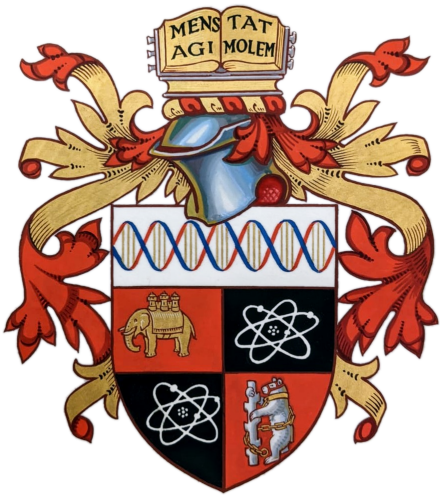
WARWICK
THE UNIVERSITY OF WARWICK



Adapting Meshfree Galerkin Schemes for Representing Highly Anisotropic Fields

by

Samuel Austin Maloney

A thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy in Physics

University of Warwick

Department of Physics

October 2023

Contents

List of figures

List of tables

Declaration

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy in Physics. It has been composed by myself and has not been submitted in any previous application for any degree. The work presented (including data generated and data analysis) was carried out by the author. Parts of this thesis have been published by the author:

[1] Maloney, S. A. & McMillan, B. F. A partially mesh-free scheme for representing anisotropic spatial variations along field lines: Conservation, quadrature, and the delta property. *Computer Physics Communications* **284,** 108629. doi:10.1016/j.cpc.2022.108629 (Mar. 2023)

In particular, much of chapters 4 and 5, and parts of chapter 8 and section 2.4 are mostly from this paper.

Abstract

Two methods for representing highly anisotropic fields are presented, based on partially and fully meshfree Galerkin formulations. In both, a mapping function is used to provide information about the local direction of the anisotropy, with one of the global coordinates chosen to parameterise the 'parallel' position along the mapping in a one-to-one manner. In the first method, dubbed FCIFEM, standard unstructured finite element meshes are used on planes of constant parallel coordinate to represent the necessary small-scale variations perpendicular to the mapping direction, with large spacings then possible between these planes because of the small variation along the mapping. This greatly reduces the number of degrees of freedom required to represent fields in this space and the associated computational cost of simulations involving such fields. No mesh connectivity is defined between planes, and field-aligned basis functions are constructed using the mapping function to extend the standard finite element bases into the full domain. In the second method, dubbed FCIMLS, the meshfree moving least squares (MLS) formalism is used to compute bases capable of representing arbitrarily high-order polynomials. A similar arrangement of nodes into distantly spaced planes is used to help ensure full domain coverage, but now with no defined mesh connectivity required between any nodes. Integration of the bases is addressed with reference to methods developed for other fully meshfree methods, and the schemes (as well as other similar element-free Galerkin schemes) are shown to be locally conservative under certain conditions. Prototype code is developed in Python and used to validate the methods with robust convergence of several test problems being demonstrated. A significant decrease in the number of degrees of freedom required for a given level of accuracy is achieved for model problems with a moderate degree of anisotropy.

List of abbreviations

ASDEX	Axially Symmetric Divertor Experiment
BC	boundary condition
BOUT++	BOUndary Turbulence C++
BVP	boundary value problem
CFL	Courant–Friedrichs–Lewy
CHEASE	Cubic Hermite Element Axisymmetric Static Equilibrium
COGENT	COntinuum Gyrokinetic Edge New Technology
D-T	deuterium and tritium
DG	discontinuous Galerkin
DoF	degree of freedom
EFG	element-free Galerkin
FCI	flux-coordinate independent
FCIFEM	flux-coordinate independent finite element method
FCIMLS	flux-coordinate independent moving least squares
FDM	finite difference method
FELTOR	Full-F ELectromagnetic model in TORoidal geometry
FEM	finite element method
FENICIA	Flux indepENdent fIeld-aligned CoordInate Approach

FFT	fast Fourier transform
FLOPS	floating-point operations per second
FVM	finite volume method
GENE	Gyrokinetic Electromagnetic Numerical Experiment
GYSELA	GYrokinetic SEmi-LAgrangian
IC	initial condition
ICF	inertial confinement fusion
ITER	ITER tokamak (initially the International Thermonuclear Experimental Reactor)
IVP	initial value problem
LCFS	last closed flux surface
LHS	left-hand side
LPI	laser-plasma instability
LS	least squares
MCF	magnetic confinement fusion
MHD	magnetohydrodynamic
MLS	moving least squares
NIF	National Ignition Facility
ODE	ordinary differential equation
PDE	partial differential equation
PIC	particle-in-cell
RHS	right-hand side
RK	Runge-Kutta
SOL	scrape-off layer
VCI	variationally consistent integration
WLS	weighted least squares
XGC	X-point (included) Gyrokinetic Code

CHAPTER 1

Introduction

Climate change is one of the most pressing issues facing our planet today, with rising global temperatures and the looming spectre of increasingly severe weather patterns threatening the sustainability of our ecosystems and the stability of our societies [2]. At the same time humanity's global energy demands are constantly rising, particularly as the billions of people in the global south work to raise their living standards towards those of the global north [3]. One promising avenue of investigation to combat this issue is that of fusion energy, a technology which, if successfully realised in commercial reactors, could offer a reliable, clean, and sustainable source of energy for generations to come.

Current renewable energy technologies such as wind, solar, and hydropower have made significant strides in recent years, both in improving technological prowess and decreasing economic costs, but they still face significant limitations [4, 5]. They are often dependent on the geographic location of specific natural phenomena such as rivers amenable to hydropower or proximity to the Earth's surface of geothermal reservoirs, while climate and weather patterns obviously dictate the suitability of locations for solar or wind farms. This makes the distribution of potentially available renewable capacity not always well-aligned to where the demand for energy is greatest, notably in growing urban centres. Additionally, the energy produced particularly by solar and wind farms is intermittent by nature, which poses challenges for grid operation [6].

In contrast, nuclear energy has the potential to provide a constant and reliable *baseload* source of electricity that can be generated without directly emitting greenhouse gases. Fission, i.e. the splitting apart of atoms into smaller constituents, is the process used in all current commercial nuclear reactors. Much progress has been made in improving the safety of modern fission plants [7–9], but accidents such as the 1986 Chernobyl disaster and the 2011 tsunami at Fukushima have indelibly tainted the public perception of the risks associated with this technology. Advancements

have also reduced the volume of hazardous waste produced, but handling and storage of long-lived radioactive byproducts remains a significant obstacle to fission as a long term energy solution [8, 9].

Nuclear fusion, i.e. combining smaller nuclei into heavier ones, is the process that powers the sun and other stars, and in the context of combatting climate change it represents a promising alternative that could provide a safe and sustainable source of energy for the future. The small quantities of fuel present in a device and robust processes that prevent runaway reactions from occurring make them very safe. The main radioactive byproduct produced is tritium, which is quite short-lived (half-life of 12.33 years [10]), making it eminently tractable even with current engineering solutions to store it until safe for disposal [11]. Some medium- and long-lived isotopes can be created by activation of materials such as steel, but careful choice of low-activation materials can minimise this issue [11]. It also does not require the enrichment and use of fissile materials such as uranium and plutonium which are essential for making atomic weapons, even fusion-based ones which do use tritium, thus mitigating the proliferation concerns which can be an unintended consequence of widespread civilian nuclear fission programs. There are, however, still significant technical and engineering challenges that must be overcome to achieve the vision of making controlled fusion on Earth a reality.

1.1 Approaches to controlled fusion

There are two general approaches to fusion energy, namely magnetic confinement fusion (MCF) and inertial confinement fusion (ICF). In both cases a fuel mixture of deuterium and tritium (D-T), isotopes of hydrogen with two and three nucleons, respectively, is considered to be the most likely candidate due to its large reaction cross-section and large mass defect [12, 13]. The main difference between the two approaches is in the way they confine the fuel and therefore how they attempt to achieve the conditions necessary for fusion to occur.

Denoting the ion density as n_i and the energy-confinement time as τ_E it is generally accepted that the product of these two quantities must exceed a certain threshold, known as the *Lawson criterion* [14] for a reactor design to be viable. It can be expressed as [12]

$$n_i \tau_E \geqslant \frac{12 k_B T}{\langle \nu\sigma \rangle Q}, \tag{1.1}$$

where k_B is the Boltzmann constant, T is the temperature, $\langle \nu\sigma \rangle$ is the fusion reactivity obtained by averaging the product of the relative ion velocity ν and the fusion cross-section σ over all relative velocities, and Q is the kinetic energy of the products of the fusion reaction. It essentially represents a balance between the rate of energy produced by fusion reactions and the rate of energy lost to the environment and thus is a necessary condition for a reactor to achieve energy self-sufficiency, where the

fusion reactions produce enough energy to cause further fusion to occur in a chain reaction known as *ignition*.

A simple estimate of the required threshold value is $n_i \tau_E \geqslant 2 \times 10^{20}$ m^{-3}s^{-1} [15] assuming a temperature of around 10 keV for practical devices using D-T fuel. It is also common for the temperature to instead be explicitly included as a variable on the left-hand side (LHS) to form the *triple product* criterion, giving $n_i T \tau_E \geqslant 26 \times 10^{20}$ m^{-3}s keV [15].

In MCF, the plasma is confined by a strong magnetic field, usually in a topologically toroidal container. A standard 'doughnut' shape used in *tokamak* devices is the most common and well-studied design, originating in the Soviet Union during the 1960s [13], whilst *spherical tokamaks*, which resemble more of a cored apple, have some promising potential advantages [16, 17]. More complex, non-axisymmetric physical and magnetic geometries are used in *stellarator* devices in an attempt to mitigate some of the instabilities associated with their tokamak brethren [18, 19]. All MCF schemes strive to achieve the Lawson criterion by focusing on long energy-confinement times with comparatively low ion pressures, on the order of one atmosphere.

ICF techniques, as the name implies, aim to use only the inertia of the fuel itself to confine the reaction, meaning that by comparison with MCF the energy-confinement times are very short and must be compensated by attaining much higher ion pressures. The most promising approach is to bombard a small fuel pellet with intense laser radiation, ablating the outer material of the pellet and thereby compressing the interior fuel to high density via the reaction forces [12]. Significant results have recently been achieved in this area, particularly at the National Ignition Facility (NIF) [20], but major challenges around aspects such as laser-plasma instabilities (LPIs) [21] and uniformity of the laser irradiation and fuel capsule manufacturing still exist [12], making the roadmap to a practical ICF power plant still unclear.

At the fusion-relevant densities and temperatures in a D-T reactor the hydrogen isotopes will be in a plasma state, which will be discussed next.

1.2 What is a plasma?

Molecules in a gas can become ionised if they collide with sufficient energy to remove one (or more) of the electrons, so the molecule becomes a positively charged ion. In a relatively cold gas this occurs infrequently compared to the timescales on which such free ions and electrons would recombine and so such charged particles are rare. However, at sufficiently high temperatures the frequency of ionising collisions exceeds that of recombination and therefore the fraction of ionised gas becomes non-negligible, and over a narrow temperature range there is a transition to most of the particles being ionised. Such ionised gases are referred to as plasmas, with a more precise definition given by [22] as "a quasineutral gas of charged and neutral particles which exhibits collective behaviour."

Unlike a neutral gas, where particles only interact via direct contact during collisions, the charged particles in a plasma also experience electromagnetic interactions, which can occur over much longer range. This means that particles can 'feel' other particles from much farther away and the effects of forces and movements in one part of the plasma can be transmitted through bulk motions to quite distant regions, which is what is referred to by *collective behaviour* [22, 23].

Unless there is an external source/sink of charged particles, the overall charge density of electrons and ions should be equal ($en_e = q_i n_i$) in such an ionised gas, making a plasma neutral overall. Systems that do not have this overall charge balance are not technically plasmas, even though some similar analyses and results may hold [23]. *Quasineutrality* means that the plasma is, to a very good approximation, neutral, except possibly on very short time or length scales.

In a homogeneous plasma with zero initial electromagnetic field, introducing a point charge causes a shell of oppositely charged particles to accumulate, with the total charge equal in magnitude, such that it screens the introduced charge in a process known as *Debye shielding* [22, 23]. The radius of this shell of charge is known as the *Debye length*, denoted λ_D [22, 23].

The approximate number of particles present in sphere with a radius equal to the Debye length (i.e. a *Debye sphere*) is given by

$$N_D = \frac{4}{3}\pi\lambda_D^3 n, \tag{1.2}$$

and is a useful quantity for characterising a plasma. We also denote L as a characteristic length scale of the overall system geometry, τ as the average time between collisions with neutral atoms, and

$$\omega_p = \sqrt{\frac{ne^2}{\varepsilon_0 m}}, \tag{1.3}$$

as the *plasma frequency*[22], where ε_0 is the permittivity of free space and m is the effective mass of the electron. One then has three crucial conditions that must be satisfied for something to be considered a plasma: [22]

$$\begin{aligned} \lambda_D &\ll L, \\ N_D &\ggg 1, \\ \omega_p\tau &> 1. \end{aligned} \tag{1.4}$$

The numerical methods focused on in this thesis will be primarily motivated by the exigencies arising from simulations of MCF configurations, where there is a strong magnetic field applied to the plasma in the reactor. We will specifically address the development of methods suitable for modelling structures that are highly elongated in one direction, so the next section will address why such structures occur in MCF devices.

1.3 Origin of anisotropy in tokamak turbulence

To see how a high degree of anisotropy arises in a magnetised plasma such as that in a MCF device, we follow the procedure of [24] and start with the *Lorentz Force*

$$\mathbf{F}_L = q(\boldsymbol{v} \times \mathbf{B}), \tag{1.5}$$

which for a single particle with charge q leads to the non-relativistic equation of motion

$$m\frac{d\boldsymbol{v}}{dt} = q(\boldsymbol{v} \times \mathbf{B}). \tag{1.6}$$

For a spatially homogeneous magnetic field with $\mathbf{B} = B\hat{\boldsymbol{z}}$ we can separate the equation of motion into the coordinate directions

$$\begin{aligned} m\dot{v}_x &= qBv_y, \\ m\dot{v}_y &= -qBv_x, \\ m\dot{v}_z &= 0, \end{aligned} \tag{1.7}$$

and the velocity component parallel to $\mathbf{B}$ (i.e. $v_\parallel = v_z$) is unaffected by the magnetic field, so particles stream freely in this direction. This means that structures such as temperature fluctuations tend to "smear out" rapidly in the parallel direction. Based on the timescale of typical processes (e.g. instabilities) one may then estimate the minimum parallel length scale relevant to this process, which is on the order of metres to tens of metres for a large tokamak such as ITER [25]. Notably, Landau damping [26] strongly inhibits the development of high-$k_\parallel$ modes [27, 28], which leads to the so-called "flute" property of turbulent structures [29].

Conversely, particle movement is strongly constrained in the perpendicular plane, as can be seen by taking a further time derivative of equation (1.7) and solving to get the equations for the positions of the particle as

$$\begin{aligned} x &= x_0 + r_g \sin(\omega_g t), \\ y &= y_0 + r_g \cos(\omega_g t), \end{aligned} \tag{1.8}$$

where ω_g is known as the *gyrofrequency* (also *cyclotron frequency*) and is given by

$$\omega_g = \frac{qB}{m}, \tag{1.9}$$

and r_g is known as the *gyroradius* (also *Larmor radius* or *cyclotron radius*) and is given by

$$r_g = \frac{v_\perp}{|\omega_g|} = \frac{mv_\perp}{|q|B}. \tag{1.10}$$

The point $\begin{bmatrix} x_0, & y_0 \end{bmatrix}^\top$ represents the coordinates of the *guiding centre* of the circular motion of the particle in the plane perpendicular to $\mathbf{B}$, which when combined with a non-zero $v_\parallel = v_z$ gives the particle an overall helical trajectory about the magnetic field line which passes through the guiding centre.

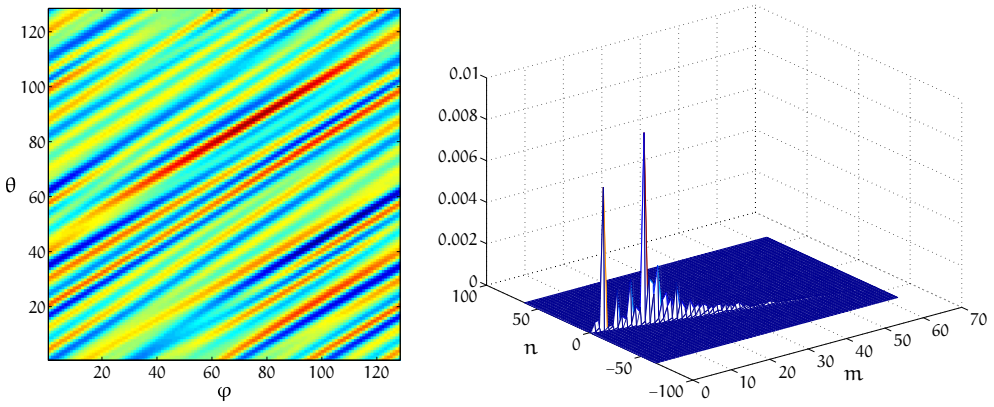


Figure 1.1: Illustrative example of field-aligned anisotropy arising in a plasma simulation performed by Ottaviani [30]. (left) The electric potential on a circular magnetic flux surface, as described in chapter 2, showing a clear alignment of structures in a single direction corresponding to the magnetic field vector. (right) Spectrum of Fourier coefficients after Fourier transforming the field on the left, showing the clustering of information along this single direction, with vanishingly small contributions elsewhere indicating the minimal resolution needed in the parallel direction to still capture the vast majority of the information in the system. This figure is combined from figures 1 and 2 in [30].

For processes that occur on long timescales compared to the gyration period, the gyrating particle tends to feel the average electromagnetic field over its orbit (this is formalised via gyrokinetic theory [29]). Thus particles do not strongly interact with slowly varying fields on length scales short compared to their gyroradius, because of the gyro-orbit-averaging.

For a very rough back-of-the-envelope calculation, we note the gyroradius is also inversely proportional to the magnetic field magnitude, with the maximum field strength of ITER given as $\leqslant 12$ T [25]. Using the thermal velocity

$$v_{\text{th}} = \sqrt{\frac{k_B T}{m}}, \tag{1.11}$$

as a characteristic speed and taking a plasma temperature of roughly 10 keV in ITER [25] this would give a approximate values for the electron and proton gyroradii of

$$r_{g,e} = 3 \times 10^{-5} \text{ m}, \qquad r_{g,i} = 1 \times 10^{-3} \text{ m}. \tag{1.12}$$

And so we can see that the ratio between the physical spatial scales defining the parallel direction (roughly 10 m in ITER) and these small gyro-orbits in the perpendicular plane is roughly four to six orders of magnitude.

