Before we start to test our scheme we derive the invariants of the original equations (1) that we can use as a quality measure for our implementation. We define mass M, momentum P and energy E:

$$M(t) := \int n \mathrm{dV} \tag{68a}$$

$$P(t) := \int nu \mathrm{dV} \tag{68b}$$

$$E(t) := \int \left[n \ln n + \frac{1}{2} n u^2 \right] \mathrm{dV} \tag{68c}$$

where $\mathrm{dV} = R \mathrm{d}R \mathrm{d}Z \mathrm{d}\varphi$ and we integrate the entire simulation domain. We can then derive

$$\frac{\mathrm{d}}{\mathrm{d}t} M = 0 \tag{69a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} P = -\int \mathrm{dV}\, \nabla_\parallel n = \int \mathrm{dV}\, n \boldsymbol{\nabla} \cdot \hat{\boldsymbol{b}} \tag{69b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} E = -\int \mathrm{dV}\, \nu (\nabla_\parallel u)^2 \tag{69c}$$

The mass is invariant, as is energy for vanishing viscosity $\nu = 0$. The total momentum is not conserved unless the divergence of $\hat{\boldsymbol{b}}$ vanishes. If the equations are related to plasma physics, the term $n \boldsymbol{\nabla} \cdot \hat{\boldsymbol{b}}$ signifies the mirror force. We here disregard surface terms, which is possible if the field-lines do not intersect the domain boundary as is the case for our example field (53).

We can thus use M, P and E to define a measure of quality for our schemes

$$\varepsilon_M := \frac{M(t_0) - M(t_1)}{M(t_0)} \tag{70}$$

$$a_P := P(t_0) - P(t_1) - \int_{t_0}^{t_1} dt \int \mathrm{dV}\, \nabla_\parallel n \tag{71}$$

$$\varepsilon_E := \frac{1}{|E(t_0)|} \left[E(t_0) - E(t_1) - \int_{t_0}^{t_1} dt \int \mathrm{dV}\, \nu (\nabla_\parallel u)^2 \right] \tag{72}$$

In Eq. (72) we divide by the absolute of the initial energy density as $E(t_0)$ may be negative. Note that positive ε_M signify a decrease in total mass whereas negative ε_M signify an increase and analogous a_P and ε_E for momentum and energy.

5.2. Field aligned initialization

An important aspect of our simulations is a judicious initialization of the fields. We want structures to be field-aligned in the beginning of the simulation with a possible modulation along the direction of the field line. This means that in a field-aligned coordinate system our initial function reads

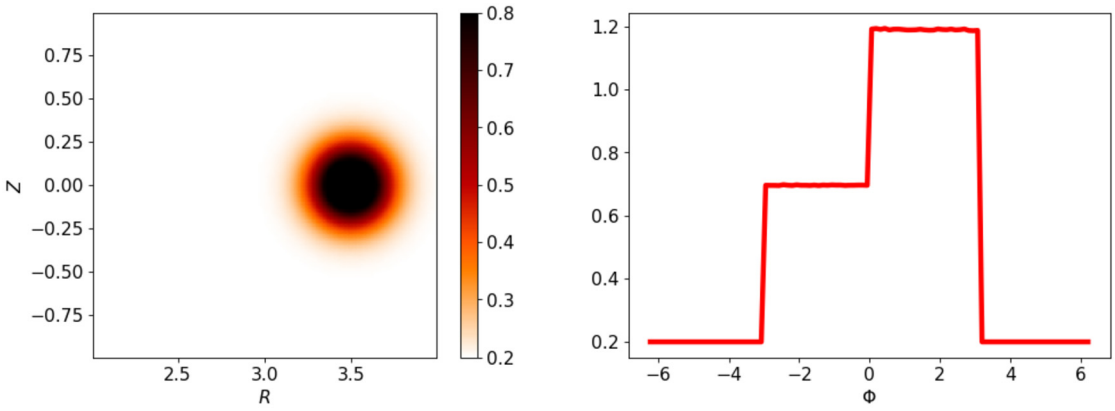

Fig. 6. Gaussian initial condition modulated with a double step function Eq. (75). Density n in the R-Z plane at $\varphi = 0$ (left) and along a single field-line starting at $(R, Z, \varphi) = (3.5, 0, 0)$ (right). The initial velocity u is zero.

$$F_0(\rho, \zeta, \Phi) = F_{\text{inv}}(\rho, \zeta) S(\Phi), \tag{73}$$

where $F_{\text{inv}}(\rho, \zeta)$ is a function that is invariant under the field line transformations. We call $S(\Phi)$ the modulation function along a field-line. The initial function in cylindrical coordinates can then be obtained using the push-forward

$$f_0(R, Z, \varphi) = F_0(\rho(R, Z, \varphi), \zeta(R, Z, \varphi), \Phi(\varphi)) \tag{74}$$

5.3. Purely advection scheme

We now test our proposed schemes on a Gaussian initial profile modulated with a double step function according to Eq. (73) with zero initial velocity u:

$$n(\rho, \zeta, \Phi, 0) = n_{\text{Gauss}}(\rho, \zeta) S_{\text{step}}(\Phi) \qquad u(\rho, \zeta, \Phi, 0) = 0 \tag{75}$$

with

$$n_{\text{Gauss}}(R, Z) := n_0 + A \exp\left(-\frac{(R - R_0)^2 + Z^2}{2\sigma^2}\right) \tag{76}$$

$$S_{\text{step}}(\Phi) := \begin{cases} 0.5 \text{ if } -\pi < \Phi < 0 \\ 1 \text{ if } 0 \leq \Phi < \pi \\ 0 \text{ else} \end{cases} \tag{77}$$

using $n_0 = 0.2$, $(R_0, Z_0) = (3.5, 0)$, $A = 1$ and $\sigma = 0.15$. This initial condition is shown in Fig. 6. On the left side we show the density in the R-Z plane at $\varphi = 0$ and on the right we show the density along a single field-line that passes through $(R, Z) = (3.5, 0)$ at $\varphi = 0$. There are slight fluctuations in the density along the field-line. These originate in interpolation errors as the field-line passes in between grid points on the various planes. The grid resolution is $96 \times 96 \times 50$ in (R, Z, φ).

In Fig. 7 we show the result of the "staggered" scheme (6) elevated with the "value-centred" approach 4.2 using the linear FE projection method Eq. (51) for the transformation operators. We do not use a slope-limiter. The result is shown at $T = 20$. In the top row we show the density, again in the R-Z plane on the left and along the field-line starting at $(R, Z, \varphi) = (3.5, 0, 0)$ on the right side. The bottom row shows the velocity in the same manner. As can be seen in the density on the top left plot and the velocity in the bottom left plot the initial condition traveled in the periodic φ direction and re-appears in the $\varphi = 0$ plane three times: two times in the positive Φ direction and one time in the negative direction. Since the magnetic field $\hat{\boldsymbol{b}}$ is sheared (see Fig. 3), the original round shape is lost in favour of "half-moon" shapes. These were also observed in studies of the heat conduction equation [16]. In the field-line plot of the density we see slight oscillations appear that may again be attributed to interpolation errors. Finally, in the bottom right plot we also show the relative mass, momentum and energy errors (70) and (72).

These same errors can be compared to Fig. 8, where we show the same plots as in Fig. 7 but for the "flux-centred" approach 4.3. The density and velocity plots are visually indistinguishable but the mass conservation is more than three orders of magnitudes better and the momentum conservation by a factor 10. This shows that the "flux-centred" approach is superior to the "value-centred" approach from the point of view of mass- and momentum-conservation. The energy error does not improve. In order to rule out errors from the time discretization we repeated our simulations with more timesteps but found no differences to the results presented here. This shows that energy is not as well conserved by the scheme as the mass and momentum are.

Finally, in Fig. 9 we show the same plots again; this time for the dG projection method (49), the "staggered" scheme (6) and the "flux-centred" approach 4.3. The 96 grid points in R and Z are now split into 32 cells with 3 polynomial coefficients each in each direction. The polynomial projection method is plagued by spurious oscillations in both the R-Z plane as well as along the magnetic field-line, especially in the density. Furthermore, the mass conservation is inferior to the linear FE projection method in Fig. 8. In addition, it is negative showing that there is a slight increase in mass compared to the initial mass.

