Computer Physics Communications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer Physics Communications

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

Dirichlet boundary conditions for arbitrary-shaped boundaries in stellarator-like magnetic fields for the Flux-Coordinate Independent method

Peter Hill *, Brendan Shanahan, Ben Dudson

York Plasma Institute, University of York, United Kingdom

ARTICLE INFO

Article history:
Received 8 August 2016
Received in revised form 21 October 2016
Accepted 22 November 2016
Available online xxxx

Keywords:
Plasma
Stellarator

ABSTRACT

We present a technique for handling Dirichlet boundary conditions with the Flux Coordinate Independent (FCI) parallel derivative operator with arbitrary-shaped material geometry in general 3D magnetic fields. The FCI method constructs a finite difference scheme for $\nabla_\parallel$ by following field lines between poloidal planes and interpolating within planes. Doing so removes the need for field-aligned coordinate systems that suffer from singularities in the metric tensor at null points in the magnetic field (or equivalently, when $q \to \infty$). One cost of this method is that as the field lines are not on the mesh, they may leave the domain at any point between neighbouring planes, complicating the application of boundary conditions.

The Leg Value Fill (LVF) boundary condition scheme presented here involves an extrapolation/interpolation of the boundary value onto the field line end point. The usual finite difference scheme can then be used unmodified. We implement the LVF scheme in BOUT++ and use the Method of Manufactured Solutions to verify the implementation in a rectangular domain, and show that it does not modify the error scaling of the finite difference scheme. The use of LVF for arbitrary wall geometry is outlined.

We also demonstrate the feasibility of using the FCI approach in non-axisymmetric configurations for a simple diffusion model in a "straight stellarator" magnetic field. A Gaussian blob diffuses along the field lines, tracing out flux surfaces. Dirichlet boundary conditions impose a last closed flux surface (LCFS) that confines the density. Including a poloidal limiter moves the LCFS to a smaller radius.

The expected scaling of the numerical perpendicular diffusion, which is a consequence of the FCI method, in stellarator-like geometry is recovered. A novel technique for increasing the parallel resolution during post-processing, in order to reduce artefacts in visualisations, is described.

© 2016 Published by Elsevier B.V.

1. Introduction

Anisotropic phenomena are prevalent in magnetised plasmas. The Lorentz force tends to confine charged particles to magnetic field lines, with the result that the characteristic size of spatial variations of macroscopic plasma quantities are larger in the direction parallel to the magnetic field compared to those in the perpendicular plane.

Computational techniques take advantage of this anisotropy by, for example, aligning the computational grid to the magnetic field and reducing the resolution in the parallel direction. However, field-aligned coordinate systems typically have difficulties handling changes in magnetic topology; X-points, for instance, introduce singularities in the metric tensor. The Flux Coordinate Independent (FCI) parallel derivative operator [1–4] does not require a field-aligned coordinate system, allowing the use of simpler grids in the perpendicular plane while still allowing efficient handling of anisotropic physics.

In this work, we extend the FCI technique to handle arbitrarily shaped boundaries, including limiters, and demonstrate its use in stellarator-like fields. This work is organised as follows: in Section 2, we explain the FCI method and discuss its implementation; in Sections 3 and 4, we discuss some issues about interpolation and non-axisymmetric magnetic fields; simulations of stellarator-like magnetic fields are in Section 5. We also describe a novel technique for upscaling visualisations in Section 5.2.

2. Flux-Coordinate Independent method for parallel derivatives

Conventionally in magnetised plasma turbulence simulations, derivatives parallel to the magnetic field are taken by using a field-aligned coordinate system. However, these are tied to flux surfaces,

* Corresponding author.
E-mail address: Peter.Hill@york.ac.uk (P. Hill).

http://dx.doi.org/10.1016/j.cpc.2016.11.004
0010-4655/© 2016 Published by Elsevier B.V.

Please cite this article in press as: P. Hill, et al., Dirichlet boundary conditions for arbitrary-shaped boundaries in stellarator-like magnetic fields for the Flux-Coordinate Independent method, Computer Physics Communications (2016), http://dx.doi.org/10.1016/j.cpc.2016.11.004

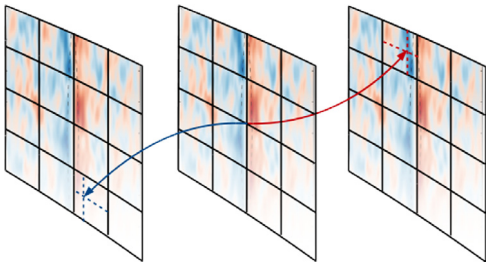

Fig. 1. Schematic of the Flux Coordinate Independent method for the parallel derivative operator. Starting from a given grid point, magnetic field lines are traced in the forward and backward directions. The argument of the operator is interpolated to find the value at the location where the field line intersects the adjacent perpendicular slices, allowing a finite difference scheme to be constructed.

and hence suffer from inevitable singularities in the metric tensor when attempting to encompass multiple magnetic topologies, i.e. crossing separatrices. These singularities can be numerically challenging to handle.