Similar results hold true for many quantities of interest, which exhibit behaviour (magnitudes, gradients, fluxes, etc.) that is strongly anisotropic with respect to the spatial orientation of $\mathbf{B}$ throughout the plasma container. To give just a couple of examples, the heat diffusion coefficient can be up to ten orders of magnitude larger parallel to $\mathbf{B}$ compared to perpendicular [31], or tearing mode instabilities due to resistive magnetohydrodynamic (MHD) physics display strong anisotropy localised to

rational surfaces [32] (see section 2.1 for definition of rational surfaces). An illustrative example of such field-aligned anisotropy arising in a plasma simulation is shown in figure 1.1. Therefore, an important challenge in simulations of MCF devices is accurately representing these highly anisotropic fields in a computationally tractable manner.

1.4 Overview of chapters

The next chapter will briefly review various methods for handling the strong anisotropy in plasma simulations, including the method this thesis will investigate. Following that, in chapter 3 is an introduction to the ideas underlying the standard finite element method (FEM) and fully meshfree methods that we will draw upon later, without yet incorporating anything to do with the highly anisotropic problems of interest. Then in chapter 4 we will specialise this general formulation to our specific flux-coordinate independent finite element method (FCIFEM) scheme, which is a blend of standard FEM and meshfree methods and explicitly includes information about the system anisotropy. Chapter 5 contains the numerical results of several test simulations carried out with the FCIFEM scheme, while in chapter 6 we explore preliminary investigations into how the ideas could be extended to higher order via a FCIMLS (flux-coordinate independent moving least squares) scheme, along with initial numerical results for this extension work presented in chapter 7. The thesis concludes in chapter 8.

CHAPTER 2

Field-aligned simulation

Many codes tackle the problem of anisotropy in plasma simulations by using a so-called *magnetic flux coordinate* system such that "the equation of a magnetic field line is that of a straight line" [33]. In theory this could mean that either the magnetic field vector is directly aligned with one of the coordinate directions, or it could be simply a straight-field-line system where the magnetic field vector is constant but not coordinate-aligned, as illustrated schematically in figure 2.1.

In practice fully coordinate-aligned systems are particularly helpful, as in such a system anisotropy can be addressed by simply using fewer degrees of freedom (DoFs) in the coordinate direction parallel to the magnetic field while maintaining high resolution in the other coordinates. This can work well in regions where the magnetic field is well-behaved and traces out closed surfaces, such as within the

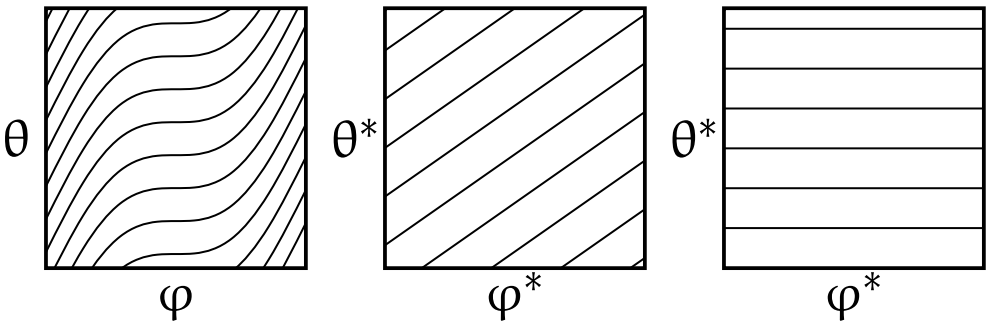


Figure 2.1: Schematic of what magnetic field lines might look like on a 2D flux surface in (left) a standard, (middle) straight-field-line, or (right) coordinate-aligned flux coordinate system. The standard system uses a polar (r, θ, φ) coordinate system where θ and φ are the poloidal and toroidal coordinates, respectively, as in figure 2.5, and where the radial coordinate r is assumed to provide a suitable flux label and is constant for each of the 2D flux surface plots shown. θ^* and φ^* are a generic notation for any coordinate pair transforming θ and φ to the specified magnetic flux coordinate system type.

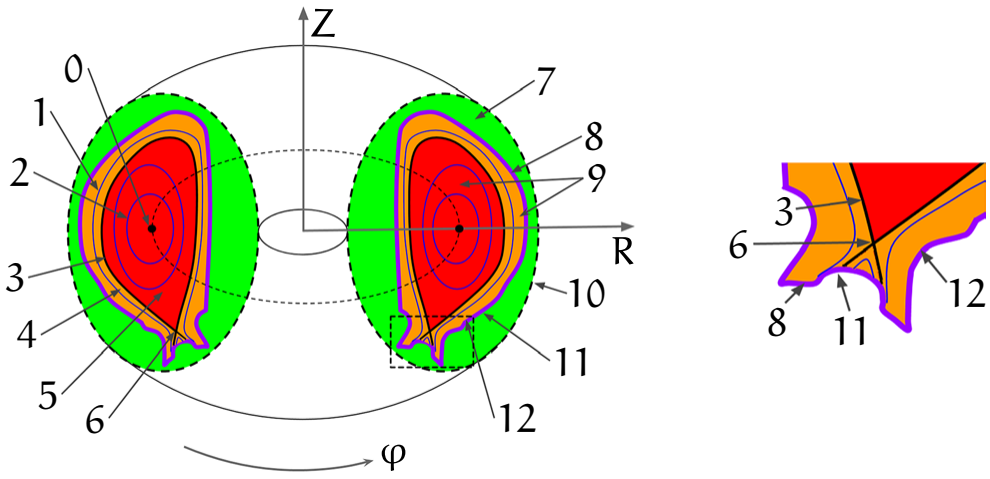


Figure 2.2: Schematic of geometric and magnetic features of a MCF device, with a close-up of the area near the X-point indicated by the dashed box in the the full image. Labelled components include the (0) magnetic axis, (1) open magnetic flux surfaces, (2) closed magnetic flux surfaces, (3) separatrices, (4) scrape-off layer, (5) plasma core, (6) X-points, (7) vacuum vessel, (8) wall area, (9) plasma area, (10) vacuum boundary, (11) outer wall boundary, and (12) inner wall boundary. This figure is taken from figure 2 in [34].

central region of a tokamak; however, in regions near the reactor edges or the *separatrix*, for example, the physical and/or magnetic geometries can become quite complicated and so generating high-quality, field-aligned grids becomes extremely difficult or even impossible. In particular the *X-point(s)* and the *magnetic axis* can induce singularities in such coordinate systems, although with some special care this is worked around in several codes to include the magnetic axis, as noted in section 2.3.2. See figure 2.2 for an illustration of many of the complicated magnetic and geometric features that must be considered for developing simulations of whole MCF devices.

There are infinitely many possible magnetic flux coordinate systems, and so specific choices must be made as to the other properties one wishes for the system to provide in order to settle on a specific system. Details of the derivations are beyond the scope of this thesis, but two of the most popular specific choices are the Hamada coordinates [35–37], in which the magnetic field lines and the lines corresponding to the MHD equilibrium current density are straight, and the Boozer coordinates [38, 39], in which the magnetic field lines and the diamagnetic $\nabla\psi \times \mathbf{B}$ lines are straight. The meaning of this ψ-coordinate and a few other basic concepts of flux coordinates are discussed in the next section.

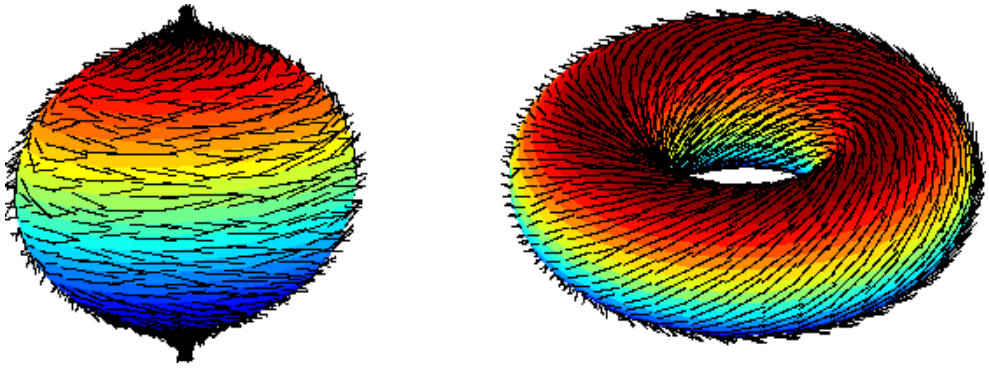


Figure 2.3: Illustration of the Poincaré–Brouwer or "hairy ball" theorem on a sphere and a torus. (left) Visually it can be seen that attempting to comb the hair flat on a topological sphere will always result in at least two points (or a singular degenerate dipole) that form either a "cowlick" or a "bald point". If we then imagine a vector field tangential to the surface as being the projection of the "hairs" onto the surface, we can see that this field will necessarily go to zero at these points. (right) Conversely, we can see that no such points are necessary in combing the hair on the topological torus flat, and so a tangential field which is non-zero everywhere on such a surface is possible. This figure is combined from `https://commons.wikimedia.org/wiki/File:Hairy_ball.png` and `https://commons.wikimedia.org/wiki/File:Hairy_doughnut.png`.

2.1 Flux surfaces and flux labels

A given surface S with its associated normal vector $\hat{\mathbf{n}}$ can be termed a *magnetic flux surface* if it satisfies the condition

$$\mathbf{B} \cdot \hat{\mathbf{n}} = 0, \tag{2.1}$$

for all points on the flux surface. The general definition for the magnetic flux F through any surface is given by

$$F = \int_S \mathbf{B} \cdot \hat{\mathbf{n}} \, dS, \tag{2.2}$$

so one can see that there is zero magnetic flux through a magnetic flux surface because the integrand is zero everywhere on the flux surface by definition.

For a general set of magnetic flux coordinates, one of the coordinates is used as a *flux label*, ψ, which is a function that is constant on a given flux surface, such that

$$\mathbf{B} \cdot \nabla\psi = 0, \tag{2.3}$$

and provides a unique label for selecting a particular nested flux surface. In 3D, as a consequence of the Poincaré–Hopf theorem [40–45] (in particular of its special case of the Poincaré–Brouwer [46] or "hairy ball" theorem, see figure 2.3), such nested flux surfaces for a non-vanishing vector field will be topologically toroidal. Since a non-zero magnetic field is critical for confinement of the plasma in MCF devices, this explains why such devices are almost universally topologically toroidal. (Technically

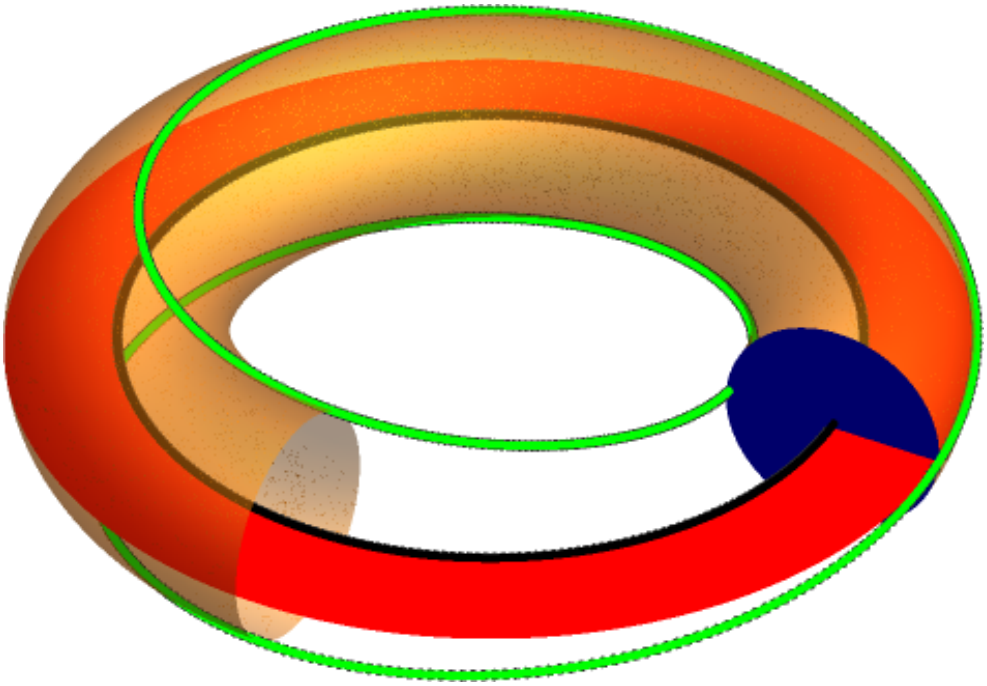


FIGURE 2.4: Schematic of a closed flux surface (yellow) in a toroidal MCF device along with its associated surfaces which are used to compute the poloidal (red) and toroidal (blue) flux labels of the surface. An example field line on the surface is shown in green which wraps back on itself after an integer (2) number of toroidal turns, indicating that this is a low-order rational flux surface. This figure is taken from `https://commons.wikimedia.org/wiki/File:Flux-surface-including-fieldline.png`.

the flux surfaces must be topological knots, of which a torus is the simplest trivial example, sometimes referred to as the *unknot* since it is not "knotted" at all; however, such *knotatron* devices are as yet purely hypothetical and barely studied [47].)

Two common choices for such flux labels [48] are the *poloidal flux*, given by

$$\Psi_p = \int_{S_p} \mathbf{B} \cdot \hat{\mathbf{n}} \, dS, \tag{2.4}$$

where the surface S_p is defined to be a ribbon stretched between the magnetic axis and the given flux surface, as illustrated by the red surface in figure 2.4, and the *toroidal flux*, given by

$$\Psi_t = \int_{S_t} \mathbf{B} \cdot \hat{\mathbf{n}} \, dS, \tag{2.5}$$

where the surface S_t is defined to be a plane of constant toroidal angle, as illustrated by the blue surface in figure 2.4. Note that some sources (e.g. [49]) use a complementary definition of the poloidal flux where the surface S_p is instead defined as a disc covering the "doughnut hole" of the torus stretching from the major axisymmetry axis to the given flux surface.

The *safety factor* is another important quantity encountered in MCF research and is defined as

$$q = \frac{d\psi_t}{d\psi_p}. \tag{2.6}$$

It is so named because it gives critical information about the MHD stability of the plasma, with a value of $q > 2$ required at the last closed flux surface (LCFS) for the plasma to be mostly MHD stable [50] although even then it can still be unstable to $m = 1$ internal modes and ballooning modes, which are also MHD in nature. (In a MCF device with a divertor, it is more useful to consider instead the q_{95} value, which is the safety factor of the flux surface which encloses 95 % of the poloidal flux within the LCFS, because $q \to \infty$ at the separatrix itself). As well, certain macroscopic MHD instabilities have been found to preferentially develop on so-called *rational flux surfaces* [51] where $q = \frac{m}{n}$ is a rational number with $m, n \in \mathbb{Z}$. Since the safety factor can also be defined as the number of toroidal transits of the MCF device a field line makes during a single complete poloidal transit, this means that field lines on such rational surfaces rejoin onto themselves after a finite integer number of toroidal turns, as shown in figure 2.4. The safety factor is constant on a given flux surface and so can itself be used as a flux label, and typically varies from $q \approx 1$ near the magnetic axis to around 3 to 4 at the edge near the LCFS [13, 51].

In addition to the flux label, a complete 3D flux coordinate system will require two other linearly independent coordinates in order to describe locations on the 2D flux surface. In the next section we will follow the lead of Ottaviani and Hariri [30, 51, 52] and provide a brief summary of the historical development of approaches that have been taken for defining field-aligned coordinate systems in order to improve computational efficiency by reducing grid resolution in one of these other two coordinate directions. In order to keep the discussion as simple as possible we will describe only the case of axisymmetric and poloidally symmetric nested toroidal flux surfaces, where the radial coordinate r then serves as the flux label and assume we already have a suitable θ^* and φ^* providing a straight-field-line coordinate system. This simplification is adequate because a transformation such as that to Hamada coordinates can be used as an intermediate step to create such a set-up for more general magnetic geometries if desired [30, 53]. We also note that it is usually sufficient to distort only the θ-coordinate to create a straight-field-line system, leaving $\varphi^* = \varphi$ unchanged, which is particularly useful to preserve toroidal symmetry in tokamak-type devices, but for generality we will use starred versions of both in the following.

In such a system the flux surfaces form nested concentric circles on poloidal planes of constant toroidal angle, and the safety factor $q(r)$, which is a function of the flux label r only, contains the information on the slope of the magnetic field on the constant r flux surfaces. Therefore, the parallel derivative operator, denoting the component of the gradient of a quantity in the direction of the local magnetic field, is given by

$$C(\mathbf{x})\nabla_{\parallel} = \frac{\partial}{\partial \varphi^*} + \frac{1}{q(r)}\frac{\partial}{\partial \theta^*}. \tag{2.7}$$

The factor $C(\mathbf{x})$ is in general a geometry dependent function of position [30], which for a typical tokamak geometry is approximately $C(\mathbf{x}) \approx R$ [51], so that is what shall be used here.