This shows in fact the major issue of using the dG projection as proposed in Eq. (49). Oscillations pollute and destabilize the simulations. For example, the present simulation crashed at around $T = 23$ for the dG scheme, while it continues smoothly for the finite

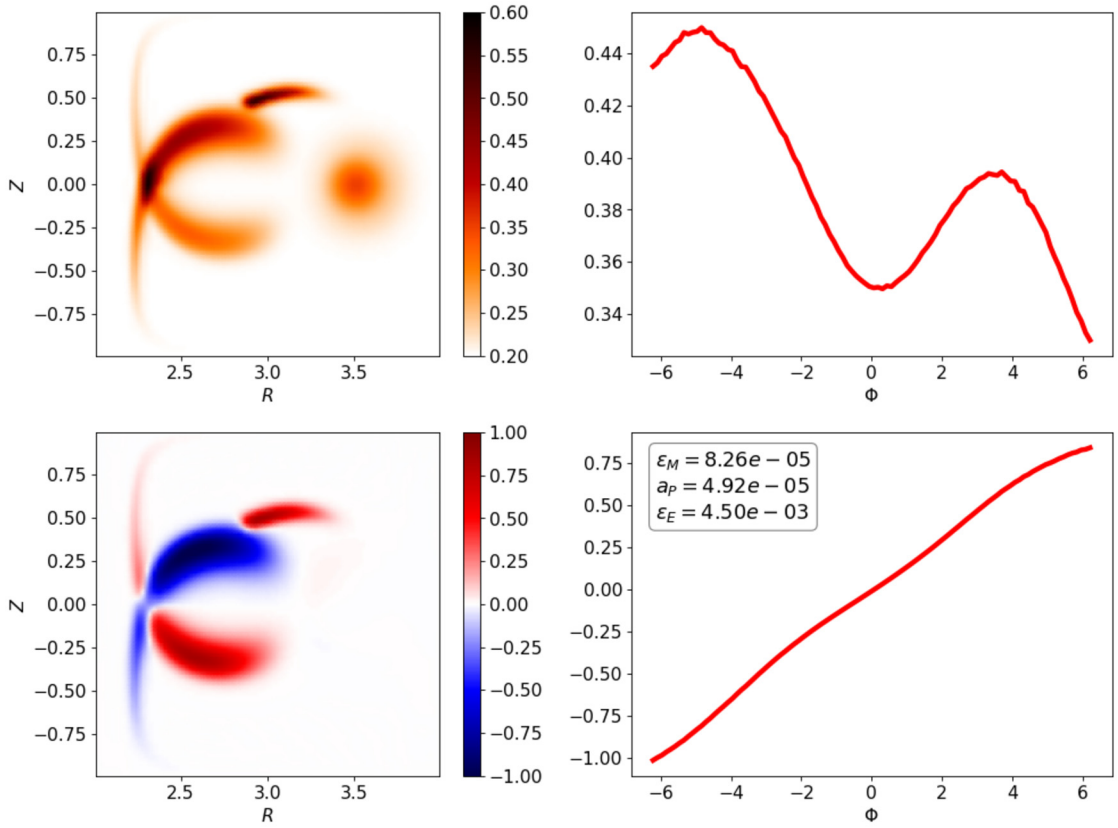

Fig. 7. Density n (top) and velocity u (bottom) in the R-Z plane at $\varphi = 0$ (left column) and along a single field-line starting at $(R, Z, \varphi) = (3.5, 0, 0)$ (right column). Gaussian initial condition (75), "staggered" scheme (6) elevated with the "value-centred" approach 4.2 using the linear FE projection method Eq. (51) with $K = 12$. Resolution $96 \times 96 \times 50$ grid points. $T = 20$, $\nu_u = 0$. The box in the bottom right plot shows the relative mass error ε_M (70), the absolute momentum error a_P (71) and the relative energy error ε_E (72).

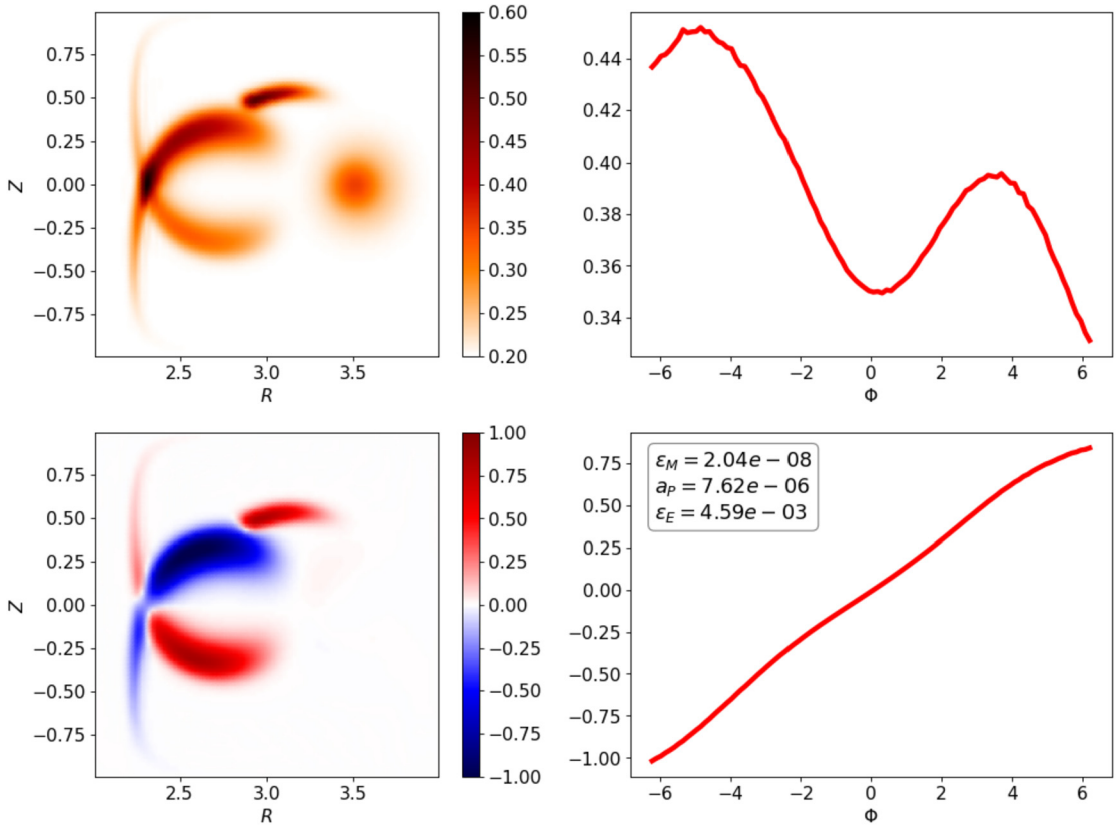

Fig. 8. Same as Fig. 7 but for "flux-centred" approach 4.3 instead of "value-centred".

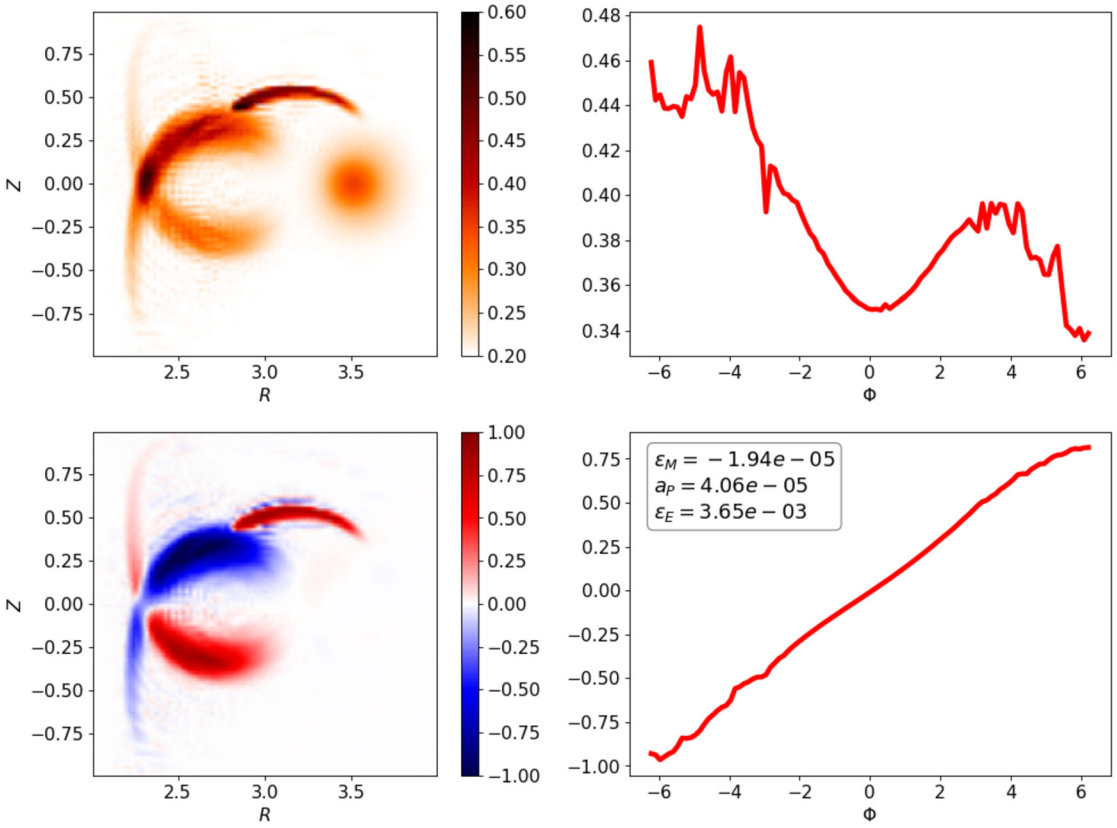

Fig. 9. Density n (top) and velocity u (bottom) in the R-Z plane at $\varphi = 0$ (left column) and along a single field-line starting at $(R, Z, \varphi) = (3.5, 0, 0)$ (right column). Gaussian initial condition (75), "staggered" scheme (6) elevated with the "flux-centred" approach 4.3 using the dG projection method Eq. (49) with $K = 12$. Resolution $96 \times 96 \times 50$ grid points (split into 32 cells with $P = 3$ in each cell in R and Z). $T = 20$, $\nu_u = 0$. The box in the bottom right plot shows the relative mass error ε_M (70), the absolute momentum error a_P (71) and the relative energy error ε_E (72). Oscillations appear.

