Computer Physics Communications 212 (2017) 39–46

Contents lists available at ScienceDirect

Computer Physics Communications

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

A partially mesh-free scheme for representing anisotropic spatial variations along field lines

Ben F. McMillan

Centre for Fusion, Space and Astrophysics, University of Warwick, Coventry, United Kingdom

ARTICLE INFO

Article history:
Received 8 April 2016
Received in revised form 15 September 2016
Accepted 14 October 2016
Available online 31 October 2016

Keywords:
Anisotropic
Galerkin
Mesh-free

ABSTRACT

A common numerical task is to represent functions which are highly spatially anisotropic, and to solve differential equations related to these functions. One way such anisotropy arises is that information transfer along one spatial direction is much faster than in others. In this situation, the derivative of the function is small in the local direction of a vector field **B**. In order to define a discrete representation, a set of surfaces M_i indexed by an integer i are chosen such that mapping along the field **B** induces a one-to-one relation between the points on surface M_i to those on M_{i+1}. For simple cases M_i may be surfaces of constant coordinate value. On each surface M_i, a function description is constructed using basis functions defined on a regular structured mesh. The definition of each basis function is extended from the surface M along the lines of the field **B** by multiplying it by a smooth compact support function whose argument increases with distance along **B**. Function values are evaluated by summing contributions associated with each surface M_i. This does not require any special connectivity of the meshes used in the neighbouring surfaces M, which substantially simplifies the meshing problem compared to attempting to find a space filling anisotropic mesh. We explore the numerical properties of the scheme, and show that it can be used to efficiently solve differential equations for certain anisotropic problems.

1. Introduction

The technique proposed here is motivated by plasma physics examples where particles travel much more easily along magnetic field lines than in the perpendicular direction, so that in quantities like fluid moments elongated structures are formed, aligned with the field lines. In particular, the technique is designed to solve problems in magnetic confinement fusion (MCF), where the field lines wind around a central axis and may be closed, trace out surfaces, or fill ergodic regions. An additional difficulty in MCF problems is that the anisotropic structures are strongly curved, because field lines are not straight (even in cylindrical coordinates) over the length scale of the structures; the departure from straightness is often considerably larger than the wavelength of the structure in the directions of rapid variation. This paper outlines a method for representing functions aligned along field lines which are not necessarily aligned on nested surfaces or closed (such as plasmas with an X-point), and for solving equations relating these functions.

A variety of techniques to deal with representing these highly anisotropic functions exist. The canonical technique is to define a 3D mesh to fill the space of interest, with the mesh strongly elongated along the field line. Achieving a very good alignment of the mesh along the field lines is in general quite a difficult meshing problem, and for this reason many MCF physics codes work only in the region where the field lines trace out a nested set of topologically toroidal magnetic surfaces: these are KAM tori [1] associated with the field line Hamiltonian. In the tokamak core, for example, because of near-axisymmetry, nested surfaces usually exist and regular grids can efficiently be generated, or angular coordinates may be employed in conjunction with a Fourier representation. This is not the case for stellarator geometry or in the tokamak edge region.

To avoid difficult meshing problems for the general case where the region of interest is not filled by nested surfaces, it is desirable to relax the requirement of mesh connectivity. The *Flux Coordinate Independent* (FCI) approach [2–4], based on a finite difference method, defines function values on nodes lying on a set of surfaces M_i which are taken to be surfaces of constant coordinate ζ. A node $\mathbf{x}$ on surface M_i can be mapped along the field direction **B** to find image points, $\mathbf{x}_\pm$ on surfaces $M_{i\pm1}$. Although these image points will not in general lie on nodes on the surfaces $M_{i\pm1}$, the function may be evaluated at these points by interpolation. Given the values of the function at points $\mathbf{x}_\pm$, derivatives along the field direction may then be determined.

Another way to relax the mesh connectivity constraint is via a finite volume technique, where the volumes are extrusions of a polygonal grid cell on one surface to the next, and a polynomial

E-mail address: B.F.McMillan@warwick.ac.uk.

http://dx.doi.org/10.1016/j.cpc.2016.10.004
0010-4655/© 2016 Elsevier B.V. All rights reserved.

representation is chosen in each volume element; smoothness constraints are then approximately imposed using a discontinuous Galerkin approach. A hybrid method incorporating finite differences along the field line and the discontinuous Galerkin method has also been investigated [5].

A natural method for representing the anisotropic functions of interest is to change coordinates by defining a grid on a surface, and extending this to a volume grid by defining an additional coordinate parameterising the distance along the mapping (this is known as the flux tube method [6] in MCF). Locally, this allows for straightforward and efficient representation of the problem anisotropy. However, the coordinate scheme becomes highly distorted for mappings with strong shear or compression. The mesh connectivity problem also resurfaces if the originating surface is eventually mapped back onto itself, as at this point the representation on two non-aligned meshes must be combined in some way.

We propose a partially mesh-free method which we call FCIFEM as it is a Finite Element Method translation of the FCI approach. The method represents anisotropic functions using a compact-support set of basis functions which are defined in a local set of coordinates aligned with the mapping. The definition of the set of basis functions is used to define weak forms of differential equations, as in a standard Galerkin method. The philosophy is to design a method which is robust and simple to implement, and requires little manual user interaction, because it avoids complex mesh generation tasks. The representation also provides a simple way to neatly tackle a series of related problems with slightly different configurations, generated, for example, when the field generating the mapping varies slowly with time.

2. Definition of the finite dimensional representation

For the sake of simplicity, we consider a 3D volume labelled by coordinates R, Z, ζ, and take the surfaces M of interest to be surfaces of constant coordinate ζ, of value ζ_i on surface M_i.