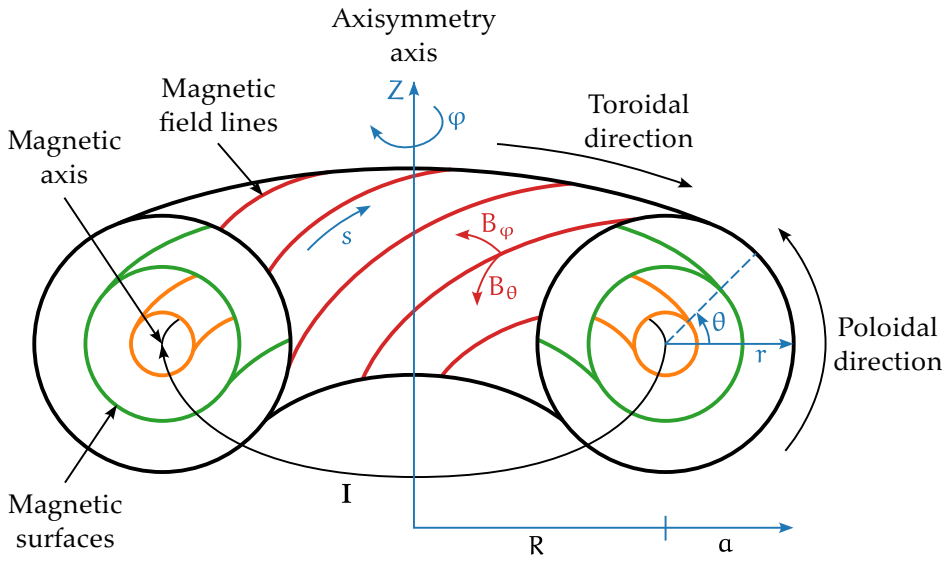


Figure 2.5: Schematic of standard coordinates used for topologically toroidal MCF geometries, namely (r, θ, φ) or (r, Z, φ). R is the major radius and a the minor radius of the torus, assuming an axisymmetric tokamak-type device, although the other general features apply equally to stellarator-type devices. Note that the positive direction of the poloidal (θ) and/or toroidal (φ) angles may be swapped depending on the convention used. Certain features of the magnetic field configuration are also labelled, such as the nested magnetic flux surfaces with the magnetic axis at their centre. This figure is adapted from figure 1.3 in [51].

2.2 Review of field-aligned coordinates

The use of field-aligned coordinates for tackling anisotropy goes back at least to the early 1990s [54] which led to a number of successful uses throughout the following decade [53, 55–57]. These early works used the poloidal angle θ^* to parameterise the location along a given field line, equivalent to the coordinate transformation visualised in figure 2.6, given by

$$\xi = \varphi^* - q(r)\theta^*, \tag{2.8a}$$

$$s = \theta^*, \tag{2.8b}$$

$$\rho = r, \tag{2.8c}$$

where s is the field-aligned coordinate, such that the parallel derivative operator simplifies to

$$R\nabla_\parallel = \frac{1}{q(r)}\frac{\partial}{\partial s}, \tag{2.9}$$

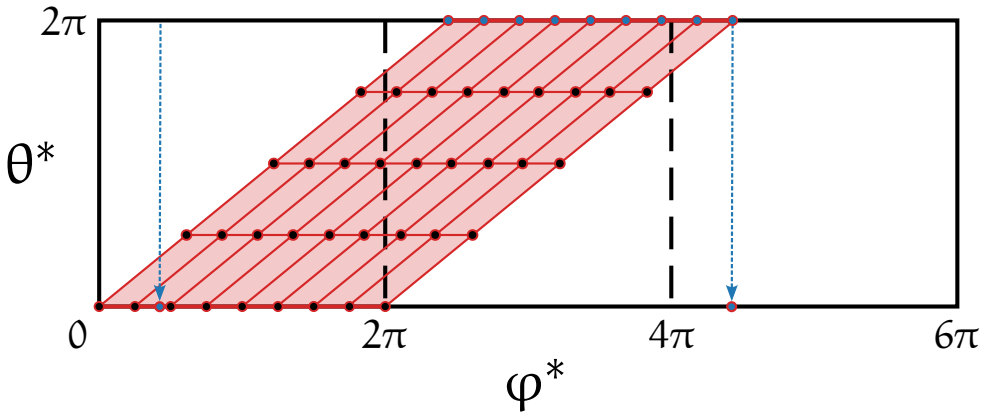


Figure 2.6: Schematic of field-aligned coordinates on a 2D flux surface using the poloidal angle θ^* to parameterise the location along the field line, allowing for decreased resolution in the poloidal coordinate direction. It can be seen with the dashed blue vertical arrows that for a general (non-rational) flux surface the field lines do not necessarily reconnect back onto themselves after a full poloidal rotation. This figure is adapted from figure 1 in [52].

and the other components of the gradient in the original (r, θ^*, φ^*) coordinates can then be written as

$$\frac{\partial}{\partial r} = \frac{\partial}{\partial \rho} - \frac{dq(r)}{dr}\theta^* \frac{\partial}{\partial \xi}, \tag{2.10a}$$

$$\frac{\partial}{\partial \varphi^*} = \frac{\partial}{\partial \xi}, \tag{2.10b}$$

$$\frac{\partial}{\partial \theta^*} = \frac{\partial}{\partial s} - q(r)\frac{\partial}{\partial \xi}. \tag{2.10c}$$

There are, however, a couple of downsides to this coordinate transform as given. Firstly, as is evident from the illustration in figure 2.6, the s-coordinate along the field lines is not trivially periodic (instead, there is a skew periodicity on the s-ξ plane) and so careful attention must be paid to the boundary conditions on s to ensure proper periodicity of the simulations, otherwise non-physical effects may occur [30, 53]. Secondly, the θ^* coefficient appearing in the latter term of the expression for $\frac{\partial}{\partial r}$ produces increasingly mixed derivatives for regions far from $\theta^* = 0$ and its numerical treatment can produce non-physical inhomogeneities even when the physical system should be poloidally symmetric; this problem can be addressed with the shifted-metric technique of the next subsection. Thirdly, this coordinate transform cannot be used at or past the X-point as it runs into trouble at the separatrix, where the safety factor $q(r)$ becomes infinite. This makes the parallel derivative operator go to zero, and the latter term of the expression for ξ dominates such that the ξ- and s-coordinates become increasingly linearly dependent as one approached the separatrix. One can also consider that the poloidal magnetic field is zero at the X-point, and so using the poloidal angle to parameterise the location along the magnetic field lines is not tenable at and near the X-point. This problem is addressed by the switch of coordinates shown in section 2.2.2.

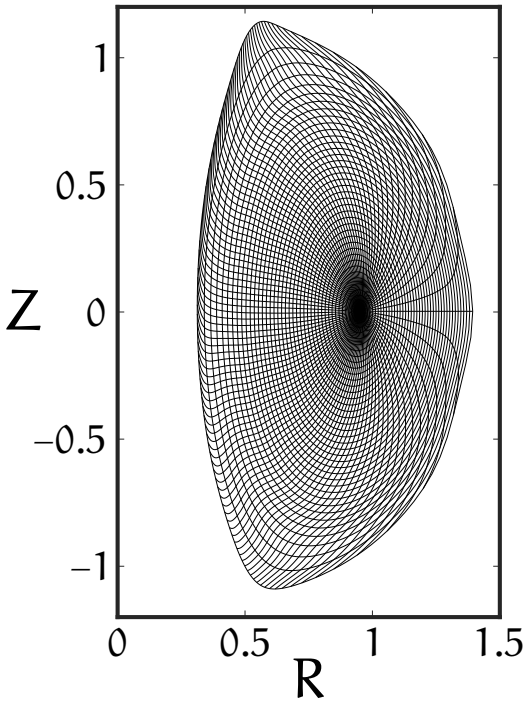


Figure 2.7: Example illustration of the coordinate contour lines of ρ and s within the LCFS of a realistic tokamak geometry. It can be seen that even within this interior region there are large distortions in certain areas of the cross-section, particularly approaching the separatrix, and large variations in the size, shape, and aspect ratio of grid cells throughout the domain. This figure was provided by B. F. McMillan.

Even inside the LCFS, fundamental non-uniformities in topologically toroidal straight-field-line coordinate systems can be problematic. An example is illustrated in figure 2.7, showing the contour lines of constant ρ- and s-coordinates from the circularly concentric straight-field-line system mapped back into the physical (R, Z) lab frame for a realistic tokamak geometry. One can see large variations in the size, shape, and aspect ratio of the Cartesian grid cells throughout the domain, with particularly large distortions near the separatrix. Thus, even in the regions where these field-aligned parameterisations are possible, one still encounters difficulties producing high quality grids for accurate simulations, which is a strong motivation for the technique described later in section 2.4 to avoid requiring straight-field-line coordinates in the first place.

It has been noted in [58] that this set of coordinates is equivalent to the ballooning transformation of [59], which appears to be one of the main benefits of using the poloidal angle to parameterise the location along the field lines, as it greatly simplifies certain aspects of the linear analytic theory [30]. This is also perhaps the reason it was the choice made in the original uses of field-aligned coordinates.

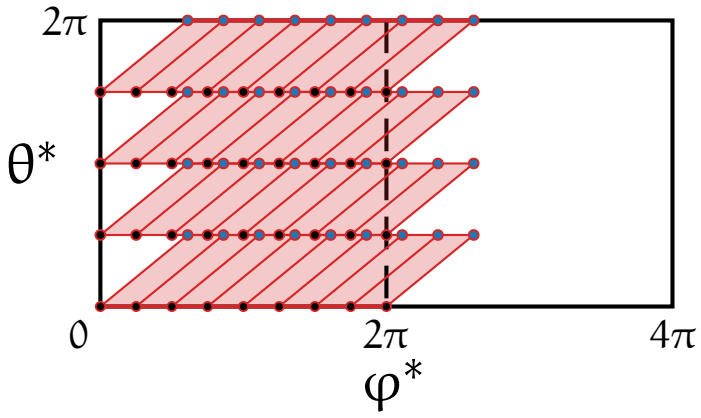


Figure 2.8: Schematic of field-aligned coordinates on a 2D flux surface using the poloidal angle θ^* to parameterise the location along the field line, allowing for decreased resolution in the poloidal coordinate direction. The shifted metric technique is illustrated, with the origin of the coordinate system being reset in each poloidal slice of the domain. This figure is adapted from figure 2 in [52].

2.2.1 Shifted-metric technique

A method for addressing the metric distortion caused by the term proportional to θ^* in the expression for $\frac{\partial}{\partial r}$ was proposed by [60] and has been termed the *shifted-metric* technique. It leads to the modification of the previous coordinate transformation to give the family of transformations

$$\xi = \varphi^* - q(r)(\theta^* - \theta^*_k), \tag{2.11a}$$

$$s_k = \theta^* - \theta^*_k, \tag{2.11b}$$

$$\rho = r, \tag{2.11c}$$

where the origin of the coordinate system is reset at regular intervals in the poloidal angle, as visualised in figure 2.8. The parallel derivative operator remains unchanged as

$$R\nabla_\parallel = \frac{1}{q(r)}\frac{\partial}{\partial s}, \tag{2.12}$$

while the other components of the gradient in the original (r, θ^*, φ^*) coordinates are now written as

$$\frac{\partial}{\partial r} = \frac{\partial}{\partial \rho} - \frac{dq(r)}{dr}(\theta^* - \theta^*_k)\frac{\partial}{\partial \xi}, \tag{2.13a}$$

$$\frac{\partial}{\partial \varphi^*} = \frac{\partial}{\partial \xi}, \tag{2.13b}$$

$$\frac{\partial}{\partial \theta^*} = \frac{\partial}{\partial s} - q(r)\frac{\partial}{\partial \xi}. \tag{2.13c}$$

One can see that this technique essentially amounts to slicing the domain into a series of poloidal wedges, each with its own independent coordinate system, but each of which is fully field-aligned, differing only in the origin point θ^*_k. By coinciding the slice locations where the origins are reset with the grid locations needed for

sufficient resolution in the poloidal direction, one then has a situation where the derivative expressions need only be evaluated at the points where $\theta^* = \theta^*_k$ and so the terms causing the metric distortion are zero at the evaluation points.

Difficulties with the separatrix remain essentially unchanged, while periodicity in s is improved, as the grid points at which the function values are stored (shown in black in figure 2.8) are now actually periodic in θ^*. The difference now is that the field lines from grid points on one slice no longer inherently line-up with grid points at any of the neighbouring slices, not just at $\theta^* = 2\pi$, and so interpolation is generally needed to compute derivatives in the parallel s-coordinate direction.

One of the advantages of using the poloidal angle to parameterise the parallel direction (in contrast to the next section) is that this interpolation is then being carried out in the toroidal angle. For an axisymmetric device such as a tokamak this means that highly accurate interpolation is straightforward and computationally inexpensive to implement using fast Fourier transforms (FFTs).

2.2.2 Switching the roles of toroidal and poloidal angles

In order to address some of the remaining issues, particularly surrounding separatrices, Ottaviani [30] proposed switching the roles of the poloidal and toroidal angles, such that it would now be the toroidal angle φ^* which would parameterise the location along a given field line. This new approach is visualised in figure 2.9 and the family of coordinate transformations defining this formulation is given by

$$\xi = \theta^* - \frac{1}{q(r)}(\varphi^* - \varphi^*_k), \tag{2.14a}$$

$$s_k = \varphi^* - \varphi^*_k, \tag{2.14b}$$

$$\rho = r, \tag{2.14c}$$

with the parallel derivative operator now even further simplified to

$$R\nabla_\parallel = \frac{\partial}{\partial s}, \tag{2.15}$$

and the other components of the gradient in the original (r, θ^*, φ^*) coordinates now written as

$$\frac{\partial}{\partial r} = \frac{\partial}{\partial \rho} - \frac{dq(r)}{dr}\frac{\varphi^* - \varphi^*_k}{q^2(r)}\frac{\partial}{\partial \xi}, \tag{2.16a}$$

$$\frac{\partial}{\partial \varphi^*} = \frac{\partial}{\partial s} - \frac{1}{q(r)}\frac{\partial}{\partial \xi}, \tag{2.16b}$$

$$\frac{\partial}{\partial \theta^*} = \frac{\partial}{\partial \xi}. \tag{2.16c}$$

One can see that now when the safety factor goes to infinity as one approaches the separatrix, the factor of $\frac{1}{q(r)}$ simply goes to zero, the coordinates remain linearly independent, and so the immediate problems of the original approach are avoided. The parallel derivative operator no longer depends directly on the safety factor at

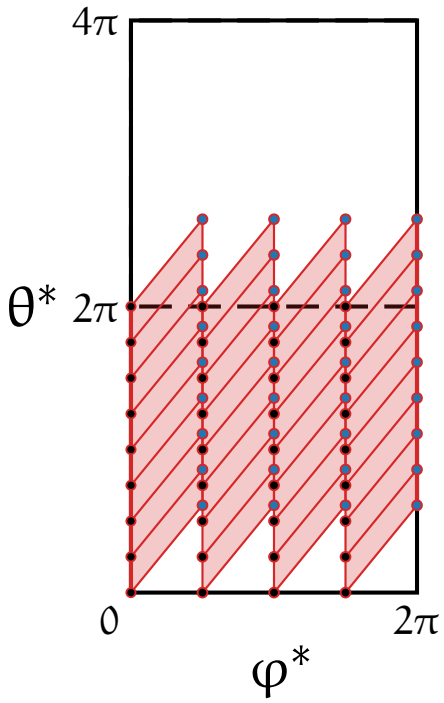


Figure 2.9: Schematic of field-aligned coordinates on a 2D flux surface using the toroidal angle φ^* to parameterise the location along the field line, allowing for decreased resolution in the toroidal coordinate direction. The shifted metric technique is illustrated, with the origin of the coordinate system being reset in each poloidal slice of the domain. This figure is adapted from figure 3 in [52].

all, and parameterising in terms of the toroidal angle still makes physical sense near the X-point where the poloidal magnetic field component goes to zero. Generally the shifted metric technique would still be used, only now dividing the domain into toroidal slices where the origin is reset to avoid problems with metric distortion from the term proportional to the toroidal angle φ^* in the expression for $\frac{\partial}{\partial r}$.

Difficulties near the separatrix due to the distortions of the underlying straight-field-line coordinate grids, as shown previously in figure 2.7, are still an issue; however, this approach can then be extended to avoid the transformation to field-aligned aligned coordinates entirely, as outlined later in section 2.4, which will ultimately be what we will build upon for the main work of this thesis.

2.3 Code implementation examples

A small, non-exhaustive selection of various plasma turbulence codes and the numerical approaches used by them is given here, with the intent just to give a few specific examples of how grid-related issues have been handled in prior codes. This discussion is restricted primarily to kinetic codes, which have especially stringent needs for good spatial grids because they also have to handle velocity-space, making them the primary target for the anisotropic methods explored in this thesis.

2.3.1 Eulerian or continuum codes

One example is the GENE (Gyrokinetic Electromagnetic Numerical Experiment) code, which is an Eulerian gyrokinetic code solving the 5D Vlasov equations using a finite difference method (FDM) and explicit Runge-Kutta (RK) time integration[61]. The code was originally written [62] to run only using a so-called *local* or *flux tube* [57] geometry. This means that the simulation domain only covers a small portion of the full poloidal cross-section of the MCF device being studied, with this small cross-section then extruded along the local magnetic field lines in the toroidal direction to produce a thin, tube-like domain, hence the nomenclature. Because of the flux coordinate system used [63] this domain appears as a simple rectangle in the simulation grid space, but in real physical space the tube is twisted and the cross-section distorted into a parallelogram by magnetic shear. If the shear is strong and the domain long enough for the distortion to accumulate to a sufficient extreme then it will fail to capture the turbulent features of interest, which has led to the development of a *non-twisting flux tube* discretisation with promising improvement in accuracy [64].

GENE was also later extended to provide the option for running simulations on what is called a *nonlocal* or *global* domain, in the sense that it has a large enough cross-section that radial profiles of properties such as temperatures and densities can also be included [65–67]. However, even this nonlocal version is limited to regions where the magnetic field forms closed flux surfaces, and so the code was again extended with features required for simulations of regions with open magnetic field lines, particularly of the *scrape-off layer (SOL)* outside of the LCFS of a MCF device [68]. Finally, in order to include the separatrix in the domain and combine both core and SOL regions simultaneously in the same simulations, an extension called GENE-X [69] was implemented to incorporate the flux-coordinate independent (FCI) scheme discussed further in the next section. Unlike the method investigated in this thesis, they still use FDMs on a locally Cartesian grid discretising each poloidal plane.

A related approach to handling the spatial anisotropy including the magnetic separatrix is that taken by the COGENT (COntinuum Gyrokinetic Edge New Technology) code [70, 71] which uses a finite volume method (FVM) discretisation coupled with a locally field-aligned multi-block coordinate system. They show promising initial simulation results, but the method still requires interpolation at toroidal block boundaries along with special considerations to maintain numerical conservation at these interfaces. The underlying mesh is also still composed of structured grids within each block, making representation of complex physical geometries at the reactor edges difficult.

Although as mentioned this thesis is primarily interested in kinetic simulations, it is perhaps worth mentioning that some fluid MHD codes such as JOREK [72–74]

are able to handle complex magnetic and physical geometries, including X-points and open field regions out to the reactor walls. JOREK uses a 2D quadrilateral FEM discretisation on the poloidal plane with a Fourier expansion in the toroidal direction. The poloidal meshes need not be flux surface aligned, but generally are for improved accuracy, and the FEM discretisation allows good flexibility in conforming the grid to the geometry. The original code is essentially limited to axially symmetric devices such as tokamaks, but recent work on JOREK3D [75] has been undertaken to also enable simulation of stellarators.