element method (51). It should also be pointed out that the interpolation scheme (43) violently crashes at around $T = 3$. Oscillations were previously observed for the heat conduction equation using the dG interpolation in the parallel Laplacian [16].

However, it is too naive to conclude that the dG projection method is inferior in general to the linear FE projection method. In fact, the reason why the linear method performs better lies in neglecting the back-transformation S_{FE}^{-1} in Eq. (51). If we use the "correct" transformation in Eq. (50), the simulations crash just as the dG method does. The reason Eq. (51) is more stable is most easily seen in the operator S_{FE} itself, which can be interpreted as a convolution operator and thus a smoothing operator. It is crucial to realize that it is the smoothing introduced by S_{FE} that makes the linear finite element scheme stable.

We have investigated smoothers that integrate well with the dG projection method but found no method superior to directly using the linear FE projection method. We point out here that even if a dG discretization is given the linear projection method can still be used with a simple intermediate step. The dG polynomials can be interpolated to an equidistant grid after which the linear method can be applied. In the end the result can be inversely transformed back to a dG discretization.

We have further studied slope-limiters [24] and adaptive median filter techniques for removing impulse noise from corrupted images [34] but found that these methods either do not integrate well with an adaptive timestepper or overdampen the solution too much. For the rest of this numerical study we will thus always use the linear FE projection method with $K = 12$.

5.4. A shock in the perpendicular plane

The smoothing operator S_{FE} stabilizes the scheme against spurious oscillations but in some situations this can be insufficient as well. Consider now the "circle" initial condition modulated again with a double step function according to Eq. (73)

$$n(\rho, \zeta, \Phi, 0) = n_{\text{circle}}(\rho, \zeta) S_{\text{step}}(\Phi) \qquad u(\rho, \zeta, \Phi, 0) = 0 \tag{78}$$

with S_{step} given by Eq. (77) and

$$n_{\text{circle}}(R, Z) := n_0 + A \begin{cases} 1 \text{ if } (R - R_0)^2 - (Z - Z_0)^2 < \sigma^2 \\ 0 \text{ else} \end{cases} \tag{79}$$

using $n_0 = 0.2$, $(R_0, Z_0) = (3.5, 0)$, $A = 1$ and $\sigma = 0.1$. This initial condition is shown in Fig. 10 and introduces a step in the R-Z plane. We use the staggered scheme (6) in the "value-centred" approach 4.2 with the linear FE projection method (51). The result at $T = 30$ is shown in Fig. 11. As is visible the intitial step function travels twice in the periodic φ direction (i.e. 4π in both positive and negative φ direction). The initial circle is visible as four half moons in the density and velocity plots. Strong oscillations appear, in particular in the density. The oscillations do not disappear even for higher resolution.

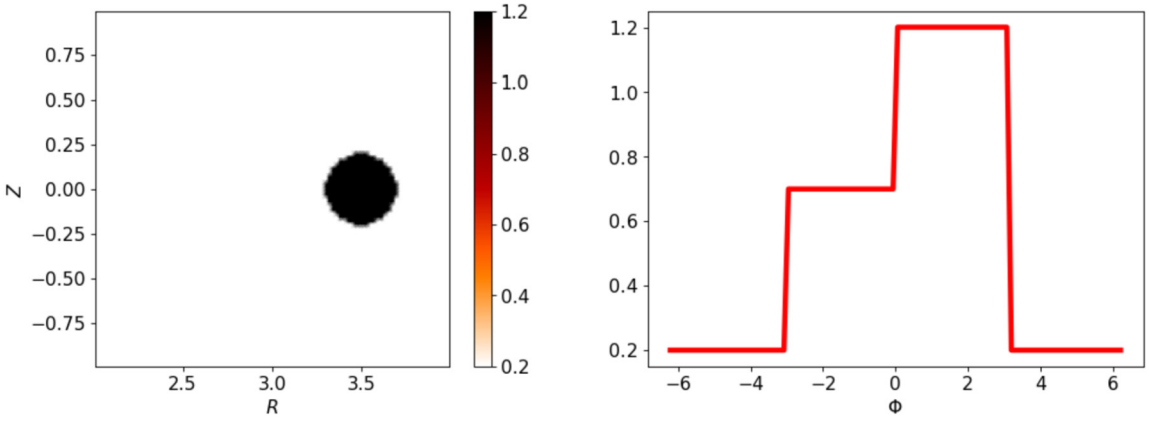

Fig. 10. Circle initial condition modulated with a double step function Eq. (78). Density n in the R-Z plane at $\varphi = 0$ (left) and along a single field-line starting at $(R, Z, \varphi) = (3.5, 0, 0)$ (right). The initial velocity u is zero.

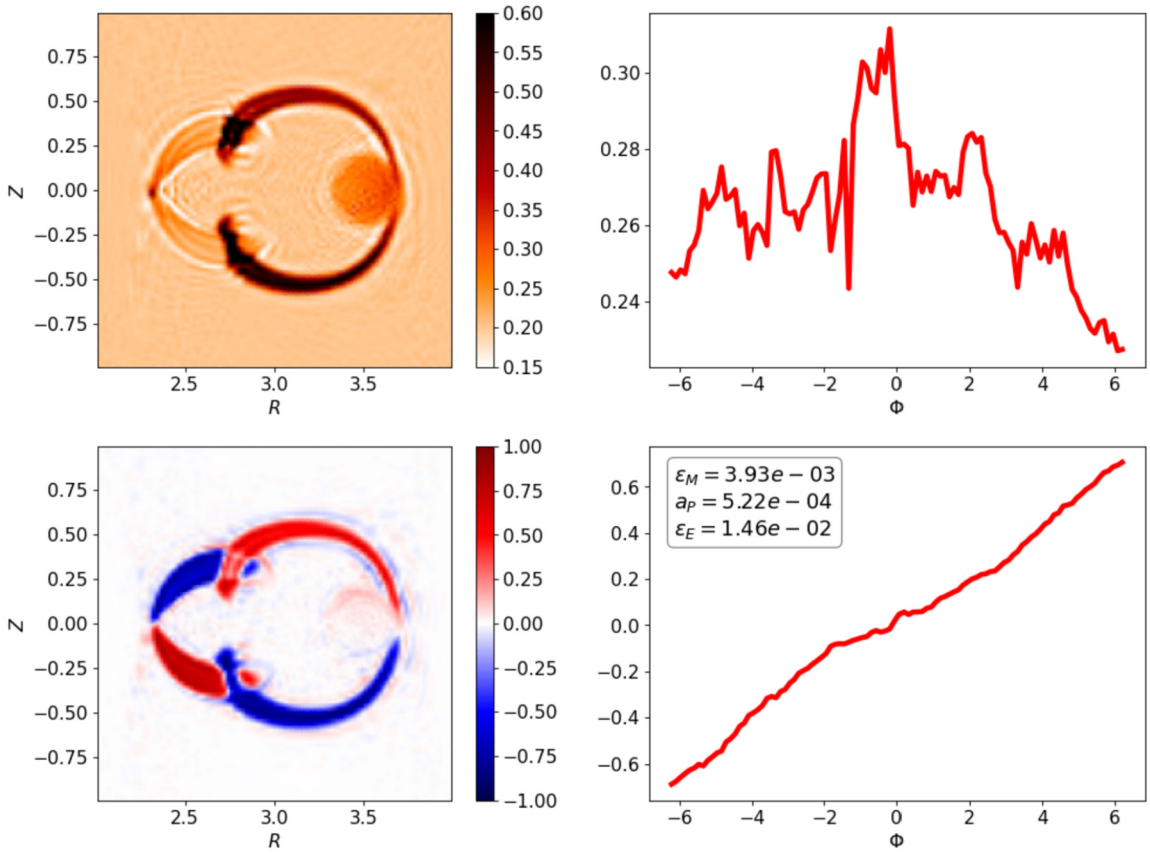

Fig. 11. Density n (top) and velocity u (bottom) in the R-Z plane at $\varphi = 0$ (left column) and along a single field-line starting at $(R, Z, \varphi) = (3.5, 0, 0)$ (right column). Circle initial condition (78), "staggered" scheme (6) elevated with the "value-centred" approach 4.2 using the linear FE projection method Eq. (51) with $K = 12$. Resolution $96 \times 96 \times 50$ grid points. $T = 30$, $\nu_u = 0$. The box in the bottom right plot shows the relative mass error ε_M (70), the absolute momentum error a_P (71) and the relative energy error ε_E (72). Oscillations appear especially along the field-lines in the density.