The Flux-Coordinate Independent (FCI) method for the parallel derivatives of a function is conceptually simple: one first follows the magnetic field line from a given grid point in both directions until it intersects the two adjacent perpendicular planes (see Fig. 1). The function to be differentiated is then interpolated in the perpendicular plane at the field intersection points, and a finite difference scheme can be constructed using these values and the value at the emitting grid point. Higher order finite difference schemes may be constructed by following the field line past further perpendicular planes, interpolating at each intersection point. It should be noted at this point that while FCI is strictly formulated on perpendicular planes, in practice, poloidal planes are often used. This is a reasonable approximation, given the assumptions of strong anisotropy required by FCI, and we use the terms "perpendicular" and "poloidal" interchangeably throughout this work. Note that this approximation, and the identification of perpendicular and poloidal planes, is weaker in low-aspect ratio tokamaks, and breaks down entirely in devices such as Reversed-Field Pinches.

As the finite difference scheme is constructed at each individual grid point, the coordinate system in the perpendicular plane is no longer tied to the flux surfaces and in principle any mesh may be used. Other concerns may limit the choice of mesh, e.g. the need for easy flux-surface averages, which may require a flux-surface mesh in part of the plasma. Another consideration is that while it is possible to vastly drop the resolution in the parallel direction (i.e. the inter-plane spacing) with only a small loss in accuracy, similar to conventional field-aligned grids, one must still retain enough resolution in the perpendicular mesh to capture the relevant physics of interest.

2.1. Comparison with the standard BOUT++ mesh

BOUT++[5–7] is a free and open source framework designed to solve partial differential equations, with an emphasis on models of magnetically confined plasmas. It has been used for a variety of applications, from edge[8–10] and scrape-off layer[11,12] physics in tokamaks, to turbulence in linear devices[13,14].

BOUT++ discretises space on a three-dimensional mesh, with the dimensions labelled x, y and z. Typically, x is the "radial" direction, y the "poloidal", and z the "toroidal". The conventional "ballooning"-style BOUT++ coordinate system[5,15], for ψ, θ, ζ the usual orthogonal tokamak coordinates, is defined as:

$$x = \psi, \qquad y = \theta, \qquad z = \zeta - \int_{\theta_0}^{\theta} \nu d\theta, \tag{1}$$

where ν is the local field line pitch, given by

$$\nu(\psi, \theta) = \frac{\partial \zeta}{\partial \theta} = \frac{\vec{B} \cdot \nabla \zeta}{\vec{B} \cdot \nabla \theta}. \tag{2}$$

By keeping z fixed and moving in y, the integral in z changes so we need to move in ζ. This moves us along a field line. Essentially, y is the coordinate along the field line while z picks out different field lines. Because the physics of interest are expected to be field-aligned, we are able to use a lower resolution in y and still resolve the physical scales.

The metric tensor for this coordinate system is orthogonal only at one y-location, meaning as we move in y, cross-terms appear in the x-derivatives. It is possible to eliminate these cross-terms by applying a shifted metric[1,16]. To do this, at each y-point, we can shift z by the integral in Eq. (1), effectively moving us back into non-field-aligned coordinates, performing the derivatives in x, and then transforming back to the field-aligned coordinates. This can be done using Fast Fourier Transforms (FFTs) which are computationally inexpensive.

At either y-end of the grid we need to shift in z in order to match the field lines in a twist-shift boundary[17]. This needs to be done regardless of whether or not we choose to use the shifted metric to eliminate the x-derivative cross-terms.

In contrast to the standard BOUT++ coordinate system, the FCI method explicitly does not use field-aligned coordinates. The construction of the parallel derivatives in fact has the major advantages of a field-aligned system (reduced resolution in the parallel direction) but allows more freedom in the choice of coordinates for the perpendicular directions. For example, two possible choices of coordinate system are tokamak coordinates:

$$x = \psi, \qquad y = \zeta, \qquad z = \theta, \tag{3}$$

or cylindrical coordinates:

$$x = R, \qquad y = \zeta, \qquad z = Z. \tag{4}$$

FCI inherently employs a shifted metric, so no cross-terms appear in the perpendicular derivatives, simplifying the calculations, and no twist-shift has to be performed.

2.2. Boundary conditions

2.2.1. Simple geometry

While FCI has already been implemented in other codes [1,2,4] and used for plasma simulations [3], the boundaries of the simulation domain were either periodic, or treated very simply. The problem is how to treat field lines correctly when they intersect with or leave the simulation boundaries. For example, in Ref. [2], the magnetic topology was a cylinder, and a mask was applied to the simulation domain such that the equations were not solved outside of a radius r. A different solution was used in Ref. [3], where the simulation was periodic in two directions, and the component of the magnetic field in the third direction was damped close to the edges, such that the resulting field was tangential to the edge. Field lines then never intersected the domain boundaries, and boundary conditions could be applied in the perpendicular direction only.

Let us first consider a scalar field f on a simple, uniform, rectangular grid with boundaries located at half the grid spacing outside the first and last points in each of the grid dimensions. For any given point in the grid where the field line traced from this point intersects the boundary before intersecting the next perpendicular plane, we need to be able to calculate parallel derivatives. This situation is depicted in Fig. 2, where f_2 is the value of the scalar field at the point in question, f_1 and f_3 are the values at the intersection points with the adjacent perpendicular planes in the negative and

Please cite this article in press as: P. Hill, et al., Dirichlet boundary conditions for arbitrary-shaped boundaries in stellarator-like magnetic fields for the Flux-Coordinate Independent method, Computer Physics Communications (2016), http://dx.doi.org/10.1016/j.cpc.2016.11.004

Download English Version:

https://daneshyari.com/en/article/4964493

Download Persian Version:

https://daneshyari.com/article/4964493

Daneshyari.com