2.3.2 Lagrangian PIC codes

ORB5 is another well-used global gyrokinetic code [76, 77] which can simulate the whole core plasma region inside the LCFS, and therefore not passing the separatrix or X-point(s). It is a Lagrangian particle-in-cell (PIC) code which uses a straight-field-line flux coordinate system with a special coordinate transformation near the magnetic axis to avoid the singularity which would otherwise arise there [77]. It has several features for reducing particle noise [78–80] but perhaps most interestingly from the perspective of field-aligned anisotropy, it uses a Fourier space representation for its linear equations which allows it to use a filtering procedure on the Fourier coefficients to remove non-physical nodes which are not well-aligned to the magnetic field, also greatly reducing particle noise and computational cost [76, 77].

XGC (X-point (included) Gyrokinetic Code) comprises a small suite of Lagrangian PIC codes [81–84] capable of simulating the full plasma volume out to the boundaries of the vacuum vessel, including both core and edge plasmas across the separatrix and X-point (as the name implies) as well as a finite thickness reactor wall if desired. It uses an unstructured grid on each poloidal plane with a field-following procedure to determine nodal placement [34], an example of which is given in figure 2.10. This is one of the more flexible approaches of any code, but still has challenges when reconnecting back to the original poloidal plane and the field-following procedure implicitly assumes device symmetry about the major axis such that the nodes will actually align on successive poloidal meshes, making it less generalisable to stellarator-type devices.

2.3.3 Semi-Lagrangian codes

Eulerian/continuum and Lagrangian codes each have advantages and disadvantages. The most troublesome for PIC codes is particle noise, due to essentially using a Monte Carlo sampling for determining quantities of interest, with the noise only converging slowly with increasing numbers of particles, whilst requiring substantial computational resources [29]. Eulerian codes avoid this noise, but are subject to often large amounts of non-physical numerical dissipation, which can aid with numerical stability but can also be large enough to impact the physical interpretation

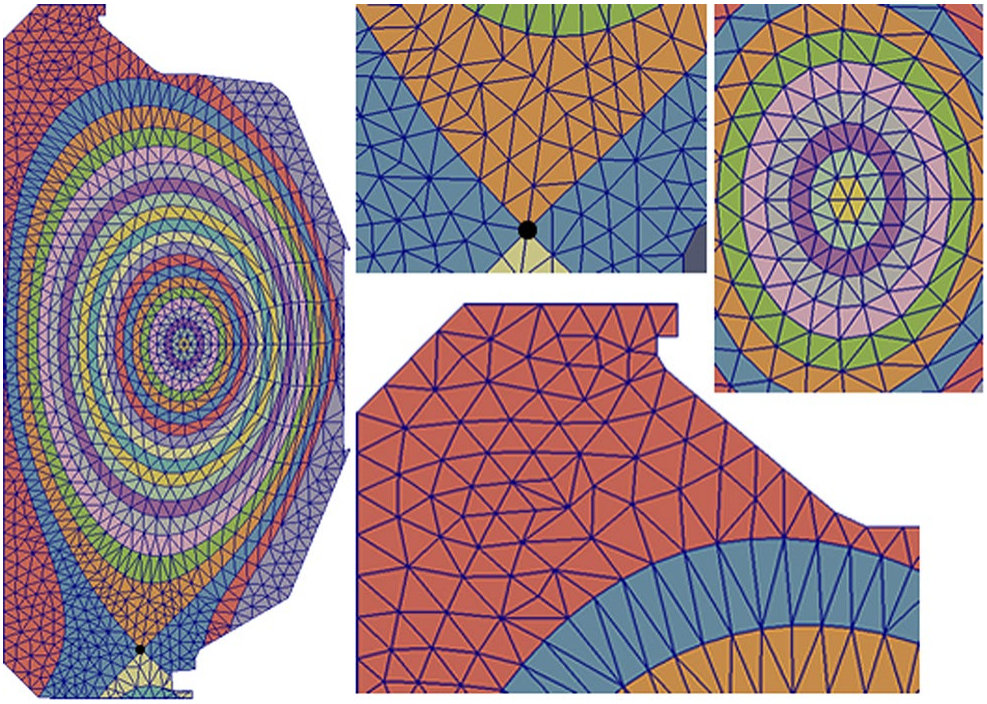


Figure 2.10: Example of an unstructured mesh for the XGC code discretising a poloidal plane in a tokamak with a single X-point. The unstructured nature allows seamless mixing of parts of the mesh following the closed magnetic flux surfaces in the well-behaved core of the plasma with unaligned refinements near the X-point and good conformation with arbitrary wall geometries. This figure is taken from figure 11 in [34].

of simulation results [29, 85]. The common use of explicit time-stepping schemes also strongly constrains the largest stable time step due to the Courant–Friedrichs–Lewy (CFL) condition [86] in many continuum codes.

An intermediate approach between the two is that of a semi-Lagrangian code such as the GYrokinetic SEmi-LAgrangian (GYSELA) code [87, 88] which uses a mesh similar to that in a continuum code to avoid particle noise but also avoids the CFL constraint similar to a PIC code by integrating the node orbits backward along their characteristics and using an interpolation procedure to compute the new value. A field-aligned interpolation procedure [89] has been investigated to greatly reduce the computational cost by leveraging the anisotropy, and has also been implemented in GYSELA [90], but is still fundamentally based on flux coordinates and regular Cartesian meshes giving rise to similar limitations as for their Eulerian cousins.

2.4 Flux coordinate independent method

The FCI class of methods has been proposed by Ottaviani [30] as one possibility to ameliorate concerns from the previous methods of accounting for anisotropy in simulations. Firstly, as the name implies, it renders unnecessary the need for

selecting any particular flux coordinate system or field-alignment scheme; rather, standard physical coordinates such as (r, θ, φ) or (r, Z, φ) can be used throughout. Secondly, it obviates the need for difficult anisotropic mesh generation, by relaxing the connectivity requirements between points on the mesh in the direction parallel to the magnetic field.

At this point it would be prudent to define exactly what is meant by a 'mesh' in this thesis. Notably, both standard and meshfree methods generally use a set of nodes/cells throughout the domain on which to define their solutions, and so we consider the 'mesh' to refer rather to the connectivity between these nodes/cells. In code, the 'mesh' would then be a (set of) data structure(s) defining not only the node/cell locations, but also which nodes/cells interact or connect to which others, and perhaps other geometrical information that can be precomputed and stored. Mesh generation would refer to the process of computing and storing this data, which for a meshfree method is then largely avoided since only node locations need to be defined.

For FCI, in essence, a series of standard 2D meshes (i.e. with connectivity information) are created on successive planes crossing the magnetic field, which we will call *FCI planes*, in order to capture the large variations possible in directions perpendicular to the magnetic field. In contrast, the spacing between adjacent meshes is much greater than the internodal spacing within each individual mesh, as it needs only to resolve the much slower variations in the direction parallel to the magnetic field, thus leveraging the spatial anisotropy to greatly reduce the number of DoFs to be solved for. This setup is illustrated in figure 2.11, and one can see that there is no mesh connectivity precomputed between planes.

In the original FCI formulation, and the quasi-totality of implementations to date, FDMs are used for computing derivatives. Since there is no defined connection between nodes in separate meshes, a mapping procedure is used to compute the location of points on neighbouring planes at which information must be obtained in order to discretise the parallel derivative operator with a FDM, for example using the second-order discretisation

$$R\nabla_{\parallel}(\cdot) \approx R\frac{(\cdot)_{k+1} - (\cdot)_{k-1}}{\varphi_{k+1} - \varphi_{k-1}}, \tag{2.17}$$

where the subscript k is a label for a given FCI plane. This mapping will generally not map a grid point on one FCI plane to a grid point on its neighbouring planes, and so interpolation must be performed in order to finally compute the derivatives in the direction parallel to the magnetic field.

Fundamentally, by retaining the physical lab frame coordinate systems (usually cylindrical), the FCI methodology should allow simulations of full reactor cross-sections, including non-axisymmetric stellarator-type devices, separatrices and X-point(s), both closed and open field line regions, and magnetic islands or stochastic/ergodic regions. An example of some of these latter challenging magnetic

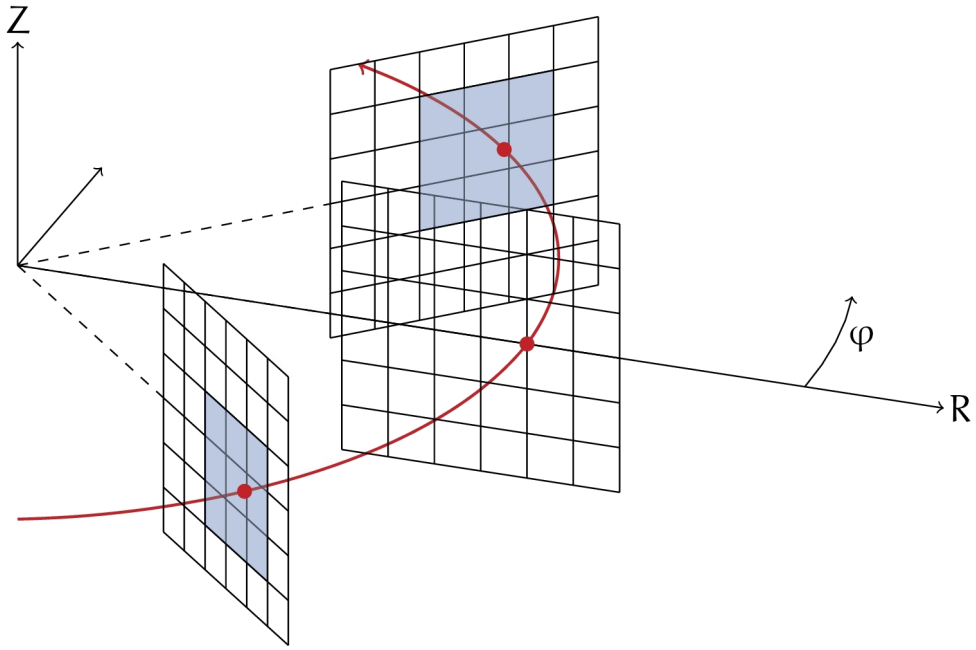


Figure 2.11: Illustration of the original FDM-based FCI scheme showing the mapping along the field line from a grid point to its nearest neighbour FCI planes. In general this will not intersect directly at a grid point on the neighbouring planes and so an interpolation procedure involving a set of nearby grid points/cells, such as those shaded in blue, must be used to compute a value for the quantity of interest at each end of the mapping line in order to fully define the FDM approximation of the parallel derivative operator. This figure is taken from figure 3 in [69].

features is shown in figure 2.12, which is a Poincaré plot from the Axially Symmetric Divertor Experiment (ASDEX) upgrade tokamak showing an ergodic layer caused by magnetic islands being formed in the usually closed field line region of the core plasma.

2.4.1 Code implementations

Refinements to the original FCI method have been made both by Hariri, Ottaviani, and Hill et al. [51, 52, 92, 93] in their FENICIA (Flux indepENdent fIeld-aligned CoordInate Approach) code and by Stegmeir, Ross, Body, and Zholobenko et al. [94–106] in their GRILLIX code. Implementations of the method have also been realised in several other codes including GENE-X, as mentioned previously, and BOUT++ (BOUndary Turbulence C++), a plasma fluid simulation code [107–110]. All of these implementations are still FDM-based, as mentioned above, and while e.g. the BOUT++ code has shown some success at modelling plasma filaments [111] and arbitrary wall geometry [112], an even more general framework allowing the use of unstructured grids would be desirable.

Additionally, FDM schemes generally do not account for conservation properties when discretising conservation laws. This makes the FCI implementation in the

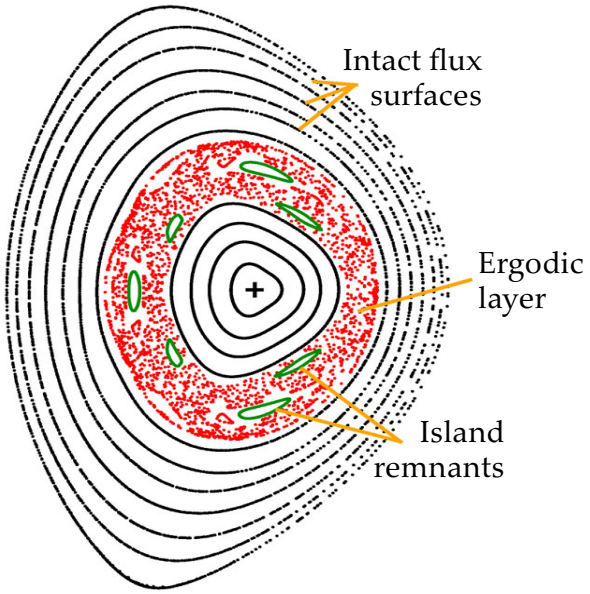


Figure 2.12: Poincaré plot of magnetic flux surfaces in the ASDEX upgrade tokamak. An ergodic or stochastic layer is clearly seen where the otherwise closed field lines in the core of the tokamak plasma have become chaotic due to the formation of magnetic islands. This figure is taken from figure 2.10 in [91].

FELTOR (Full-F ELectromagnetic model in TORoidal geometry) code [113–115] particularly interesting, as it uses discontinuous Galerkin (DG) schemes, and has very recently seen the addition of a FVM-type FCI scheme [116]. Both of these methodologies should allow for much better realisation of conservation properties, but the implementations appear to still use structured grids, with the associated limitations thereof. One drawback to FVM and DG methods is that in general they produce discontinuous representations of fields and their derivatives. In conjunction with suitable flux limiters, this property can make such schemes extremely useful for simulations involving shocks, such as is necessary in simulations for ICF; however, in MCF devices the fields are generally sufficiently smooth that shock-capturing codes are unnecessary, while having appropriately smooth representations can be very helpful for simulations. For example, PIC codes will need to compute the interactions of particles with fields at arbitrary points in the domain, and so having smooth representations is beneficial to avoid extra noise from field discontinuities.

This thesis will focus primarily on extending the FCI technique to use FEM-based approaches for the numerical discretisation, building on the efforts first demonstrated by McMillan [117]. This would allow for unstructured meshes to be used on the perpendicular planes, thus greatly assisting with conforming these grids to both the complex physical geometry at the edges of MCF reactors and complex magnetic geometry near the X-points and separatrices or islands and stochastic layers in the interior. An example of such a mesh was shown previously in figure 2.10 of a mesh generated for the XGC code, and our FEM-based methodology would allow

such meshes to be used with the FCI technique. While XGC assumes axisymmetry, with a FCI-type implementation one could use meshes which are similar looking, but that are independent on individual poloidal planes to also allow for arbitrary non-axisymmetric physical and magnetic field profiles on each poloidal plane.

Our initial methodology will be broadly referred to as FCIFEM, and makes use of a meshfree-type formulation to compute the field discretisation between the planes, with a later extension to a fully meshfree scheme to achieve higher-order convergence.

CHAPTER 3

Basic Galerkin methods

Given that the initial method proposed in chapter 4 of this thesis will consist of a blending of ideas between FEM and meshfree schemes, this chapter will give a very brief introduction to the relevant parts of the established literature for those numerical methods. A thorough investigation of these techniques has already filled many a textbook, and so the content here will be quite a limited selection of what will be needed for the novel developments in the later chapters. Some further implementation aspects for moving least squares (MLS) based methods will also be detailed in chapter 6 covering the additional steps necessary for extending the ideas from our FCIFEM scheme to a higher-order accurate method.

3.1 Finite element method

The main idea is to provide a method for approximating continuous functions, and thereby an associated computational method for finding approximate solutions of partial differential equations (PDEs). A FEM formulation starts by decomposing the simulation domain Ω into a set of N_e discrete elements: lines in 1D and most often simplices in 2D and 3D (i.e. triangles and tetrahedra, respectively) although other shapes such as rectangles and boxes are also possible in the 2D and 3D cases. The vertices of this mesh form the set of N_n nodes at which the discrete solution will be computed. A set of *basis functions* (a.k.a. *trial functions*) $\phi_i(\mathbf{x}) : \Omega \to \mathbb{R}$ is then selected, one associated to each node, with which the final numerical solution will be approximated by the following expansion

$$u(\mathbf{x}, t) \approx u^h(\mathbf{x}, t) = \sum_{i=1}^{N_n} u_i(t)\phi_i(\mathbf{x}), \tag{3.1}$$

where $u(\mathbf{x}, t)$ is the continuous solution function of space and time one wishes to approximate by the FEM approximant $u^h(\mathbf{x}, t)$, and both are defined throughout the

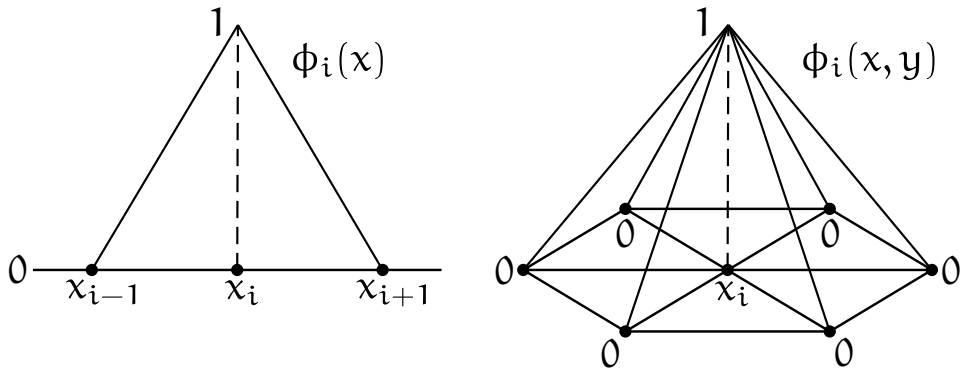


Figure 3.1: Illustration of linear Lagrange FEM hat functions in 1D (left) and 2D (right) showing their compact support and Kronecker delta property, i.e. they are non-zero only in the neighbouring elements of their associated node and they go to zero as they reach their neighbouring nodes.

entire domain Ω. The $u_i(t)$ are the discrete coefficients of the corresponding basis functions, and form the set of unknowns to be solved for in the numerical simulation. Since the $u_i(t)$ are scalar coefficients they have no spatial dependence, while the $\phi_i(\mathbf{x})$ do not vary with time. This separation of variables in the approximation allows one to focus on the discretisation of the space and time dimensions separately. Here we will address the the spatial discretisation first followed by the temporal discretisation in section 3.2.