5.5. A problem in the parallel derivative

In order to elucidate the origin of these oscillations we now consider the parallel derivative $|\nabla_\parallel n|/n$ of the circle initial condition (78). Analytically, the parallel derivative of an exactly field-aligned function should be zero. However, numerically the interpolation at or very close to the step in the R-Z plane (i.e. $(R - R_0)^2 + (Z - Z_0)^2 \approx \sigma^2$) is necessarily incorrect, as the step cannot be resolved by either linear or dG polynomials. This can be seen for example for the dG projection method in Fig. 12 (left). Strong oscillations around the step appear likely due to the oscillatory nature of polynomials. However, these oscillations never vanish even for higher resolution or simple linear interpolation as seen in Fig. 12 (right). Since the density gradient directly appears in the momentum equation (1b) the oscillations will always be present in the simulations and the smoothing operator S_{FE} is not strong enough to dampen them.

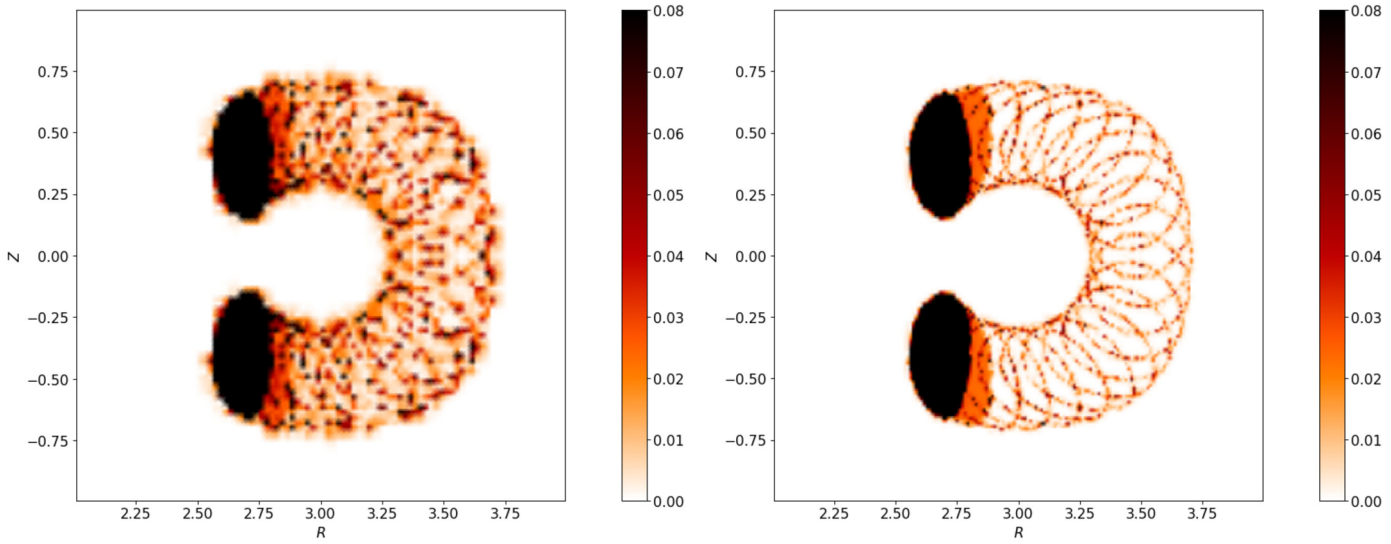

Fig. 12. The toroidally averaged parallel derivative of the circle initial condition (78) using the dG projection method Eq. (49) with $N_R = N_Z = 96$, $N_\varphi = 20$ with $P = 3$ polynomial coefficients and $K = 12$ (left) and a linear interpolation Eq. (44) with $N_R = N_Z = 180$, $N_\varphi = 20$ (right). Analytically, the result should be zero except at the two black ovals. Strong oscillations originating from the interpolation scheme are visible.

5.6. Advection-diffusion system

Any sharp enough gradient in the simulation domain may lead to oscillations and/or a crash of the simulation. A possible remedy is to introduce a perpendicular diffusion term of the form $\nu_{\perp,n}\Delta_{RZ}n$ into the density equation (1a) and analogously $\nu_{\perp,u}\Delta_{RZ}u$ into the momentum equation (1b). Here we denote $\Delta_{RZ}n := \partial_R(R\partial_R n)/R + \partial_Z^2 n$. Our tests show that this form of stabilization does indeed dampen oscillations. The issue is however that typically the R-Z plane is much higher resolved than the φ direction. This after all was one of the motivations to use the FCI scheme in the first place. The amount of diffusion needed is typically so high that the CFL condition severely suffers from adding the term at least in an explicit timestepper. Even in an implicit timestepper, the high resolution in R and Z together with a large diffusion coefficient can lead to an unnecessarily high condition number of the implicit equation.

This is why we here prefer the second method, which is to use parallel diffusion to stabilize the scheme. We already have the viscosity $\nu_u\Delta_\parallel u$ as part of Eq. (1b). It turns out, however, that density diffusion is equally needed to ensure stability. We therefore add a parallel diffusion operator $\nu_n\Delta_\parallel n$ to the density Eq. (1a). It is important for energetic consistency that diffusion in the density equation entails a corresponding term in the momentum equation in order to avoid unphysical generation of energy. We follow Reference [35] and propose

$$\frac{\partial}{\partial t}n + \boldsymbol{\nabla}\cdot\left(nu\hat{\boldsymbol{b}}\right) = \nu_n\Delta_\parallel n \tag{80a}$$

$$\frac{\partial}{\partial t}(nu) + \boldsymbol{\nabla}\cdot\left(nu^2\hat{\boldsymbol{b}}\right) = -\nabla_\parallel n + \nu_u\Delta_\parallel u + \nu_n\boldsymbol{\nabla}\cdot(u\hat{\boldsymbol{b}}\nabla_\parallel n) \tag{80b}$$

The additional term proportional to ν_n in the momentum equation (80b) is simply the advection of velocity by the density diffusion current $\boldsymbol{j}_n = -\nu_n\nabla_\parallel n\hat{\boldsymbol{b}}$. Numerically, we therefore propose an upwind scheme with $\boldsymbol{j}_n$ as the "velocity" as a discretization. The velocity formulation then reads

$$\frac{\partial}{\partial t}u + u\nabla_\parallel u = -\nabla_\parallel n + \nu_u\frac{\Delta_\parallel u}{n} + \nu_n\frac{\nabla_\parallel n}{n}\nabla_\parallel u$$

Eqs. (80) do not change the mass and momentum conservation from Eqs. (69a) and (69b) but change the energy conservation (69c) to

$$\frac{\mathrm{d}}{\mathrm{d}t}E = -\int \nu_u(\nabla_\parallel u)^2 + \nu_n\frac{(\nabla_\parallel n)^2}{n}\mathrm{dV} \tag{81}$$

Both viscosity and density diffusion terms are quadratic and therefore dissipate energy.

Since we now have parallel diffusion we can use the "velocity-staggered" scheme (10) to discretize Eqs. (80) in velocity formulation. We use $\nu_n = \nu_u = 0.2$. In Fig. 13 we show the result for the circle initial condition (78), the "flux-centred" approach 4.3 and the linear FE projection method (51). We find that the oscillations are absent from the density and velocity plots and that the mass conservation is still at a high level.