Consider a continuous function $\mathcal{Q} : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}^3$, that we will refer to as the *mapping*, which takes a point $\mathbf{x}$ and a parameter s and returns a point $\mathbf{y}$. We will use this function to define projections of the 3D space, along curves locally aligned with the direction of the anisotropy, onto each of the surfaces M_i; we require $\mathcal{Q}(\mathbf{x}, s)_\zeta = s$, so that the projection associated with surface M_i has parameter $s = \zeta_i$. We also require $\mathcal{Q}(\mathbf{x}, x_\zeta) = \mathbf{x}$ so points on the surface map to themselves.

One way to generate such a mapping would be to consider the action of a static flow field $\mathbf{B}$ displacing the position, leading to a field line equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{B}(\mathbf{x}). \tag{1}$$

If we followed a field line from position $\mathbf{x}$ until it had toroidal coordinate s, where the field line was at the point $\mathbf{y}$, we could define the mapping as $\mathcal{Q}(\mathbf{x}, s) = \mathbf{y}$. We will refer to this as an exact mapping, which satisfies the equation

$$\frac{\partial}{\partial \epsilon} \mathcal{Q}(\mathbf{x} + \epsilon \mathbf{B}, \zeta)|_{\epsilon=0} = 0. \tag{2}$$

It is convenient to allow the mapping function to be more general, however, and not necessarily exactly be the solution to this field line mapping equation (or to any equation with a modified $\mathbf{B}$), either because we do not know the exact solution, or because an approximate solution is numerically more desirable. This has consequences for the quality of approximation, as the anisotropic direction will not align exactly with the mapping direction, and certain statements on convergence will be shown only for the case of an exact mapping. For consistency properties to hold, the mapping will be required to be one-to-one and at least of the same order of smoothness as the element functions defined in the next paragraph. The geometry of this mapping is shown in Fig. 1.

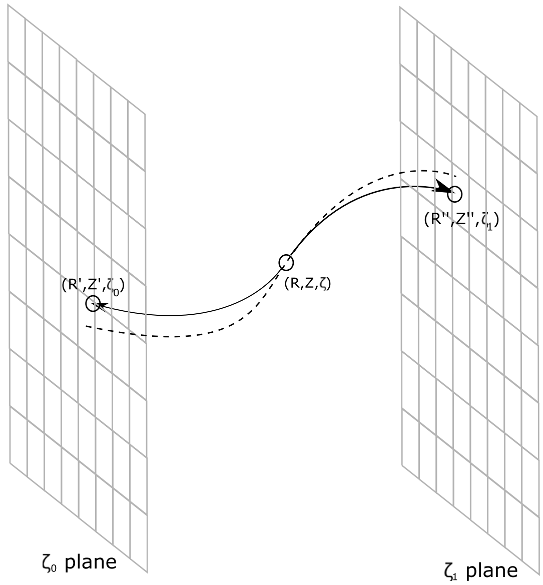

Fig. 1. The geometry of the approximate flow line mapping. Here, the mapping from the point $\mathbf{x} = (R, Z, \zeta)$ to the point (R', Z', ζ_0) on surface M_0 is depicted, with $(R', Z') = \mathcal{Q}(\mathbf{x}, \zeta_0)$, as well as the analogous mapping to surface M_1. The dashed line shows the field line from the point $\mathbf{x}$, for which Eq. (2) holds exactly.

The second element of the FCIFEM method to be chosen is the representation on the planes M_i. In general it might be helpful to choose a general unstructured mesh, but for the purposes of explanation and initial testing in this paper, we will use a simple uniformly spaced Cartesian mesh on each plane M. In the interior region the representation of a scalar function of position is defined as

$$\begin{aligned} \phi(R, Z, \zeta) = \sum_{i,j,k} & \phi_{i,j,k} \times \Omega_R \left[\mathcal{R}(R, Z, \zeta, \zeta_k) - R_i\right] \\ & \times \Omega_Z \left[\mathcal{Z}(R, Z, \zeta, \zeta_k) - Z_j\right] \\ & \times \Omega_\zeta \left[\zeta - \zeta_k\right] \end{aligned} \tag{3}$$

with compact support basis functions Ω, a regular set of Cartesian nodes i, j, k, and using the componentwise notation $\mathcal{Q} = (\mathcal{R}, \mathcal{Z}, \zeta)$. We will choose the functions Ω to be B-Spline basis functions for the remainder as their properties are sufficient to ensure smoothness and lowest order consistency (and this is similar to a finite element approach used earlier in MCF codes [7,8]). An example of the shape of a distorted 2D basis function (the coefficient of $\phi_{i,j,k}$ for some chosen i, j and k) is plotted in Fig. 2. To evaluate the function value at point $\mathbf{x}$, each term of the sum in Eq. (3) is evaluated by calculating the mapping $\mathcal{Q}(\mathbf{x}, \zeta_k)$, and the product of the basis functions can then be directly calculated. For a smooth mapping, the overall representation smoothness depends on the order of the spline. The arguments about convergence are most simply made in the case with uniform nodes where $R_i = i\,\delta R$, $Z_j = j\,\delta Z$ and $\zeta_k = k\,\delta\zeta$. The space spanned by these functions will be denoted S.

3. Basic properties and consistency of FCIFEM

Although it is less obvious than in a standard Finite Element formalism, these elements have a partition of unity property, and can represent the unity function exactly. Substituting unity in the spline coefficients, rearranging the sums and defining quantities