The basis functions ϕ_i are usually defined to have *compact support*, meaning that they are non-zero in value only in a small portion of the domain within the local neighbourhood of the node to which they are associated, often one or two layers of mesh elements surrounding the node. Additionally, the ϕ_i are usually defined to have a value of unity at the associated node $\mathbf{x}_i$ itself, and a value of zero at every other node, including nodes on the edge of or within the region of the function's compact support, where it could otherwise be non-zero in value. This results in the basis function coefficients being equal to the value of the numerical solution function evaluated at the respective nodal point, i.e. $u^h(\mathbf{x}_i, t) = u_i(t)$ for all nodes, a result known as the *Kronecker delta property*. Example illustrations of the most commonly used type of linear Lagrange basis functions (often referred to as *hat functions*) are given in figure 3.1 for 1D and 2D.

3.1.1 Galerkin assembly

For this thesis we will focus on the standard Galerkin formulation of the weighted residual approach to spatial discretisation, beginning with the strong form of a PDE defining a problem of interest

$$\frac{\partial u}{\partial t} = \mathcal{L}(u) + f, \tag{3.2}$$

where $\mathcal{L}$ is a spatial differential operator, $f = f(\mathbf{x}, t)$ is a forcing term, and $u = u(\mathbf{x}, t)$ is an unknown function of both space and time in general. One can then arrive at the weak form of the problem by taking the inner product of the equation with a suitable *test function*, which for the most common Bubnov-Galerkin method is taken to be from the same set of trial basis functions used for the approximation, giving

$$\left\langle \frac{\partial u}{\partial t} \,\middle|\, \phi_i \right\rangle = \left\langle \mathcal{L}(u) \,\middle|\, \phi_i \right\rangle + \left\langle f \,\middle|\, \phi_i \right\rangle, \tag{3.3}$$

where the angle brackets denote the L^2 inner product defined by

$$\left\langle g \,\middle|\, h \right\rangle = \int_\Omega g(\mathbf{x}) h(\mathbf{x}) \, d\Omega. \tag{3.4}$$

We can then substitute the expansion of u from equation (3.1) into equation (3.3) to give

$$\left\langle \frac{\partial}{\partial t} \left(\sum_{j=1}^{N_n} u_j(t) \phi_j(\mathbf{x}) \right) \,\middle|\, \phi_i(\mathbf{x}) \right\rangle = \left\langle \mathcal{L} \left(\sum_{j=1}^{N_n} u_j(t) \phi_j(\mathbf{x}) \right) \,\middle|\, \phi_i(\mathbf{x}) \right\rangle + \left\langle f \,\middle|\, \phi_i(\mathbf{x}) \right\rangle, \tag{3.5a}$$

$$\sum_{j=1}^{N_n} \frac{\partial u_j}{\partial t} \left\langle \phi_j \,\middle|\, \phi_i \right\rangle = \sum_{j=1}^{N_n} u_j \left\langle \mathcal{L}(\phi_j) \,\middle|\, \phi_i \right\rangle + \left\langle f \,\middle|\, \phi_i \right\rangle, \tag{3.5b}$$

which can then be rewritten as a matrix equation

$$\mathbf{M}\dot{\mathbf{u}} = \mathbf{K}\mathbf{u} + \mathbf{f}, \tag{3.6}$$

where $\mathbf{M}$ is often referred to as the *mass matrix*, $\mathbf{K}$ is the *stiffness matrix* or *operator matrix*, the overdot represents time differentiation, and the individual matrix entries are given as

$$\mathbf{M}_{ij} = \left\langle \phi_j \,\middle|\, \phi_i \right\rangle, \qquad \mathbf{K}_{ij} = \left\langle \mathcal{L}(\phi_j) \,\middle|\, \phi_i \right\rangle, \qquad \mathbf{f}_i = \left\langle f \,\middle|\, \phi_i \right\rangle. \tag{3.7}$$

These matrix entries are computed in a simulation code by looping over each of the elements and using some numerical quadrature to approximate the integral in the inner product defined in equation (3.4) over just the local element area to find the local (or elemental) stiffness matrices $\mathbf{K}^e$ and load vectors $\mathbf{f}^e$. While in this thesis we consider only the Galerkin weighted residual approach given above for computing these local matrices, other approaches such as using a suitable variational principle or equilibrium conditions (also termed the direct method) are possible [118].

The local contributions are then *assembled* to form the global linear system by summing over all elements as [119]

$$\mathbf{K}_{ab} = \sum_{e=1}^{N_e} \mathbf{K}^e_{ab}, \qquad \text{and} \qquad \mathbf{f}_a = \sum_{e=1}^{N_e} \mathbf{f}^e_a, \tag{3.8}$$

where the integer subscripts $(a, b) \in [1, N_n]$ show the link between the (row, column) indices of the entries in the global stiffness matrix $\mathbf{K}$ and load vector $\mathbf{f}$ to the indices of the set of nodes whose associated basis functions are non-zero over the element e on which the local stiffness matrix and load vector are computed.

3.1.2 Diffusion equation example

To give a concrete example of this procedure we look at the weak form and operator matrix contributions for the diffusion equation

$$\frac{\partial u}{\partial t} = \mathbf{D}\nabla^2 u + f, \tag{3.9}$$

where the *diffusion coefficient* $\mathbf{D}$ is taken to be constant in time and space.

Taking the inner product with the test functions according to equation (3.3) gives us the corresponding weak form of the equation as

$$\sum_{j=1}^{N_n} \frac{\partial u_j}{\partial t} \int_\Omega \phi_j \phi_i \, d\Omega = \sum_{j=1}^{N_n} u_j \int_\Omega \left[\mathbf{D}\nabla^2 \phi_j\right] \phi_i \, d\Omega + \int_\Omega f \phi_i \, d\Omega. \tag{3.10}$$

The contributions to the mass matrix on the LHS are quite standard, while for the diffusion term on the right-hand side (RHS) we can apply integration by parts to get the contributions to the operator matrix as

$$\begin{aligned} \mathbf{K}_{ij} &= \mathbf{D} \int_\Omega \phi_i \nabla^2 \phi_j \, d\Omega, \\ &= \mathbf{D} \left[\int_\Gamma \phi_i \nabla \phi_j \cdot \hat{\mathbf{n}} \, d\Gamma - \int_\Omega \nabla \phi_j \cdot \nabla \phi_i \, d\Omega \right]. \end{aligned} \tag{3.11}$$

This use of integration by parts is essential for this example of solving a second-order PDE with linear elements, as it reduces the order of derivatives of the trial functions that must be computed to first order, whereas the second-order derivatives would obviously zero. Integration by parts is also found to play an important role in local flux conservation as elucidated in section 4.2. It does, however, introduce a boundary term in the weak form that must be computed separately in some cases. For most standard FEM bases that possess the Kronecker delta property, the nodes at Dirichlet boundaries are not DoFs and so do not technically have associated basis functions in the trial space, while all other trial functions will go to zero on said boundaries, so this boundary term will produce zero contribution and can be ignored. For meshfree trial functions such as those described shortly in section 3.3 neither of these things are true and the contributions from the boundary term will be non-zero and therefore cannot be neglected, thus necessitating an additional numeral quadrature procedure on Dirichlet boundaries.

It is also worth noting that if one eliminates the time derivative and sets the diffusion coefficient $\mathbf{D}$ to scalar unity then the diffusion equation in equation (3.9) reduces to the Poisson equation. This means that the Poisson equation can be discretised using the same weak form, operator matrix contributions, and boundary considerations as derived above for the diffusion equation, and we will use the Poisson equation as the paradigmatic elliptic equation when testing our numerical schemes.

3.2 Time integration

Having completed the spatial discretisation of the problem, the semi-discrete matrix equation in equation (3.6) is then a system of ordinary differential equations (ODEs) which can be solved numerically using a suitable time integration method, of which there are many available choices in common use. This procedure of discretising a PDE in all but one dimension (usually omitting time) and then integrating the solution with an ODE solver in the remaining dimension is often referred to as the *method of lines*.

For this thesis a few simple time integration options were implemented, including a low-storage, explicit, m-stage RK scheme. Equation (3.6) can be rewritten as

$$\dot{\mathbf{u}} = \mathbf{M}^{-1}\left(\mathbf{K}\mathbf{u} + \mathbf{f}\right) = \mathfrak{L}(\mathbf{u}), \tag{3.12}$$

where $\mathfrak{L}$ is merely a simple notation used to represent the spatial discretisation of the continuous differential operator $\mathcal{L}$ (with the additional forcing term now also included), as outlined above. The low-storage RK scheme then has the following Shu-Osher form [120–122]

$$\begin{aligned} \mathbf{u}^{(0)} &\equiv \mathbf{u}^n, \\ \mathbf{u}^{(k)} &= \mathbf{u}^{(0)} + \beta_k \Delta t \mathfrak{L}\left(\mathbf{u}^{(k-1)}\right), \qquad k = 1, 2, ..., m, \\ \mathbf{u}^{n+1} &\equiv \mathbf{u}^{(m)}, \end{aligned} \tag{3.13}$$

where n is the global time index, k is the index of the RK stage within each time step, and $\Delta t = t^{n+1} - t^n$ is the size of the current time step. The specific time-marching scheme is defined by the β_k coefficients, such as the first-order accurate forward-Euler method with only $\beta_1 = 1$ or the fourth-order accurate classical RK4 scheme given by [123]

$$\beta_1 = 0.125, \quad \beta_2 = 0.25, \quad \beta_3 = 0.5, \quad \beta_4 = 1. \tag{3.14}$$

Other available integrator options include the first-order accurate backward-Euler scheme

$$\begin{aligned} \mathbf{M}\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= \mathbf{K}\mathbf{u}^{n+1} + \mathbf{f}, \\ \mathbf{u}^{n+1} &= \left(\frac{\mathbf{M}}{\Delta t} - \mathbf{K}\right)^{-1}\left(\frac{\mathbf{M}}{\Delta t}\mathbf{u}^n + \mathbf{f}\right), \end{aligned} \tag{3.15}$$

and the second-order accurate Crank-Nicolson [124] scheme

$$\begin{aligned} \mathbf{M}\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= \mathbf{K}\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} + \mathbf{f}, \\ \mathbf{u}^{n+1} &= \left(\frac{\mathbf{M}}{\Delta t} - \frac{\mathbf{K}}{2}\right)^{-1}\left[\left(\frac{\mathbf{M}}{\Delta t} + \frac{\mathbf{K}}{2}\right)\mathbf{u}^n + \mathbf{f}\right], \end{aligned} \tag{3.16}$$

both of which are implicit and so can be used to avoid the CFL-type constraints on the maximum stable time step inherent to explicit schemes. Of the two, the

Crank-Nicolson scheme should essentially always be preferred over the backward-Euler scheme, as the matrix needing to be inverted is very similar and thus the Crank-Nicolson scheme delivers an extra order of convergence accuracy despite the computational cost of solving the linear system being almost identical for both methods.

This then gives a complete temporal and spatial discretisation of the PDE in equation (3.2) which can be solved numerically for the FEM coefficients u_i. These, in turn, define via equation (3.1) the discrete approximation u^h for the unknown function of interest u.

3.3 Moving least squares meshfree formalism

There are many different meshfree schemes that have been studied in the literature (see e.g. [125] for a comprehensive review) but in this thesis we will focus exclusively on properties of the MLS formalism first proposed in [126] and developed further in [127]. The following section is essentially a brief summary of the MLS formulation as given in Nguyen et al. [128], as it follows the derivations in a form intended to be well-suited for implementation into code.

MLS methods begin in essentially the same manner as for a FEM, where one needs to define a set of basis functions for approximating continuous functions, which can then be used for discretising PDEs. Each of these basis functions $\Phi_I(\mathbf{x}) : \Omega \to \mathbb{R}$ will again be associated with a node $\mathbf{x}_I$ and are usually referred to as *shape functions* when discussing meshfree schemes, also now denoted with a capital Φ to differentiate them from their FEM counterparts. Essentially the same form of approximation expansion can be used

$$u(\mathbf{x}, t) \approx u^d(\mathbf{x}, t) = \sum_{I=1}^{N_n} u_I(t)\Phi_I(\mathbf{x}), \tag{3.17}$$

with $u^d(\mathbf{x}, t)$ denoting the MLS approximant. The main difference compared to a FEM is that the domain Ω is not split into cells, but rather the set of N_n solution nodes is simply distributed throughout the domain with no well-defined mesh connectivity between them (as illustrated later in figure 3.3).

The following subsections will outline the ideas needed for actually computing these shape functions, with the primary motivating criteria of creating a basis set capable of exactly representing polynomials up to a prescribed order. Section 3.3.1 will discuss 1D weight functions that are necessary to define the size and shape of the compact support for the shape functions as well as determining their continuity, with section 3.3.2 briefly detailing how higher dimensional shapes are formed from these 1D functions. Section 3.3.3 will then go through the final computations of the shape functions by using these building blocks in combination with a complete set of monomials up to the prescribed polynomial order.

3.3.1 Weight functions

As with FEM bases, the shape functions Φ_I also usually have compact support, with the domain of that support defined by a *weight function* (also called a *kernel* or *window* function) as used in the computation of the shape functions described later in section 3.3.3. For the MLS scheme used in this thesis, the continuity of the shape functions are also solely determined by, and equal to, the continuity of the underlying weight function [128]. Examples of common 1D weight functions are given here (and visualised later in this section in figure 3.2), and include the cubic spline

$$w_{1D}(r) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3, & r \leqslant \frac{1}{2}, \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3, & \frac{1}{2} < r \leqslant 1, \\ 0, & r > 1, \end{cases}, \tag{3.18}$$

the quartic spline

$$w_{1D}(r) = \begin{cases} 1 - 6r^2 + 8r^3 - 3r^4, & r \leqslant 1, \\ 0, & r > 1, \end{cases}, \tag{3.19}$$

or the quintic spline

$$w_{1D}(r) = \begin{cases} \frac{11}{20} - \frac{9}{2}r^2 + \frac{81}{4}r^4 - \frac{81}{4}r^5, & r \leqslant \frac{1}{3}, \\ \frac{17}{40} + \frac{15}{8}r - \frac{63}{4}r^2 + \frac{135}{4}r^3 - \frac{243}{8}r^4 + \frac{81}{5}r^5, & \frac{1}{3} < r \leqslant \frac{2}{3}, \\ \frac{81}{40} - \frac{81}{8}r + \frac{81}{4}r^2 - \frac{81}{4}r^3 + \frac{81}{8}r^4 - \frac{81}{40}r^5, & \frac{2}{3} < r \leqslant 1, \\ 0, & r > 1, \end{cases} \tag{3.20}$$

where r is a normalised measure of distance which is specified in the next section. The main requirements for determining a good kernel function are the continuity of the function itself and the values of the function and its derivatives at $r = 0$ and $r = 1$, in particular that

$$\begin{aligned} w_{1D}(0) > 0, \quad & w'_{1D}(0) = 0, \\ w_{1D}(1) = 0, \quad & w'_{1D}(1) = 0. \end{aligned} \tag{3.21}$$

The function should have continuous derivatives at least up to the order of the desired MLS polynomial basis, and for higher-order continuity, it is desirable to have as many higher-order derivatives go to zero at $r = 1$ as possible to ensure the resultant shape function fades to zero as smoothly as possible at edge of its compact support. Additionally, it is desirable to have as many odd derivatives go to zero at $r = 0$ as possible to ensure the weight function will be as evenly symmetric about the associated node as possible.

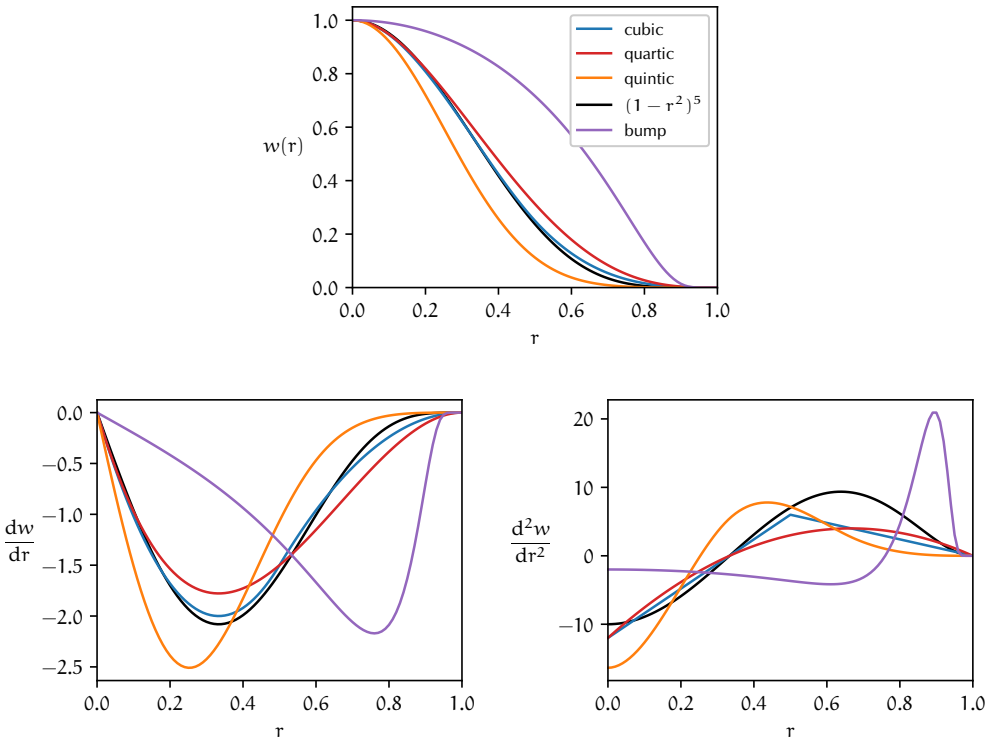


Figure 3.2: Comparison of selected weight functions along with their first and second derivatives.