In order to compare the conservation properties of the various scheme combinations presented in this paper we now plot the mass, momentum and energy conservation as a function of time. To compare to existing literature we implemented the finite difference approach proposed in Reference [4]. Here, the logarithm of the density is integrated. In 1d the scheme reads

$$q_{k+1/2} := \frac{1}{\Delta x}u_{k+1/2}((\ln n)_{k+1} - (\ln n)_k), \qquad u_k := \frac{1}{2}(u_{k+1/2} + u_{k-1/2})$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(\ln n)_k = -\frac{1}{\Delta x}(u_{k+1/2} - u_{k-1/2}) - \frac{1}{2}\left(q_{k+1/2} + q_{k-1/2}\right) + \nu_n\frac{1}{n_k\Delta x^2}\left(n_{k+1} - 2n_k + n_{k-1}\right) \tag{82a}$$

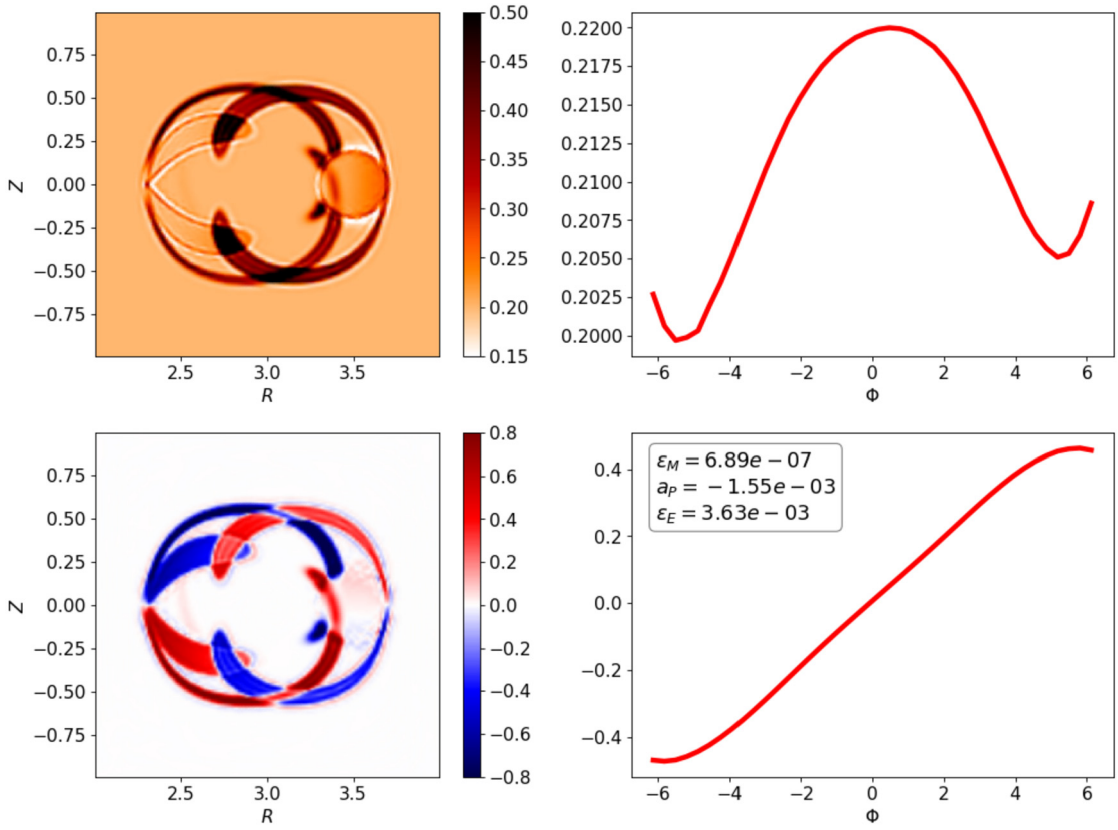

Fig. 13. Simulation of Eqs. (80) in velocity formulation with diffusion coefficients $\nu_u = \nu_n = 0.2$. Density n (top) and velocity u (bottom) in the R-Z plane at $\varphi = 0$ (left column) and along a single field-line starting at $(R, Z, \varphi) = (3.5, 0, 0)$ (right column). Circle initial condition (78), "velocity-staggered" scheme (10) elevated with the "flux-centred" approach 4.3 using the linear FE projection method Eq. (51) with $K = 12$. Resolution $150 \times 150 \times 20$ grid points. $T = 40$. The box in the bottom right plot shows the relative mass error ε_M (70), the absolute momentum error a_P (71) and the relative energy error ε_E (72).

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{k+1/2} = -\frac{1}{\Delta x}u_{k+1/2}(u_{k+1} - u_k) - \frac{1}{\Delta x}((\ln n)_{k+1} - (\ln n)_k) + \nu_u \frac{1}{\Delta x^2}\left(u_{k+3/2} - 2u_{k+1/2} + u_{k-1/2}\right) \tag{82b}$$

We call this scheme the "staggered-direct" scheme. The elevation to 3d uses the "flux-centred" approach in the sense that we directly use Eqs. (59) (applied to $\ln n$) and (61) to compute the "fluxes" $q_{k+1/2}$ and u_k and in a second step transform to the value-centred grids as in Eq. (67). As this is not a finite volume scheme $q_{k+1/2}$ and u_k are not technically fluxes, however. The scheme is unstable without diffusion.

We plot the mass (left), momentum (middle) and energy (right) conservation as a function of time for the "velocity-staggered" scheme in both "value-centred" and "flux-centred" variants and compare the "staggered-direct" scheme in Fig. 14. In the left panel we can confirm again that the "flux-centred" transformation has the best mass conservative properties and is four to five orders of magnitude better than the "staggered-direct" scheme. The "value-centred" approach is about a factor 10 better than the "staggered-direct" scheme. It should also be noted that values for ε_M for the "staggered-direct" scheme are negative. This means that the scheme gains mass, while all other schemes dissipate mass. The non-conservative and mass-generating properties of the "staggered-direct" scheme can also be reproduced in one-dimensional tests analogous to Fig. 1 and 2 (shown in the accompanying dataset). Finally, we also used the discontinuous Galerkin projection method in combination with the staggered and "flux-centred" approaches. It turns out that the parallel diffusion is sufficient to dampen oscillations in the resulting density and velocity fields (not shown). Judging from Fig. 14 (left) the mass conservation of the dG projection is approximately a factor 10 worse than using a linear finite element projection. This is somewhat surprising since Table 2 would suggest best volume conservation for the dG method.

In the middle panel of Fig. 14 we show the corresponding absolute momentum errors a_P (71) and in the right panel the relative energy errors ε_E (72). The staggered flux-centred dG scheme has the lowest momentum errors by about 2 orders of magnitude towards the staggered-direct scheme with the highest errors. The velocity-staggered schemes as they are not formulated in momentum form have about a factor 10 higher error than the staggered scheme. For the energy conservation on the right panel all FV-FCI schemes have about the same error while again the staggered-direct scheme yields the highest errors with a negative sign.

6. Conclusions

In conclusion, we show in this paper how the FCI scheme can be combined with finite volume advection schemes in 1d to form a new class of schemes we call finite-volume FCI (FV-FCI). This new family of schemes is visualized in Fig. 5. Our method bases on an interpretation of FCI as a finite difference discretization within a locally field-aligned coordinate system. We re-derive previously suggested expressions for the parallel derivative and suggest new expressions for the divergence and the parallel Laplacian in Eqs. (22) and (23). This requires the additional integration of the volume element as demonstrated in Eq. (17c).

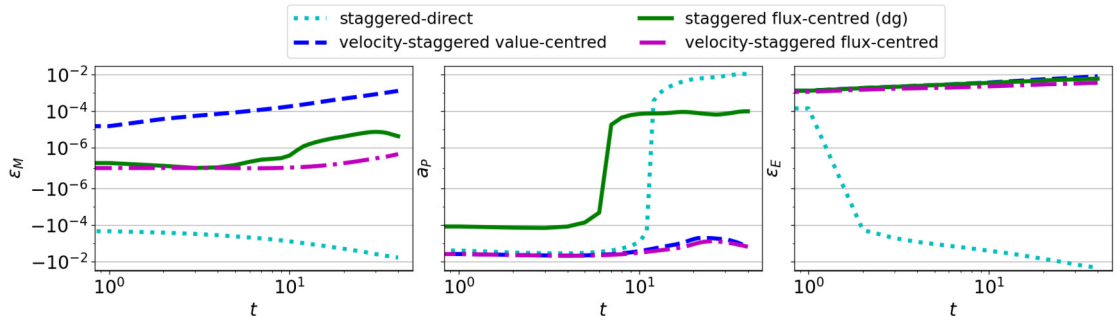

Fig. 14. The relative mass error ε_M (left), absolute momentum error a_P (middle) and relative energy error ε_E (right) of four schemes simulating Eqs. (80) with $\nu_n = \nu_u = 0.2$ dependent on time. Resolution $150 \times 150 \times 20$ grid points. All schemes except the staggered flux-centred scheme use the linear FE projection. The staggered-direct scheme is given in Eqs. (82). The staggered scheme is combined with a dG projection method.

The core of the FCI method is in our interpretation the transformation from locally non-field-aligned to field-aligned coordinates. We study various methods starting with simple polynomial interpolation to a projection on dG polynomials or finite elements. We explicitly show convergence of second order for all of the suggested methods.

The FV-FCI method is a result of elevating a one-dimensional advection scheme to three dimensions using either a "value-centred" or "flux-centred" approach and formulating the transformation operators using either interpolation or projection (cf. Fig. 5). We show this approach explicitly at the example of a staggered finite volume scheme in two variations. Our tests using a field-aligned Navier-Stokes equation along a given magnetic field show that the "flux-centred" approach presented in Section 4.3 has better mass- and momentum-conservation than the "value-centred" approach 4.2. The projection method for the coordinate transformation in Section 3.3.3 is found superior to the interpolation method 3.3.2. Projection onto triangular finite elements entails a smoothing kernel that stabilizes simulations against spurious oscillations. However, if perpendicular gradients become too large, these spurious oscillations, caused by the coordinate transformations, destabilize and crash simulations. In this case parallel diffusion must be added for stability.

In conclusion, the best performing scheme is the "staggered" finite volume scheme (6) elevated using the "flux-centred" method 4.3 in combination with projection onto linear finite elements (51). If strong perpendicular gradients are an issue in the simulation domain, it is preferable to start from the "velocity-staggered" (10) scheme instead, adding parallel numerical diffusion to both density and velocity equation.

An open question remains whether our finite volume FCI scheme can be made resilient to shocks in the perpendicular planes without resorting to artificial diffusion. Possible candidates could include an appropriately constructed perpendicular flux-limiter, however, this may be difficult since oscillations near sharp gradients do not vanish even for low order discretizations. Another avenue for future work is the extension to a discontinuous Galerkin method in the parallel direction, resulting in a dG-FCI scheme. With the methods presented in this paper such an endeavour should be feasible in the near future.

In all discussions of the scheme in this paper we avoid boundary conditions by aligning the simulation boundary box with the magnetic field. The question remains what to do in a situation where a field-line intersects with the boundary of the simulation domain. Various suggestions exist [16,32,4]. However, the main trouble with existing approaches is that they may introduce numerical instability in the scheme and/or introduce boundary layers that are hard to resolve. A dedicated investigation on this topic is postponed to the future.

Finally, we mention an idea that stems from the observation that the parallel derivative $\nabla_\parallel$ is the adjoint of the divergence $\boldsymbol{\nabla} \cdot \hat{\boldsymbol{b}}\cdot$ in the L_2 norm (at least if surface terms vanish). The hypothesis is that this relation can be realized also numerically, i.e. by adjoining the discretization for $\nabla_\parallel$ we may arrive at a discretization for $\boldsymbol{\nabla} \cdot \hat{\boldsymbol{b}}\cdot$ analogous to [36] (where it is called the "support operator method"). There is empirical evidence that the numerically adjoint discretization may or may not converge with grid refinement [16,32]. This calls for a more formal theoretical analysis with a possible proof of convergence in the future.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Matthias Wiesenberger reports financial support was provided by European Consortium for the Development of Fusion Energy. Markus Held reports financial support was provided by Austrian Science Fund.

Data availability

The data is available as described in Appendix C.

Acknowledgements

We acknowledge fruitful discussions on explicit staggered schemes with Raphaèle Herbin and Jean-Claude Latché. Further, we thank A. Stegmeir and P. Tamain for helpful comments on the manuscript. This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No. 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them. This work was supported by the UiT Aurora Centre Program, UiT The Arctic University of Norway (2020). This research was funded in whole or in part by the Austrian Science Fund (FWF) [P 34241-N]. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission.

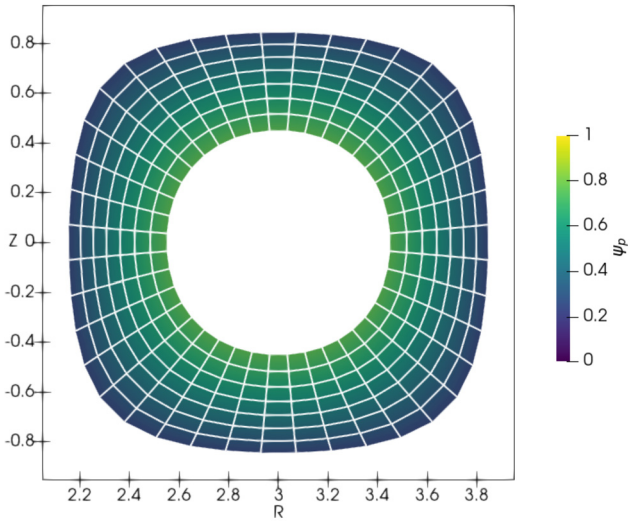

Fig. A.15. A curvilinear grid ξ, η with resolution 8×40 between $\psi_p = 0.2$ and 0.8. Compare to Fig. 4.

Appendix A. Curvilinear base coordinates

In the main text we start to develop our scheme from cylindrical coordinates R, Z, φ. There is nothing special about these coordinates, in fact the entire procedure holds for any coordinate system η, ξ, φ as long as one coordinate called φ exists such that $b^\varphi \neq 0$. The only issue that may arise is if $\hat{\boldsymbol{b}}$ is given analytically in a specific coordinate system (for example cylindrical). Then the question is how to integrate the field lines in the η, ξ, φ system. There are two possibilities. First, interpolate $R(\xi_i, \eta_i)$, $Z(\xi_i, \eta_i)$ for all i, then integrate $\hat{\boldsymbol{b}}$ in (R, Z) space and finally use Newton iteration to find $\xi(R_i^\pm, Z_i^\pm)$, $\eta(R_i^\pm, Z_i^\pm)$.

The second possibility (the one we implemented) is to integrate entirely in the transformed coordinate system ξ, η, φ. The magnetic field can be easily transformed with the Jacobian of the coordinate transformation (and assuming in this example we do not transform the φ coordinate)

$$b^\xi(\xi,\eta) = \left(\frac{\partial \xi}{\partial R} b^R + \frac{\partial \xi}{\partial Z} b^Z\right)_{R(\xi,\eta),Z(\xi,\eta)} \tag{A.1a}$$

$$b^\eta(\xi,\eta) = \left(\frac{\partial \eta}{\partial R} b^R + \frac{\partial \eta}{\partial Z} b^Z\right)_{R(\xi,\eta),Z(\xi,\eta)} \tag{A.1b}$$

$$b^\varphi(\xi,\eta) = b^\varphi(R(\xi,\eta), Z(\xi,\eta)) \tag{A.1c}$$

The field-line equations (17) are still

$$\frac{\mathrm{d}\xi}{\mathrm{d}\varphi} = \frac{b^\xi}{b^\varphi} \tag{A.2a}$$

$$\frac{\mathrm{d}\eta}{\mathrm{d}\varphi} = \frac{b^\eta}{b^\varphi} \tag{A.2b}$$

$$\frac{\mathrm{d}\sqrt{G}}{\mathrm{d}\varphi} = \nabla \cdot \left(\frac{\hat{\boldsymbol{b}}}{b^\varphi}\right)\sqrt{G} \tag{A.2c}$$

i.e. the original R and Z are simply replaced by ξ and η and the initial value for $\sqrt{G}$ is now the volume form of the ξ, η, φ coordinate system instead of the cylindrical coordinate system.

The issue here is that when integrating field-lines in a numerically given coordinate system we have to interpolate the vector field $\hat{\boldsymbol{b}}$ at arbitrary points instead of simply evaluating the exact values analytically. As a remark this could of course also be the case if $\hat{\boldsymbol{b}}$ is only available at discrete points in the first place. When using a dG discretization, the interpolation error vanishes with order P in the perpendicular plane. Apart from the issue of how to integrate the transformed vector field the remaining algorithms remain entirely unchanged. In Fig. A.15 we plot an example curvilinear grid using Eq. (53) between $\psi_0 = 0.2$ and $\psi_1 = 0.8$.

The advantage of the particular grid in Fig. A.15 is that it is flux aligned, i.e. the magnetic field in ξ, η, φ coordinates has no component in ξ. This means that the interpolation and projection operators are truly one dimensional operations in η, which may reduce the error. This can be seen in the convergence Table A.4, where we use the same test-functions as in Section 3.5.

Appendix B. The complete metric tensor in the field-aligned coordinate system

In this section we derive all Jacobian elements for the coordinate transformation from R, Z, φ to the locally field-aligned ρ, ζ, Φ. These can be used to transform any tensor (including the metric tensor) into the locally field-aligned coordinate system. We have the contravariant basis vectors

Table A.4
Convergence Table for $K = 100$ and $N_\xi = 24$, using the linear FE projection method (51).