One simple option for constructing higher-order weight functions from these criteria would be

$$w_{1D}(r) = \begin{cases} (1-r^2)^n, & r \leqslant 1 \\ 0, & r > 1 \end{cases}, \quad \text{for } n \geqslant 2, \tag{3.22}$$

for which all odd derivatives are zero at $r = 0$ and all derivatives of order less than n are zero at $r = 1$. This is not quite as optimal as solving for the true best coefficients of a given order polynomial, but is very simple and quick to implement for achieving arbitrarily high-order continuity. In theory a *bump function* which is infinitely continuous throughout the entire domain, including both $r = 0$ and $r = 1$, such as

$$w_{1D}(r) = \begin{cases} \exp\left(\dfrac{r^2}{r^2-1}\right), & r \leqslant 1 \\ 0, & r > 1 \end{cases}, \tag{3.23}$$

would seem like an interesting choice, but does not seem to be widely used.

Plots of these selected weight functions are shown in figure 3.2 along with their first and second derivatives. It is immediately evident that the bump function is quite distinct, with larger values in all plots much closer to the $r = 1$ end compared to the other functions, giving a qualitative indication as to why it might not be as ideal as pure continuity considerations would suggest. Other obvious features are the monotonicity of all functions, with the first derivatives being entirely negative, and

the first derivatives going to zero on both ends. Conversely, the second derivatives clearly go to zero only at $r = 1$ and not $r = 0$, since only even symmetry at $r = 0$ is required.

The choice of $n = 5$ in the tenth-order generated $w_{1D}(r) = (1 - r^2)^5$ spline was made to give it the same minimum continuity at $r = 1$ as for the the quintic spline, demonstrating both the simplicity of generating a high-order weight function with this method, but also the non-optimality, as only a seventh-order polynomial would truly be needed to match the quintic spline continuity if the coefficients were solved for optimally. We can clearly see in the second derivative plot for these higher-order splines that the third derivatives will also go to zero at both ends, while that is not true for the lower-order cubic and quartic splines. A clear kink in the second derivative of the cubic spline is also visible, compared to the quartic spline which has the same-order continuity at the ends but is infinitely continuous otherwise. Similar kinks would presumably become evident at $r = \frac{1}{3}$ and $r = \frac{2}{3}$ in the quintic spline if higher derivatives were shown, while the tenth-order spline would also remain infinitely continuous there.

3.3.2 Support shapes

Using d_I to denote the size of the shape function's support, which is a free parameter in MLS often referred to as the *dilation parameter* and can vary between nodes, in 1D this gives linear support as

$$w(x - x_I) = w_{1D}\left(\frac{|x - x_I|}{d_I}\right), \tag{3.24}$$

which extends directly to 2D (or 3D) to give circular (spherical in 3D) support as

$$w(\mathbf{x} - \mathbf{x}_I) = w_{1D}\left(\frac{\|\mathbf{x} - \mathbf{x}_I\|}{d_I}\right), \tag{3.25}$$

or rectangular (rectangular box in 3D) support, defined as

$$w(\mathbf{x} - \mathbf{x}_I) = w_{1D}\left(\frac{|x - x_I|}{d_I^x}\right) w_{1D}\left(\frac{|y - y_I|}{d_I^y}\right) w_{1D}\left(\frac{|z - z_I|}{d_I^z}\right). \tag{3.26}$$

An example illustration is given in figure 3.3 to compare between the discretisations of a 2D domain using a FEM scheme and a meshfree scheme with nodes having circular supports.

3.3.3 Shape functions

In order to compute the shape functions, we start from the fundamental idea of the MLS formulation, namely that the approximate solution can be posed as a polynomial with *moving* (i.e. *non-constant*) coefficients, which are themselves functions of position. This means we require a MLS basis $\mathbf{p}$ which is a complete set of monomials up to a

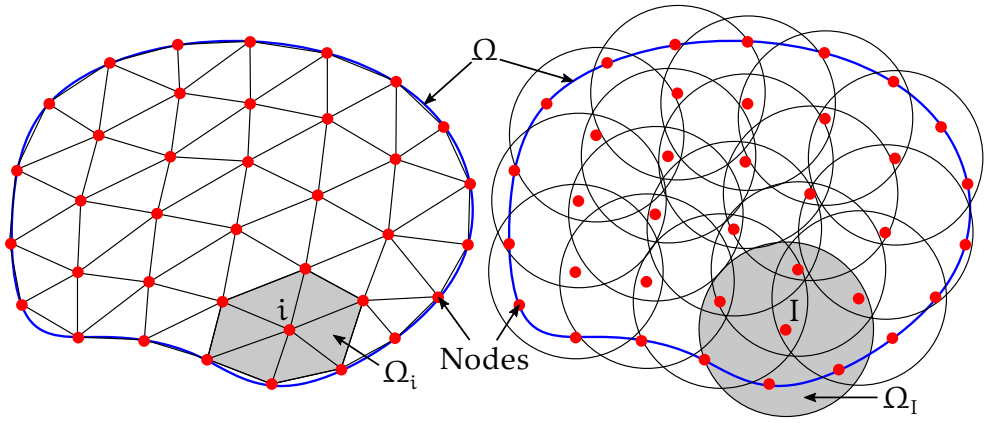


Figure 3.3: Comparative illustration of a standard FEM scheme (left) and a fully meshfree scheme (right) discretising a 2D domain Ω. The meshfree scheme uses nodal domains having compact circular support, shown only for the interior nodes. This figure is adapted from figure 1 in [128].

desired polynomial order m, which for the linear case would be

$$\mathbf{p}^T = \begin{cases} \begin{bmatrix} 1 & x \end{bmatrix}, & 1D \\ \begin{bmatrix} 1 & x & y \end{bmatrix}, & 2D\,, \\ \begin{bmatrix} 1 & x & y & z \end{bmatrix}, & 3D \end{cases} \tag{3.27}$$

and for the quadratic case gives

$$\mathbf{p}^T = \begin{cases} \begin{bmatrix} 1 & x & x^2 \end{bmatrix}, & & 1D \\ \begin{bmatrix} 1 & x & y & x^2 & y^2 & xy \end{bmatrix}, & & 2D\,. \\ \begin{bmatrix} 1 & x & y & z & x^2 & y^2 & z^2 & xy & xz & yz \end{bmatrix}, & & 3D \end{cases} \tag{3.28}$$

The local MLS approximation for a continuous function u evaluated at a point $\mathbf{x} \in \Omega$ nearby to an arbitrary fixed point $\bar{\mathbf{x}} \in \Omega$ can then be written in an alternative form compared to equation (3.17) as

$$u_L^d(\mathbf{x}, \bar{\mathbf{x}}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\bar{\mathbf{x}}), \tag{3.29}$$

where $\mathbf{a}$ is the vector of moving coefficients which are functions of position, as described above. These values are determined by minimising the weighted least squares residual $\mathcal{R}(\bar{\mathbf{x}})$ between this local approximation evaluated at each of the nodes $\mathbf{x}_I$ and the function to be approximated, also evaluated at $\mathbf{x}_I$, giving

$$\mathcal{R}(\bar{\mathbf{x}}) = \sum_{I=1}^{N_n} w(\bar{\mathbf{x}} - \mathbf{x}_I)[u_L^d(\mathbf{x}_I, \bar{\mathbf{x}}) - u(\mathbf{x}_I)]^2. \tag{3.30}$$

Different choices with regards to $\bar{\mathbf{x}}$ give rise to slightly different numerical schemes such as the least squares (LS) (setting $w(\bar{\mathbf{x}} - \mathbf{x}_I) = 1$) and weighted least squares (WLS) (setting $\bar{\mathbf{x}} = \mathbf{x}_I$) approximations, for which the interested reader can

consult e.g. [125] for more details. For this thesis we restrict ourselves to the MLS scheme where $\mathbf{a}$ is a function of position, and this is obtained by letting $\bar{\mathbf{x}} \to \mathbf{x}$, thus giving the residual to be minimised as

$$\mathcal{R}(\mathbf{x}) = \sum_{I=1}^{N_n} w(\mathbf{x} - \mathbf{x}_I)[\mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - u(\mathbf{x}_I)]^2. \tag{3.31}$$

One can see that by including the complete set of monomials in $\mathbf{p}$, if the function $u(\mathbf{x})$ is itself a polynomial of maximally order m then the approximation can be made exact (and the residual minimised to zero) simply by setting $\mathbf{a}$ to a constant vector of the polynomial coefficients in u. Thus the shape functions derived from minimising this residual will always be able to exactly represent polynomials up to the order m used to define $\mathbf{p}$.

For general approximations, extrema of the residual function can be found by solving for the roots of the derivative of $\mathcal{R}$ with respect to $\mathbf{a}(\mathbf{x})$, which gives the matrix equation

$$\sum_{I=1}^{N_n} w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) = \sum_{I=1}^{N_n} w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)u(\mathbf{x}_I). \tag{3.32}$$

A matrix $\mathbf{A}(\mathbf{x})$ is then introduced to simplify the notation. It is often referred to as the *moment matrix* and is defined as

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{N_n} w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I), \tag{3.33}$$

thus allowing the coefficients $\mathbf{a}(\mathbf{x})$ to be solved for as

$$\mathbf{a}(\mathbf{x}) = \sum_{I=1}^{N_n} [\mathbf{A}(\mathbf{x})]^{-1} w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)u(\mathbf{x}_I). \tag{3.34}$$

This can then be substituted into the local approximation posed in equation (3.29) and related back to the original MLS expansion given in equation (3.17) to give

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \sum_{I=1}^{N_n} [\mathbf{A}(\mathbf{x})]^{-1} w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)u(\mathbf{x}_I) = \sum_{I=1}^{N_n} u_I \Phi_I(\mathbf{x}), \tag{3.35}$$

and inspecting the two forms of the approximation, one can see that on the RHS we need a shape function associated to each node that is a function of position which is multiplied by a coefficient for each node. Thus by comparing to the middle expression one finds the shape functions are given by

$$\Phi_I(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})[\mathbf{A}(\mathbf{x})]^{-1} w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I). \tag{3.36}$$

When starting from a known function to be approximated, the coefficients of the shape functions are found in equation (3.35) to be the value of this known function evaluated at the associated node, $u(\mathbf{x}_I)$. However, we can see that the shape functions themselves do not depend on $u(\mathbf{x})$ in any way, but rather they depend only on the choices of $w(\mathbf{x})$ and $\mathbf{p}(\mathbf{x})$, and the set of nodes $\mathbf{x}_I$. Thus, even though we began by

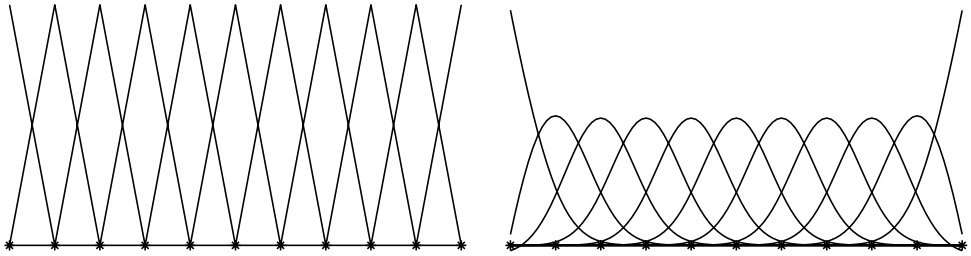


Figure 3.4: Illustration of 1D shape functions with $d/h \approx 1$ (similar to a FEM) and $d/h = 2.6$, with cubic spline weight function and linear consistency, where h is the distance between the uniformly spaced nodes. This figure is taken from figure 5 in [129].

posing the computation of $\mathbf{a}(\mathbf{x})$ as minimising the weighted least squares residual from $u(\mathbf{x})$, it turns out that the optimal set of shape functions for minimising this residual are actually independent of the specific function to be approximated. The procedure has therefore produced a set of basis functions for the given nodal arrangement which can be used to approximate other, unknown, continuous functions, where the u_I coefficients would then be a set of unknowns to be solved for, analogously to a FEM.

Unlike in a FEM, however, the Φ_I need not have a value of unity at the associated node $\mathbf{x}_I$ itself, and multiple shape functions will have non-zero value at any given point in the domain, including at the nodes. This results in the basis function coefficients for an unknown function *not* necessarily being equal to the value of the numerical solution function evaluated at the respective nodal point, i.e. $u(\mathbf{x}_I, t) \neq u_I(t)$ in general, and equation (3.17) must be used to compute the value of the numerical solution even at the node points. This also complicates the imposition of Dirichlet boundary conditions (BCs) compared to a FEM. An illustration in 1D of what these shape functions look like with linear $\mathbf{p}(\mathbf{x})$ is given in figure 3.4, where one can see the transition from essentially FEM-type interpolant basis functions to highly overlapping approximant shape functions as the size of the support d is increased compared to the node spacing.

Note that the moment matrix $\mathbf{A}(\mathbf{x})$ must be inverted at every point at which the shape function is to be evaluated. As this matrix is $n \times n$, where n is the number of functions present in the MLS basis $\mathbf{p}(\mathbf{x})$ of polynomial order m, one may show that this means that every point in the domain must be within the support of at least as many shape functions as there are functions in the basis (i.e. within the support of at least n shape functions) in order for $\mathbf{A}(\mathbf{x})$ to be invertible.

3.3.4 Galerkin and collocation assembly

The same Galerkin approach as defined for a FEM in section 3.1 can then be used, simply replacing the FEM basis functions ϕ_i with the MLS shape functions Φ_I in equations (3.3) to (3.7). This approach is often termed the *element-free Galerkin (EFG)*

method [130, 131]. The only major difference comes in evaluating the integrals in the inner products, since there are now no mesh elements over which to perform quadrature, and the shape functions are non-polynomial (because of the non-constant coefficients) so an exact quadrature is no longer possible. Quadrature is therefore discussed further in section 4.3.

An alternative approach known as the *collocation method* can be formulated via the same weighted residual procedure, where instead of using the shape functions Φ_I as the test functions in equation (3.3) one uses the Dirac delta functions $\delta_I = \delta(\mathbf{x} - \mathbf{x}_I)$ evaluated at the nodal positions, giving

$$\left\langle \frac{\partial u}{\partial t} \,\middle|\, \delta_I \right\rangle = \left\langle \mathcal{L}(u) \,\middle|\, \delta_I \right\rangle + \left\langle f \,\middle|\, \delta_I \right\rangle, \tag{3.37a}$$

$$\left\langle \frac{\partial}{\partial t} \left(\sum_{J=1}^{N_n} u_J(t)\Phi_J(\mathbf{x}) \right) \,\middle|\, \delta_I(\mathbf{x}) \right\rangle = \left\langle \mathcal{L} \left(\sum_{J=1}^{N_n} u_J(t)\Phi_J(\mathbf{x}) \right) \,\middle|\, \delta_I(\mathbf{x}) \right\rangle + \left\langle f \,\middle|\, \delta_I(\mathbf{x}) \right\rangle, \tag{3.37b}$$

$$\sum_{J=1}^{N_n} \frac{du_J}{dt} \left\langle \Phi_J \,\middle|\, \delta_I \right\rangle = \sum_{J=1}^{N_n} u_J \left\langle \mathcal{L}(\Phi_J) \,\middle|\, \delta_I \right\rangle + \left\langle f \,\middle|\, \delta_I \right\rangle. \tag{3.37c}$$

The sifting property of the Dirac delta function can then be used to simplify the integral inner products to merely evaluations at the nodal points, giving

$$\sum_{J=1}^{N_n} \frac{du_J}{dt} \Phi_J(\mathbf{x}_I) = \sum_{J=1}^{N_n} u_J \, \mathcal{L}(\Phi_J)\Big|_{\mathbf{x}=\mathbf{x}_I} + f(\mathbf{x}_I), \tag{3.38a}$$

$$\mathbf{M}\dot{\mathbf{u}} = \mathbf{K}\mathbf{u} + \mathbf{f}, \tag{3.38b}$$

where

$$\mathbf{M}_{IJ} = \Phi_J(\mathbf{x}_I), \qquad \mathbf{K}_{IJ} = \mathcal{L}(\Phi_J)\Big|_{\mathbf{x}=\mathbf{x}_I}, \qquad \mathbf{f}_I = f(\mathbf{x}_I). \tag{3.39}$$

One can see that unlike the Galerkin method, collocation does not require any numerical quadrature as there are no longer any integrals to be evaluated. This greatly simplifies the procedure for computing the linear system, but the trade-off is that higher-order derivatives of the shape functions generally have to be computed, as integration by parts can otherwise be used to decrease the order of the differential operator when computing the entries of $\mathbf{K}_{IJ}$ in equation (3.7) of the Galerkin method, as was demonstrated in section 3.1.2.

Additionally, the Galerkin method can be viewed as the projection of the continuous function onto the finite dimensional function space defined by the shape functions. This means that the residual error is orthogonal to all of the shape functions according to inner product used to define the weak form and is therefore optimal in the sense of minimising the least squares error for the L^2 inner product used here. This means that both the absolute error is larger and the order of convergence generally lower for collocation as compared to the Galerkin method [132], with higher continuity weight functions also often required to achieve optimal-order convergence [125], further increasing computational cost.

3.4 Meshfree drawbacks

The lack of required mesh is the main advantage to the EFG, as it also allows for easy refinement or changing of node positions without any remeshing; however, it also introduces several drawbacks that lead us to initially seek a merely partially meshfree solution, namely:

- Computing shape functions and their derivatives is much more expensive, due to inversion of the *moment matrix* and having to search for all nodes with non-zero support at every evaluation point.
- The stability and accuracy of the solution are strongly dependent on the *choice of parameters* such as support size/shape and weight function, necessitating substantial user knowledge and intervention.
- MLS bases do not have the *Kronecker delta property*, which complicates imposition of Dirichlet BCs.

One previously mentioned drawback which we must still contend with even in a partially meshfree scheme is that exact quadrature is not possible due to the lack of defined element areas and shape functions not being polynomial in general. This is addressed further in section 4.3.

CHAPTER 4

FCIFEM SCHEME

One of the global coordinate directions is chosen to parameterise the direction parallel to the anisotropy, and is labelled ζ. We thus require that ζ be a good (one-to-one) parameter for these mapping lines, but the normals of the constant-ζ surfaces need not be parallel to the mapping direction. Practically, it often happens that one of the global coordinates has this property, e.g. in most MCF devices the toroidal field is non-zero everywhere and so the toroidal angle φ is a good choice.

The interior nodes of the discretisation are organised to lie on planes of constant ζ, referred to as FCI planes, with fine enough spacing of nodes within each plane to resolve the fast variations perpendicular to the anisotropy. Successive planes are then spaced comparatively far apart, as this spacing need only resolve the slow variation parallel to the anisotropy. On each of the planes, as well as on the domain boundary, standard FEM meshes are used, which are essentially one dimension lower than the overall simulation domain. The nodes located at the intersection of the FCI planes and domain boundaries will thus be associated with the FEM basis functions of the neighbouring elements on both surfaces.