		$\nabla_\parallel^{CC}$ Eq. (21c)		$\nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$ Eq. (22c)		$\Delta_\parallel^{CC}$ Eq. (23)		$\int dV \nabla\cdot(\hat{\boldsymbol{b}}f)^{CC}$		$\int dV \Delta_\parallel^{CC} f$	
		error	order	error	order	error	order	error	order	error	order
N_η	N_φ										
10	**5**	8.05e-01	n/a	8.10e-01	n/a	5.03e-01	n/a	-4.65e-03	n/a	8.65e-05	n/a
20	**10**	2.79e-01	1.53	2.84e-01	1.51	1.61e-01	1.64	-6.53e-04	2.83	9.41e-08	9.84
40	**20**	7.63e-02	1.87	7.79e-02	1.86	4.32e-02	1.90	-8.41e-05	2.96	1.39e-08	2.76
80	**40**	1.96e-02	1.96	2.01e-02	1.96	1.11e-02	1.96	-1.06e-05	2.99	1.43e-07	-3.36
160	**80**	5.15e-03	1.93	5.27e-03	1.93	3.12e-03	1.84	-1.29e-06	3.03	1.76e-07	-0.30

$$\partial_\rho = R_\rho \partial_R + Z_\rho \partial_Z \tag{B.1a}$$
$$\partial_\zeta = R_\zeta \partial_R + Z_\zeta \partial_Z \tag{B.1b}$$
$$\partial_\Phi = R_\Phi \partial_R + Z_\Phi \partial_Z + \varphi_\Phi \partial_\varphi \tag{B.1c}$$

By definition (of ∂_Φ) we have

$$R_\Phi = \frac{b^R}{b^\varphi}, \quad Z_\Phi = \frac{b^Z}{b^\varphi}, \quad \varphi_\Phi = 1 \tag{B.2}$$

We have the covariant base vectors

$$\mathrm{d}\rho = \rho_R \mathrm{d}R + \rho_Z \mathrm{d}Z + \rho_\varphi \mathrm{d}\varphi \tag{B.3a}$$
$$\mathrm{d}\zeta = \zeta_R \mathrm{d}R + \zeta_Z \mathrm{d}Z + \zeta_\varphi \mathrm{d}\varphi \tag{B.3b}$$
$$\mathrm{d}\Phi = \mathrm{d}\varphi \tag{B.3c}$$

The last identity expresses that equidistant planes in φ and Φ are the same. Note that the elements of the forward and backward transformation are related through the Jacobian

$$\begin{pmatrix} R_\rho & R_\zeta & R_\Phi \\ Z_\rho & Z_\zeta & Z_\Phi \\ \varphi_\rho & \varphi_\zeta & \varphi_\Phi \end{pmatrix} = \begin{pmatrix} \rho_R & \rho_Z & \rho_\varphi \\ \zeta_R & \zeta_Z & \zeta_\varphi \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \zeta_Z & -\rho_Z & (\rho_Z\zeta_\varphi - \zeta_Z\rho_\varphi) \\ -\zeta_R & \rho_R & (\rho_\varphi\zeta_R - \zeta_\varphi\rho_R) \\ 0 & 0 & (\rho_R\zeta_Z - \zeta_R\rho_Z) \end{pmatrix} \frac{1}{(\rho_R\zeta_Z - \zeta_R\rho_Z)} \tag{B.4}$$

We immediately see that the determinant of the Jacobian (relevant for the volume form) is given by $\det J = (R_\rho Z_\zeta - Z_\rho R_\zeta) = 1/(\rho_R\zeta_Z - \zeta_R\rho_Z)$. In order to integrate the elements of the Jacobian we employ Nemov's algorithm [37]. Notice first that $\mathrm{d}\rho/\mathrm{d}\Phi = \mathrm{d}\zeta/\mathrm{d}\Phi = 0$. In words, if we go along a streamline of ∂_Φ (a field-line) we stay on the same value of ρ (this is basically the definition of a coordinate line). We can write $\mathrm{d}/\mathrm{d}\Phi = (\hat{\boldsymbol{b}}/b^\varphi)\cdot\nabla$ and thus $\hat{\boldsymbol{b}}\cdot\nabla\rho = 0$. From this we immediately obtain

$$\rho_\varphi = -\frac{b^R}{b^\varphi}\rho_R - \frac{b^Z}{b^\varphi}\rho_Z, \qquad \zeta_\varphi = -\frac{b^R}{b^\varphi}\zeta_R - \frac{b^Z}{b^\varphi}\zeta_Z \tag{B.5}$$

If we derive $\mathrm{d}\rho/\mathrm{d}\Phi = 0$ with respect to R, we get

$$\partial_R((\hat{\boldsymbol{b}}/b^\varphi)\cdot\nabla\rho) = \partial_R(\hat{\boldsymbol{b}}/b^\varphi)\cdot\nabla\rho/b^\varphi + (\hat{\boldsymbol{b}}/b^\varphi)\cdot\nabla\rho_R = 0$$

from which we get $\mathrm{d}\rho_R/\mathrm{d}\Phi = -\partial_R(\hat{\boldsymbol{b}}/b^\varphi)\cdot\nabla\rho$ and analogous for the other elements, ρ_Z, ζ_R and ζ_Z. In total we find

$$\frac{\mathrm{d}R}{\mathrm{d}\Phi} = \frac{b^R}{b^\varphi}, \qquad \frac{\mathrm{d}Z}{\mathrm{d}\Phi} = \frac{b^Z}{b^\varphi}, \tag{B.6a}$$
$$\frac{\mathrm{d}\rho_R}{\mathrm{d}\Phi} = -\left(\frac{b^R}{b^\varphi}\right)_R \rho_R - \left(\frac{b^Z}{b^\varphi}\right)_R \rho_Z, \qquad \frac{\mathrm{d}\rho_Z}{\mathrm{d}\Phi} = -\left(\frac{b^R}{b^\varphi}\right)_Z \rho_R - \left(\frac{b^Z}{b^\varphi}\right)_Z \rho_Z, \tag{B.6b}$$
$$\frac{\mathrm{d}\zeta_R}{\mathrm{d}\Phi} = -\left(\frac{b^R}{b^\varphi}\right)_R \zeta_R - \left(\frac{b^Z}{b^\varphi}\right)_R \zeta_Z, \qquad \frac{\mathrm{d}\zeta_Z}{\mathrm{d}\Phi} = -\left(\frac{b^R}{b^\varphi}\right)_Z \zeta_R - \left(\frac{b^Z}{b^\varphi}\right)_Z \zeta_Z \tag{B.6c}$$

As initial conditions we use that the coordinate system is supposed to coincide with R, Z at the origin, that is $\rho_R = 1$, $\rho_Z = 0$, $\zeta_R = 0$, $\zeta_Z = 1$. Together with Eq. (B.5) this completely determines all elements in the Jacobian. We can derive further

$$\frac{\mathrm{d}(\rho_R\zeta_Z - \zeta_R\rho_Z)}{\mathrm{d}\Phi} = -\left(\left(\frac{b^R}{b^\varphi}\right)_R + \left(\frac{b^Z}{b^\varphi}\right)_Z\right)(\rho_R\zeta_Z - \rho_Z\zeta_R)$$
$$\frac{\mathrm{d}(R_\rho Z_\zeta - R_\zeta Z_\rho)}{\mathrm{d}\Phi} = \left(\left(\frac{b^R}{b^\varphi}\right)_R + \left(\frac{b^Z}{b^\varphi}\right)_Z\right)(R_\rho Z_\zeta - Z_\rho R_\zeta)$$

The last identity is particularly interesting if we are only interested in the volume element in the new coordinate system $\sqrt{G} = \det J\sqrt{g}$. In order to obtain the volume element directly (without integrating all the elements of the Jacobian separately) we compute

$$\frac{\mathrm{d}\sqrt{G}}{\mathrm{d}\Phi} = \frac{\mathrm{d}\left(\sqrt{g}\det J\right)}{\mathrm{d}\Phi} = \left[\frac{b^R}{b^\varphi}\sqrt{g}_R + \frac{b^Z}{b^\varphi}\sqrt{g}_Z + \left(\frac{b^R}{b^\varphi}\right)_R + \left(\frac{b^Z}{b^\varphi}\right)_Z\right]\det J$$
$$= \left[\partial_R\left(\sqrt{g}\frac{b^R}{b^\varphi}\right) + \partial_Z\left(\sqrt{g}\frac{b^Z}{b^\varphi}\right)\right]\det J = \nabla\cdot\left(\frac{\hat{\boldsymbol{b}}}{b^\varphi}\right)\sqrt{g}\det J = \nabla\cdot\left(\frac{\hat{\boldsymbol{b}}}{b^\varphi}\right)\sqrt{G} \tag{B.7}$$

which confirms Eq. (17c). The metric in the transformed coordinates is given by $G_{ij} = J^k_i J^l_j g_{kl}$. With this, all metric dependent operations like general divergences, gradients and cross-products are completely determined in the transformed coordinates.

Appendix C. Data access

The implementation of the FV-FCI approach as well as the one-dimensional schemes is available freely inside the Feltor library [26,27], a C++ code project developed on GitHub. Ample documentation is available on https://feltor-dev.github.io. As Feltor is a GPU native library there is naturally a GPU implementation of the FV-FCI scheme available. In addition, a CPU-only as well as OpenMP shared memory version is implemented. Finally, through an MPI+X (with X ∈ [CPU,GPU,OpenMP]) approach separating communication from computation we manage to run the FV-FCI scheme on high performance compute clusters with MPI parallelization possible in all three dimensions.

Feltor interoperates with our simple Python simulation database manager simplesimdb a Python package available at https://pypi.org/project/simplesimdb/. This enables the data analysis in Jupyter Notebooks in https://github.com/mwiesenberger/advection and https://github.com/mwiesenberger/convergence_ds. With these Notebooks, the Feltor library and simplesimdb the results in the entire paper can be reproduced.

References

[1] M. Wiesenberger, Gyrofluid computations of filament dynamics in tokamak scrape-off layers, Ph.D. thesis, University of Innsbruck, 2014, http://resolver.obvsg.at/urn:nbn:at:at-ubi:1-1799.
[2] M. Held, Full-f gyro-fluid modelling of the tokamak edge and scrape-off layer, Ph.D. thesis, University of Innsbruck, 2016, http://resolver.obvsg.at/urn:nbn:at:at-ubi:1-6853.
[3] F. Hariri, P. Hill, M. Ottaviani, Y. Sarazin, Plasma Phys. Control. Fusion 57 (5) (2015) 054001, https://doi.org/10.1088/0741-3335/57/5/054001.
[4] A. Stegmeir, D. Coster, A. Ross, O. Maj, K. Lackner, E. Poli, Plasma Phys. Control. Fusion 60 (3) (2018) 035005, https://doi.org/10.1088/1361-6587/aaa373.
[5] W. Zholobenko, A. Stegmeir, T. Body, A. Ross, P. Manz, O. Maj, D. Coster, F. Jenko, M. Francisquez, B. Zhu, B. Rogers, Contrib. Plasma Phys. 60 (5–6) (2020) e201900131, https://doi.org/10.1002/ctpp.201900131, https://onlinelibrary.wiley.com/doi/abs/10.1002/ctpp.201900131.
[6] D. Galassi, C. Theiler, T. Body, F. Manke, P. Micheletti, J. Omotani, M. Wiesenberger, M. Baquero-Ruiz, I. Furno, M. Giacomin, E. Laribi, F. Militello, P. Ricci, A. Stegmeir, P. Tamain, H. Bufferand, G. Ciraolo, H. De Oliveira, A. Fasoli, V. Naulin, S.L. Newton, N. Offeddu, D.S. Oliveira, E. Serre, N. Vianello, Phys. Plasmas 29 (1) (2022) 012501, https://doi.org/10.1063/5.0064522.
[7] R.L. Dewar, A.H. Glasser, Phys. Fluids 26 (10) (1983) 3038–3052, https://doi.org/10.1063/1.864028.
[8] M.A. Beer, S.C. Cowley, G.W. Hammett, Phys. Plasmas 2 (7) (1995) 2687–2700, https://doi.org/10.1063/1.871232.
[9] B. Scott, Phys. Plasmas 8 (2001) 447, https://doi.org/10.1063/1.1335832.
[10] B. Scott, Phys. Plasmas 5 (1998) 2334, https://doi.org/10.1063/1.872907.
[11] M. Wiesenberger, M. Held, L. Einkemmer, A. Kendl, J. Comput. Phys. 373 (2018) 370–384, https://doi.org/10.1016/j.jcp.2018.07.007.
[12] R. Aymar, P. Barabaschi, Y. Shimomura, Plasma Phys. Control. Fusion 44 (5) (2002) 519–565, https://doi.org/10.1088/0741-3335/44/5/304.
[13] D.D. Ryutov, Phys. Plasmas 14 (6) (2007) 064502, https://doi.org/10.1063/1.2738399.
[14] J. Wesson, D.J. Campbell, Tokamaks, vol. 149, Oxford University Press, 2011.
[15] F. Halpern, P. Ricci, S. Jolliet, J. Loizu, J. Morales, A. Mosetto, F. Musil, F. Riva, T. Tran, C. Wersal, J. Comput. Phys. 315 (2016) 388–408, https://doi.org/10.1016/j.jcp.2016.03.040, https://www.sciencedirect.com/science/article/pii/S0021999116001923.
[16] M. Held, M. Wiesenberger, A. Stegmeir, Comput. Phys. Commun. 199 (2016) 29–39, https://doi.org/10.1016/j.cpc.2015.10.009, https://www.sciencedirect.com/science/article/pii/S0010465515003896.
[17] A. Stegmeir, T. Body, W. Zholobenko, Comput. Phys. Commun. (2023) 108801, https://doi.org/10.1016/j.cpc.2023.108801, https://www.sciencedirect.com/science/article/pii/S0010465523001467.
[18] F. Hariri, M. Ottaviani, Comput. Phys. Commun. 184 (11) (2013) 2419–2429, https://doi.org/10.1016/j.cpc.2013.06.005, https://www.sciencedirect.com/science/article/pii/S0010465513001999.
[19] A. Stegmeir, D. Coster, O. Maj, K. Lackner, Contrib. Plasma Phys. 54 (4–6) (2014) 549–554, https://doi.org/10.1002/ctpp.201410041, https://onlinelibrary.wiley.com/doi/abs/10.1002/ctpp.201410041.
[20] R. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
[21] P. Gunawan, R. Eymard, S. Pudjaprasetya, Comput. Geosci. 19 (2015) 1197–1206, https://doi.org/10.1007/s10596-015-9533-4.
[22] R. Herbin, J.-C. Latché, T. Nguyen, ESAIM Proc. 40 (2013) 83–102, https://doi.org/10.1051/proc/201340006.
[23] Raphaèle Herbin, Jean-Claude Latché, Trung Tan Nguyen, ESAIM: M2AN 52 (3) (2018) 893–944, https://doi.org/10.1051/m2an/2017055.
[24] B. Cockburn, C.W. Shu, J. Sci. Comput. 16 (3) (2001) 173–261, https://doi.org/10.1023/a:1012873910884.
[25] J.S. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods, 1st edition, Springer, New York, NY, 2008.
[26] M. Wiesenberger, M. Held, Feltor v7.0, https://doi.org/10.5281/zenodo.596442, 2023.
[27] M. Wiesenberger, L. Einkemmer, M. Held, A. Gutierrez-Milla, X. Saez, R. Iakymchuk, Comput. Phys. Commun. 238 (2019) 145–156, https://doi.org/10.1016/j.cpc.2018.12.006.
[28] C.A. Kennedy, M.H. Carpenter, Appl. Numer. Math. 44 (1) (2003) 139–181, https://doi.org/10.1016/S0168-9274(02)00138-1, https://www.sciencedirect.com/science/article/pii/S0168927402001381.
[29] P. Bogacki, L. Shampine, Appl. Math. Lett. 2 (4) (1989) 321–325, https://doi.org/10.1016/0893-9659(89)90079-7, https://www.sciencedirect.com/science/article/pii/0893965989900797.
[30] M. Wiesenberger, M. Held, L. Einkemmer, J. Comput. Phys. 340 (2017) 435–450, https://doi.org/10.1016/j.jcp.2017.03.056.
[31] F. Hariri, P. Hill, M. Ottaviani, Y. Sarazin, Phys. Plasmas 21 (8) (2014) 082509, https://doi.org/10.1063/1.4892405.
[32] A. Stegmeir, O. Maj, D. Coster, K. Lackner, M. Held, M. Wiesenberger, Comput. Phys. Commun. 213 (2017) 111–121, https://doi.org/10.1016/j.cpc.2016.12.014.
[33] S. Günter, Q. Yu, J. Krüger, K. Lackner, J. Comput. Phys. 209 (1) (2005) 354–370, https://doi.org/10.1016/j.jcp.2005.03.021, https://www.sciencedirect.com/science/article/pii/S0021999105001373.
[34] S. Akkoul, R. Ledee, R. Leconge, R. Harba, IEEE Signal Process. Lett. 17 (6) (2010) 587–590, https://doi.org/10.1109/LSP.2010.2048646.
[35] J.-L. Guermond, B. Popov, V. Tomov, Comput. Methods Appl. Mech. Eng. 300 (2016) 402–426, https://doi.org/10.1016/j.cma.2015.11.009, https://www.sciencedirect.com/science/article/pii/S004578251500362X.
[36] M. Shashkov, S. Steinberg, J. Comput. Phys. 118 (1) (1995) 131–151, https://doi.org/10.1006/jcph.1995.1085, https://www.sciencedirect.com/science/article/pii/S0021999185710856.
[37] V. Nemov, Nucl. Fusion 28 (10) (1988) 1727–1736, https://doi.org/10.1088/0029-5515/28/10/002.