The local direction of the anisotropy is represented by defining a mapping function $Q : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, where n is the spatial dimension of the full domain. This function takes as arguments a starting point and final parallel coordinate value ζ, and returns the point on the same mapping line as the starting point having the requested ζ-coordinate. In general, this mapping is likely to be approximated by a local, low-order polynomial expansion for ease of computation.

For the purposes of this thesis we consider the actual determination of this mapping function to be outside our scope; here we require only that it exists and provides the necessary one-to-one mapping throughout the domain. In practice the mapping would be determined by solving the Grad-Shafranov equation [133–135] for the magnetic field associated with the MHD equilibrium of the desired MCF

device geometry. This is a relatively well-studied problem and existing codes such as CHEASE (Cubic Hermite Element Axisymmetric Static Equilibrium) [136, 137] could be used to furnish a solution.

4.1 Basis function definition

For a given evaluation point $\mathbf{x} = \left[\zeta, \mathbf{x}_\perp\right]^\intercal$ we define the mapped point $\mathbf{x}_{\text{map}} = Q(\mathbf{x}, \zeta_i)$ where the mapping does not intersect the boundary before the next FCI plane. Otherwise $\mathbf{x}_{\text{map}}$ will be such an intersection point as discussed in section 4.1.2. Regardless, we decompose our basis functions as the product of two functions

$$\phi_i = \phi_{i,\text{FEM}}(\mathbf{x}_{\text{map}})\rho_i(\mathbf{x}). \tag{4.1}$$

The first functions $\phi_{i,\text{FEM}}$ are standard FEM basis functions of one dimension lower than the global simulation dimension defined on the standard FEM meshes covering the FCI planes and domain boundaries. In our simulations we use linear Lagrange polynomials.

The second functions ρ_i are weighting functions that must be continuous and differentiable, and ideally possess the Kronecker delta property discussed in section 4.1.1. For our tests we define them as linear ramp functions along any given mapping line such that they will have a value of unity on the FCI plane or domain boundary on which the corresponding node is located (which could include both if the node is located at the intersecting edge of a FCI plane with the domain boundary) and then ramp linearly to zero as a function of ζ along the mapping away from the surface, reaching zero when the next FCI plane or boundary is reached. This gives

$$\rho_i(\mathbf{x}) = \frac{\zeta - \zeta_t}{\zeta_o - \zeta_t}, \tag{4.2}$$

where ζ_o and ζ_t are the ζ-coordinates of the points $\mathbf{x}_-$ and $\mathbf{x}_+$ found by mapping from $\mathbf{x}$ in both the negative and positive ζ-coordinate directions until a FCI plane or boundary is encountered. ζ_o is from the *origin* point on the FCI plane or boundary containing $\mathbf{x}_i$ whilst ζ_t is from the other *terminus* point.

This scheme is illustrated in figure 4.1 for two different points $\mathbf{x}_1$ and $\mathbf{x}_2$. Mapping in either direction from $\mathbf{x}_1$ does not encounter the domain boundary, but only the two FCI planes enclosing the region of the domain where the point resides. Thus it is the basis functions associated with nodes $a, b, c,$ and d which will contain $\mathbf{x}_1$ within their compact support, and all other basis functions will be exactly zero there. To compute the value of these basis functions at $\mathbf{x}_1$, we will then have $\zeta_o = \zeta_k$, $\zeta_t = \zeta_{k+1}$, and $\mathbf{x}_{\text{map}} = \mathbf{x}_{L1}$ for ϕ_a and ϕ_b whilst $\zeta_o = \zeta_{k+1}$, $\zeta_t = \zeta_k$, and $\mathbf{x}_{\text{map}} = \mathbf{x}_{R1}$ for ϕ_c and ϕ_d.

For $\mathbf{x}_2$, however, we encounter the domain boundary when mapping in the positive parallel direction, and so it is the basis functions associated with nodes $e, f, g,$ and h which will be non-zero at $\mathbf{x}_2$. We will then have $\zeta_o = \zeta_k$, $\zeta_t = \zeta_{R2}$, and $\mathbf{x}_{\text{map}} = \mathbf{x}_{L2}$ for ϕ_e and ϕ_f whilst $\zeta_o = \zeta_{R2}$, $\zeta_t = \zeta_k$, and $\mathbf{x}_{\text{map}} = \mathbf{x}_{R2}$ for ϕ_g and ϕ_h.

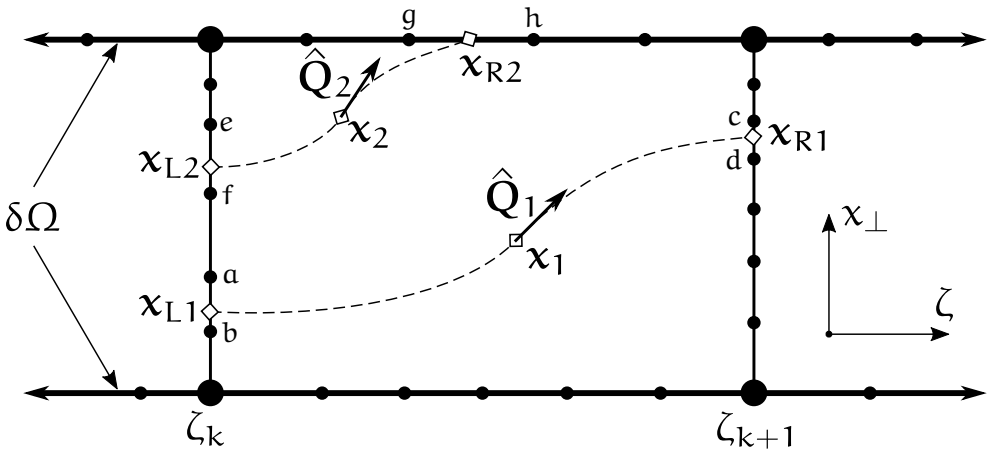


Figure 4.1: Illustration of the FCIFEM scheme discretising a 2D domain Ω. Evaluation points are mapped to the surrounding meshes on FCI planes and boundaries via a mapping function $Q : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, e.g. $Q(\mathbf{x}_1, \zeta_k) = \mathbf{x}_{L1}$.

4.1.1 Basis properties

The basis functions ϕ_i defined as above have *compact support*, meaning that they are non-zero only in a small portion of the domain nearby the node to which they are associated. In a standard FEM scheme this would often be one layer of mesh elements surrounding the node. In the FCIFEM scheme this manifests as having support extending only as far as one neighbouring plane in the parallel direction and only over a small fraction of the perpendicular coordinate space corresponding to one layer of elements of the mesh on which the corresponding node resides.

Crucially, the basis functions form a *partition of unity*, i.e. when we evaluate the FCIFEM basis function expansion with all coefficients set to unity, we get

$$u^h(\mathbf{x}, t) = \sum_{i=1}^{N_n} \phi_i(\mathbf{x}) = 1 \qquad \forall \mathbf{x} \in \Omega. \tag{4.3}$$

This follows from the use of standard FEM basis functions $\phi_{i,\mathrm{FEM}}$ on the FCI planes and domain boundaries, which are well-known to satisfy the partition of unity. Evaluating at the point $\mathbf{x}$, we then have the mapped points $\mathbf{x}_-$ and $\mathbf{x}_+$ which exist on these conventional FEM surfaces, so evaluating at either mapped point yields unity. The ρ_i functions are then defined to ramp linearly in the mapping variable between these surfaces, ensuring that the full basis definition is also a partition of unity. This property is important for consistency of the method and enforcement of numerical conservation for simulations involving conservation laws, as discussed later in section 4.2.

Lastly, this definition of ϕ_i gives a value of unity at node $\mathbf{x}_i$ itself and a value of zero at every other node, including nodes on the edge of the region of its compact support. This results in the basis function coefficients being equal to the value of the numerical solution function evaluated at the respective nodal point, i.e. $u(\mathbf{x}_i, t) =$

$u_i(t)$ for all nodes $\mathbf{x}_i$. This means that our basis does indeed possess the Kronecker delta property and simplifies the enforcement of essential BCs.

4.1.2 Boundary nodes

Consider the simulation boundary to be the locus of points $S(\mathbf{x}) = 0$ for some function S. A point $\mathbf{x}$ will map to the boundary iff $\exists \zeta_b \, S(Q(\mathbf{x}, \zeta_b)) = 0$, and we will denote this associated boundary point $\mathbf{x}_b = Q(\mathbf{x}, \zeta_b)$. If such a point exists, then $\mathbf{x}_{map} = \mathbf{x}_b$ for computing the basis functions of the associated boundary nodes. Mappings that 'brush' against the boundary before re-entering the domain (i.e. have an extremum tangential to the boundary) would generally not be allowed as this would result in discontinuities in the associated basis functions as $\mathbf{x}_{map}$ jumps suddenly to the boundary at the extremum.

While not strictly necessary for the overall method, computation of boundary terms can be greatly simplified if the forms of Q and S allow one to define a boundary function $\mathbf{x}_b = B(\mathbf{x})$ which directly returns the mapped point on the boundary associated with any point $\mathbf{x}$ for which ζ_b exists. This function could have multiple solutions in general if multiple such ζ_b exist for a given $\mathbf{x}$, but for the method to be well-behaved at the boundary we will assume that either only one such ζ_b will exist between any two neighbouring FCI planes or the nearest intersection point to the evaluation point will be used.

To illustrate this in figure 4.1, there are no solutions to $B(\mathbf{x}_1)$ between ζ_k and ζ_{k+1}, whereas the solution $\mathbf{x}_b = \mathbf{x}_{R2} = B(\mathbf{x}_2)$ does exist in the interval. For general functions, solving for intersections can be a difficult problem, but in most practical simulations the boundary is anyway represented by a piecewise linear (or at least polynomial) discretisation and the mapping would likely be a quadratic expansion, making the intersection computation relatively straightforward.

4.1.3 Computing gradients of the basis functions

Gradients in the global coordinate system must be computed using the chain rule, as differential changes in the evaluation point will result also in differential changes of the mapping points mediated by the mapping function. This gives the general form of the gradient as

$$\nabla\phi_i(\mathbf{x}) = \rho_i(\mathbf{x})\nabla\phi_{i,FEM}(\mathbf{x}_{map}) \cdot \nabla Q(\mathbf{x}, \zeta_i) + \phi_{i,FEM}(\mathbf{x}_{map})\nabla\rho_i(\mathbf{x}). \tag{4.4}$$

One especially easy derivative to compute is that directed along the mapping itself, denoted $\nabla_\parallel$, because movement along the mapping direction does not change any of the mapped points and so the only change is due to the slope of the $\rho_i(\mathbf{x})$ function. This means the first term in equation (4.4) vanishes completely, and only the ζ component of $\nabla\rho_i(\mathbf{x})$ remains in the second term. Letting $\hat{\mathbf{Q}}$ denote the unit vector in the mapping direction at the evaluation point as shown in figure 4.1, then

we have

$$\nabla_{\parallel}\phi_i(\mathbf{x}) = \phi_{i,\mathrm{FEM}}(\mathbf{x}_{\mathrm{map}})\frac{\partial \rho_i(\mathbf{x})}{\partial \zeta}\hat{\mathbf{Q}}. \tag{4.5}$$

This would make the scheme particularly attractive for problems involving such directed derivatives, such as those with anisotropic diffusion or advection terms directed along known fields.

4.2 Local flux conservation

Hughes et al. [138] demonstrated an interesting property of continuous Galerkin schemes on standard FEM meshes, namely that they are locally conservative with respect to element nodal fluxes. However, their derivation relies on exact quadrature being possible in a standard FEM, which is not true for meshfree schemes. We therefore wish to extend their conservation argument to show that a similar local point-wise conservation property can be achieved for any Galerkin scheme, regardless of the choice of quadrature scheme and the non-polynomial nature of general meshfree shape functions.

To start, for a conservation law of a conserved quantity u, we define $\boldsymbol{\sigma}(u)$ as the total conservative flux of u to give a general PDE of the form

$$\frac{\partial u}{\partial t} = -\nabla \cdot \boldsymbol{\sigma}(u). \tag{4.6}$$

One could also include a prescribed forcing term f, but as this is simply an imposed source/sink term and therefore by nature non-conservative, we will omit it to focus on the conservation properties arising from the discretisation of the differential operators. We use also the L^2 inner product defined in equation (3.4) and select a set of test functions ϕ_i which must form a partition of unity, as is the case for our FCIFEM basis functions, such that

$$\sum_{i=1}^{N_n} \phi_i = 1 \quad \Longrightarrow \quad \sum_{i=1}^{N_n} \nabla\phi_i = 0. \tag{4.7}$$

We can then take the inner product of equation (4.6) with the test functions to get the weak form

$$\int_\Omega \frac{\partial u}{\partial t}\phi_i \,\mathrm{d}\Omega = \int_\Omega \nabla\phi_i \cdot \boldsymbol{\sigma}(u)\,\mathrm{d}\Omega - \int_\Gamma \phi_i \boldsymbol{\sigma}(u)\cdot\hat{\mathbf{n}}\,\mathrm{d}\Gamma, \tag{4.8}$$

where integration by parts has been used to transform the integration of the conservative differential operator to include an integral over the boundary $\Gamma = \delta\Omega$ of the domain, with $\hat{\mathbf{n}}$ the outward normal unit vector on Γ. For any problem with only periodic or homogeneous Neumann BCs the boundary integral will disappear, and the case of non-homogeneous Neumann BCs would again be simply an imposed source/sink term which we will again ignore. For the case of Dirichlet BCs, unlike in the interior where we do not have well-defined elements on which to write local conservation laws with respect to fluxes over element boundaries, comparable to a

FVM or DG scheme, we do still have well-defined global domain boundaries. Therefore, the procedure of [138] for using a *postprocessing* step to compute the conservative fluxes through the essential boundaries should be also directly applicable in the meshfree case.

On a computer, the integrals over Ω will be computed with a numerical quadrature over a set of points with coordinates $\mathbf{x}_q$ and weights w_q. At any single quadrature point $\mathbf{x}_q$ one can then sum the contributions over the full set of test functions to get

$$\sum_{i=1}^{N_n} w_q \left.\frac{\partial u}{\partial t}\phi_i\right|_{\mathbf{x}_q} = \sum_{i=1}^{N_n} w_q \left.\boldsymbol{\nabla}\phi_i \cdot \boldsymbol{\sigma}(u)\right|_{\mathbf{x}_q}, \tag{4.9a}$$

$$w_q \left.\frac{\partial u}{\partial t}\sum_{i=1}^{N_n}\phi_i\right|_{\mathbf{x}_q} = w_q \left.\boldsymbol{\nabla}\left(\sum_{i=1}^{N_n}\phi_i\right)\cdot\boldsymbol{\sigma}(u)\right|_{\mathbf{x}_q}, \tag{4.9b}$$

$$\left.\frac{\partial u}{\partial t}\right|_{\mathbf{x}_q} = 0, \tag{4.9c}$$

where we have used the partition of unity property given in equation (4.7) to show that the RHS is always exactly zero. Analogous to equation (83) in [138] we can denote $f_q^i \equiv w_q \left.\boldsymbol{\nabla}\phi_i \cdot \boldsymbol{\sigma}(u)\right|_{\mathbf{x}_q}$ as the point-wise contribution of test function ϕ_i to the flux at point q. Having $\sum_i f_q^i = 0$ confirms that positive and negative contributions to the flux must exactly balance, which is a statement of local conservation of the point-wise fluxes at every point q, analogous to equation (88) in [138]. Crucially, this balance arises independently of $\mathbf{x}_q$ and w_q and so holds for any choice of quadrature scheme, and is unaffected by error in the resulting numerical integration.

It is important to note that this result only arises in the case where the general operator is first transformed via integration by parts such that we have the gradient operator acting on the test functions ϕ_i. This is required as the sum of contributions over the test functions is then equivalent to taking the gradient of a summed partition of unity, ensuring the result will always be exactly zero irrespective of the quadrature scheme used to evaluate the integrals in the inner product.

4.2.1 Global conservation

To demonstrate global conservation we can sum over the test functions in the weak form of the conservation law given in equation (4.8) to get

$$\sum_{i=1}^{N_n}\int_\Omega \frac{\partial u}{\partial t}\phi_i \,d\Omega = \sum_{i=1}^{N_n}\int_\Omega \boldsymbol{\nabla}\phi_i\cdot\boldsymbol{\sigma}(u)\,d\Omega - \sum_{i=1}^{N_n}\int_\Gamma \phi_i\boldsymbol{\sigma}(u)\cdot\hat{\mathbf{n}}\,d\Gamma, \tag{4.10a}$$

$$\int_\Omega \frac{\partial u}{\partial t}\sum_{i=1}^{N_n}\phi_i \,d\Omega = \int_\Omega \boldsymbol{\nabla}\left(\sum_{i=1}^{N_n}\phi_i\right)\cdot\boldsymbol{\sigma}(u)\,d\Omega - \int_\Gamma \sum_{i=1}^{N_n}\phi_i\boldsymbol{\sigma}(u)\cdot\hat{\mathbf{n}}\,d\Gamma, \tag{4.10b}$$

$$\int_\Omega \frac{\partial u}{\partial t}\,d\Omega = 0 - \int_\Gamma \boldsymbol{\sigma}(u)\cdot\hat{\mathbf{n}}\,d\Gamma, \tag{4.10c}$$

which is equivalent to the approach of [138] in setting the test functions (which they refer to as *weighting functions*) to unity in equation (4.8) because of the partition of unity property, which has also been used to show that the middle term evaluates to

zero. It must be noted that again as in the previous local section, this result therefore only holds in the case where the full boundary value problem (BVP) does not have any Dirichlet BCs imposed on it, so Γ consists entirely of Neumann BCs (or has measure zero in the case of periodic BCs), otherwise the postprocessing procedure of [138] would be needed to compute the consistent conservative fluxes through the essential boundaries.

We then expand u in the chosen trial function basis according to equation (3.1) and introduce the numerical quadrature to compute the integrals, now also including a set of quadrature points along the boundary Γ denoted by q' to give

$$\sum_{q=1}^{N_q} w_q \frac{\partial}{\partial t}\left(\sum_{j=1}^{N_n} u_j(t)\phi_j(\mathbf{x}_q)\right) = -\sum_{q'=1}^{N_{q'}} w_{q'}\, \boldsymbol{\sigma}\left(\sum_{j=1}^{N_n} u_j(t)\phi_j(\mathbf{x})\right)\cdot\hat{\mathbf{n}}\Big|_{\mathbf{x}_{q'}}, \tag{4.11a}$$

$$\sum_{j=1}^{N_n} \frac{\partial u_j}{\partial t}\sum_{q=1}^{N_q} w_q\phi_j(\mathbf{x}_q) = -\sum_{j=1}^{N_n} u_j \sum_{q'=1}^{N_{q'}} w_{q'}\, \boldsymbol{\sigma}(\phi_j)\cdot\hat{\mathbf{n}}\Big|_{\mathbf{x}_{q'}}, \tag{4.11b}$$

$$\sum_{j=1}^{N_n} \frac{\partial u_j}{\partial t} m_j = -\sum_{j=1}^{N_n} u_j \sum_{q'=1}^{N_{q'}} w_{q'}\, \boldsymbol{\sigma}(\phi_j)\cdot\hat{\mathbf{n}}\Big|_{\mathbf{x}_{q'}}, \tag{4.11c}$$

where the global *mass* m_j associated with each node is given by

$$m_j = \sum_{q=1}^{N_q} w_q\phi_j(\mathbf{x}_q). \tag{4.12}$$

Equation (4.11c) then clearly represents the summed time-rates of change of the mass-weighted nodal values on the LHS balanced against the prescribed Neumann boundary fluxes, which are essentially just a prescribed external source/sink for the conserved quantity, meaning the discretisation of the conservative operator itself does not induce any change in the global summation.

4.2.2 Matrix view on global conservation

We can extend the definition of the L^2 inner product in equation (3.4) for vector-valued functions as

$$\left\langle \mathbf{g} \,\middle|\, \mathbf{h} \right\rangle_\Omega = \int_\Omega \mathbf{g}\cdot\mathbf{h}\,\mathrm{d}\Omega, \tag{4.13}$$

where the subscript now also indicates the domain over which the integral is evaluated. We can then apply integration by parts to the matrix element definition for the operator matrix $\mathbf{K}$ given in equation (3.7) for a conservative differential operator to get

$$\mathbf{K}_{ij} = \left\langle \mathcal{L}(\phi_j) \,\middle|\, \phi_i \right\rangle_\Omega, \tag{4.14a}$$

$$= \left\langle -\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}(\phi_j) \,\middle|\, \phi_i \right\rangle_\Omega, \tag{4.14b}$$

$$= \left\langle \boldsymbol{\sigma}(\phi_j) \,\middle|\, \boldsymbol{\nabla}\phi_i \right\rangle_\Omega - \left\langle \boldsymbol{\sigma}(\phi_j)\cdot\hat{\mathbf{n}} \,\middle|\, \phi_i \right\rangle_\Gamma. \tag{4.14c}$$

Summing over the test functions as we did in section 4.2.1 is then equivalent to summing down the columns of the numerically computed operator matrix, and

we have thus shown that these column sums will be identically zero (to machine precision and ignoring the inherently non-conservative boundary integral) no matter the numerical quadrature used. This means that for *any* vector $\mathbf{u}$ which is multiplied by $\mathbf{K}$ the elements of the resulting vector will sum to zero

$$\sum_{i=1}^{N_n} (\mathbf{K}\mathbf{u})_i = 0 \quad \forall \mathbf{u}, \tag{4.15}$$

providing an alternative demonstration that such an operator is globally conservative in the sense that it produces no net contribution to the summed rates of change of the mass-weighted nodal values of the approximate solution of the conserved field.

4.3 Quadrature

As mentioned in the introduction, achieving accurate quadrature is a well-known challenge for meshfree methods and has been extensively studied by that community [139–149]. In particular, the non-polynomial nature of the basis functions and arbitrary spatial overlap of the basis function supports make exact quadrature impossible.

This issue is intimately related to the properties of polynomial consistency that the MLS formulation is constructed to uphold. In section 3.3 we showed how the MLS construction guarantees that the shape functions can exactly represent polynomials of a desired order despite the arbitrary node placement and overlap of shape function supports. Here we address the analogous idea of imposing similar consistency conditions on the quadrature used to discretise PDEs with these shape functions to ensure the expected order of accuracy is achieved.

A common approach to numerical integration in meshfree schemes is to use a very simple background mesh which fully covers the domain and uses a uniform or Gauss-Legendre distributed set of quadrature points on each cell of this mesh; however, this usually requires a large numbers of quadrature points per cell to obtain good solutions. We therefore focus on the work of Chen et al. [149], who developed variationally consistent integration (VCI) schemes and found this to be an effective tool for improving the accuracy of the method back towards the optimal rate of convergence with less computational effort.

Essentially, in their work they derive the constraints with which a quadrature scheme must conform in order to achieve a given order of accuracy in an overall simulation. For example, the first-order (linear) constraints for a Poisson problem are that the quadrature exactly (i.e. to machine precision) integrates the following equality (often referred to as a numerical *divergence condition*, despite not actually involving a divergence)

$$\langle \nabla \phi_i \rangle_\Omega = \left\langle \phi_i \hat{\mathbf{n}} \right\rangle_{\delta\Omega} \qquad \forall i, \tag{4.16}$$

where $\langle\cdot\rangle_\Omega = \sum_{q=1}^{N_q}(\cdot)w_q$ and $\langle\cdot\rangle_{\delta\Omega} = \sum_{q'=1}^{N_{q'}}(\cdot)w_{q'}$ follow Chen's notation to denote the numerical domain and boundary integrals, respectively, and $\hat{\mathbf{n}}$ is the outward facing unit normal vector of the boundary. The second-order (quadratic) constraints in 2D are given by

$$\left\langle \phi_{i,x} x \right\rangle_\Omega = -\langle \phi_i \rangle_\Omega + \langle \phi_i x n_1 \rangle_{\delta\Omega} \qquad \forall i, \tag{4.17a}$$

$$\left\langle \phi_{i,y} y \right\rangle_\Omega = -\langle \phi_i \rangle_\Omega + \langle \phi_i y n_2 \rangle_{\delta\Omega} \qquad \forall i, \tag{4.17b}$$

$$\left\langle \phi_{i,x} y + \phi_{i,y} x \right\rangle_\Omega = \langle \phi_i (y n_1 + x n_2) \rangle_{\delta\Omega} \qquad \forall i, \tag{4.17c}$$

where $\nabla\phi_i = \begin{bmatrix}\phi_{i,x},\ \phi_{i,y}\end{bmatrix}^T$ and $\hat{\mathbf{n}} = \begin{bmatrix}n_1,\ n_2\end{bmatrix}^T$. All lower-order constraints must simultaneously be met when these higher-order constraints are considered.

It is important to note that the ϕ_i functions in these constraint equations refer to the test functions, which for a Bubnov-Galerkin scheme will be the same as the set of basis functions forming the trial function space. However, in [149] a Petrov-Galerkin approach is proposed for enforcing the integration constraints by computing a set of correction terms that is added to each of the corresponding test functions, which is termed the *assumed strain method*. Their specific formulation has the advantage of decoupling the correction terms such that each can be computed entirely independently of the others, but it has the disadvantage that the test functions no longer form a partition of unity. As shown in section 4.2, this renders the method no longer conservative.

4.3.1 Maintaining conservation

We therefore propose to instead enforce the integration constraints by computing a correction to the weights of the quadrature points, keeping the test functions as a partition of unity and the overall method Bubnov-Galerkin. This preserves the conservation but with the drawback that the corrections now form a coupled linear system that must be solved. Therefore, for simulations where conservation is not necessary, such as those not involving conservation laws or for stationary problems, it may still be sensible to first try the assumed strain formulation to see if it already gives satisfactory results.

This approach involves looping once over each quadrature point to compute the starting values of all terms in the integration constraints of interest. These contributions are then used to construct a linear system of equations where the unknowns to be solved for are the corrections ξ_q to the quadrature weights w_q which will result in the constraints being met to machine precision. As an example, the construction of the matrix equation for the linear constraints of equation (4.16) is

shown here

$$\sum_{q=1}^{N_q} \nabla\phi_i(w_q + \xi_q)\Big|_{x_q} = \sum_{q'=1}^{N_{q'}} \phi_i(x_{q'})\hat{\mathbf{n}} w_{q'} \qquad \forall i, \tag{4.18a}$$

$$\sum_{q=1}^{N_q} \nabla\phi_i \xi_q\Big|_{x_q} = \sum_{q'=1}^{N_{q'}} \phi_i(x_{q'})\hat{\mathbf{n}} w_{q'} - \sum_{q=1}^{N_q} \nabla\phi_i w_q\Big|_{x_q} \qquad \forall i, \tag{4.18b}$$

$$\mathbf{G}\boldsymbol{\xi} = \mathbf{r}, \tag{4.18c}$$

where $\mathbf{G}$ is an $N_c \times N_q$ matrix, with the associated vectors $\boldsymbol{\xi}$ and $\mathbf{r}$ then requiring N_q and N_c entries, respectively. For this linear case there will be two constraints per basis function, giving $N_c = 2N_n$ total constraints. The matrix entries are computed as

$$\mathbf{G}_{i,q} = \phi_{i,x}(x_q), \qquad \mathbf{G}_{i+N_n,q} = \phi_{i,y}(x_q), \tag{4.19}$$

and the RHS residual vector entries are computed as

$$\mathbf{r}_i = \sum_{q'=1}^{N_{q'}} \phi_i(x_{q'}) n_1 w_{q'} - \sum_{q=1}^{N_q} \phi_{i,x}(x_q) w_q, \tag{4.20a}$$

$$\mathbf{r}_{i+N_n} = \sum_{q'=1}^{N_{q'}} \phi_i(x_{q'}) n_2 w_{q'} - \sum_{q=1}^{N_q} \phi_{i,y}(x_q) w_q. \tag{4.20b}$$

There are generally significantly more quadrature points than basis functions, and it is obviously essential that there are at least as many quadrature points as integration constraints if true machine precision must be achieved. Thus to meet the linear constraints one requires that $N_q \geqslant N_c$ while in practice it would be unlikely to be right at this threshold, and so one will have $N_q > N_c$ which results in an under-determined matrix equation to be solved. In our testing we found that solving for the solution with the minimum Euclidean norm (i.e. minimising the magnitude of the correction) gave the best results, whereas in our testing a solution found by computing only the first N_c corrections and leaving the others as zero—which still met all integration constraints to machine precision—was found to be less effective at recovering the optimal convergence order.

If higher-order bases are used then the number of constraints per node will also be higher (e.g. five per basis function for the 2D quadratic constraints), and so more quadrature points will be needed to provide enough DoFs to meet the constraints; however, as higher-order quadrature would be necessary for higher-order accuracy even in a standard FEM scheme, this is unlikely to be a limiting consideration.

Additionally, there are several possibilities for reducing the size of the linear system to be solved for larger problems. For any meshfree scheme, the under-determined nature of the problem means blocking together quadrature points into groups with only one correction calculated per group is possible. In FCIFEM specifically, since the FCI planes provide well-defined slices of the domain, unlike for entirely meshfree schemes, corrections can be computed to enforce the VCI constraints on each slice

of the domain independently, producing a set of multiple linear systems to be solved but with the scaling of the individual linear system sizes reduced by one dimension.

While the linear system solve(s) necessary for the proposed conservative VCI correction can be costly, it is only incurred once when the spatial discretisation is first computed. Therefore, it is hoped that for real problems of interest in MCF the overall decrease in the number of DoFs required for an accurate solution would sufficiently decrease the computational effort and memory requirements for time integration of the unsteady system to make the overall method cost-effective. For stationary problems such a trade off may or may not be worthwhile, but properly comparing such timings would require implementation into an at least somewhat optimised framework, which falls beyond the scope of this present work.

4.4 Error bounds

The error bounds under which a standard FEM converges with respect to the element size h (defined usually in terms of the diameter of the bounding or circumscribing circle of an element for an unstructured mesh) have been well established for several decades [150–153]. Looking particularly at Strang's seminal work [152] the convergence rate is given by

$$\|u - u^h\| = \mathcal{O}\left(h^{k+1} + h^{2(k+1)-m}\right), \tag{4.21}$$

where k is the polynomial order of completeness of the chosen FEM trial space, m is the order of the strong form of the PDE to be solved, and the norm is that induced by the L^2 inner product from equation (3.4).

For most simulations it is the first term that will dominate, and one will see an order of convergence with respect to the mesh size that is one order higher than the polynomial order of the chosen basis. Only in the event where a relatively low-order basis is chosen to discretise a relatively high-order PDE (e.g. a basis complete only to quadratic order used for a fourth-order PDE) will the second term generally become the limiting term.

For MLS-based meshfree schemes similar convergence rates have been elucidated [154–160] and can be written in an analogously simple form as

$$\|u - u^d\| = \mathcal{O}\left(d^{k+1}\right), \tag{4.22}$$

where d is the dilation parameter used in the scaling of the weight functions to determine the support size (it is often denoted ρ, but here d is used to avoid confusion with the notation used for the ramp function in the FCIFEM scheme), and k is again the order of polynomial basis used in the construction of the MLS shape functions. This use of the dilation parameter is important because there is obviously no direct element size in a meshfree method analogous to h in the FEM scheme. However, generally the dilation parameter is scaled proportionally to the average local node spacing in order to maintain a similar number of nodes covering all areas of the

domain, so convergence with respect to $\mathfrak{d}$ is generally directly comparable to $\mathfrak{h}$-refinement for a FEM, and the order of expected convergence is indeed the same in equations (4.21) and (4.22), being one order higher than the polynomial order of the basis.

Since our FCIFEM scheme hews as closely as possible to a linear FEM scheme, and given the similar spatial convergence results for meshfree schemes, it is therefore reasonable to expect second-order accuracy for our method, as will be demonstrated experimentally in the following chapter.

CHAPTER 5

FCIFEM NUMERICAL RESULTS

ALL OF THE simulations in this chapter were carried out using a 2D unit square domain $\Omega : (0,1)\times(0,1)$, where x and y are respectively associated with the ζ- and $x_\perp$-coordinates of the general FCIFEM scheme outlined in chapter 4 and may be used interchangeably. The prototype code that was used to carry out the tests is implemented in Python for its ease in rapidly iterating new developments and wide availability of external libraries.

It is perhaps worth noting here that the primary purpose of the code in its present state is only for testing the validity of the methodology and so little attention has been paid to optimisation at this point, and commensurately little attention will be given here to simulation timings. It is understood that this is not entirely optimal given the ultimate aim of reducing computational cost, but the intent for this thesis is to validate the avoidance of expensive field-aligned meshing procedures by using meshfree ideas. By achieving good error convergence using these meshfree ideas while still demonstrating that a substantial reduction in the number of DoFs is possible for spatial discretisations via field alignment, this should then inherently lead to a reduced cost of time integration for unsteady problems.

5.1 Representation accuracy

We first test the representation power of the proposed basis by computing the L^2 projection of a smooth function into our FCIFEM basis function space, with figure 5.1 showing the results of two convergence studies. This simply means we are solving the equation

$$u(x,y) = f(x,y), \tag{5.1}$$

using the FCIFEM scheme with a standard Bubnov-Galerkin approach.

In the left column we are approximating

$$f(x,y) = \sin(2\pi x)\sin(2\pi y), \tag{5.2}$$

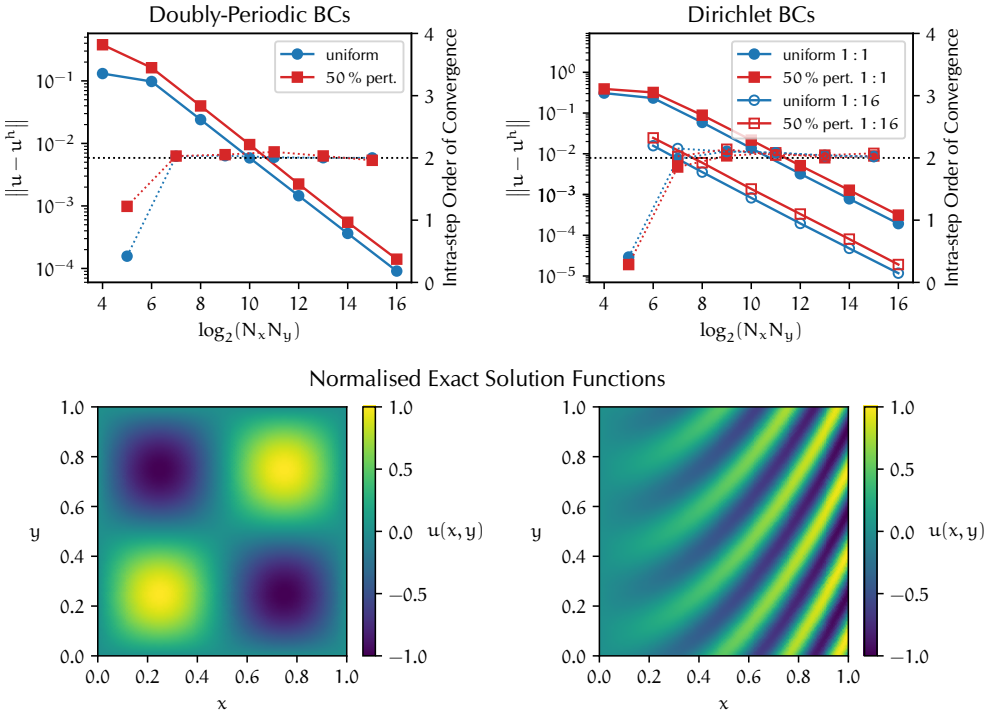


FIGURE 5.1: Convergence study of L^2 projection for FCIFEM, where left column uses periodic BCs and sinusoidal mapping while right column uses Dirichlet BCs with quadratic mapping. *Top row* shows L^2 norms of error (solid lines, left y-axes) and convergence orders (dotted lines, right y-axes) for L^2 projections of smooth functions into the FCIFEM function space. Data for uniform node spacing is shown in blue with circle markers while data for node locations randomly perturbed by up to 50 % of the uniform spacing is shown in red with square markers. Filled markers indicate a 1 : 1 ratio of $N_x : N_y$ grid divisions while unfilled markers indicate a 1 : 16 ratio. *Bottom row* shows the normalised analytic functions being approximated.

as shown in the bottom left plot of figure 5.1, with doubly periodic BCs. A 1 : 1 ratio of the number of grid divisions $N_x : N_y$ is used along with a sinusoidal mapping function given by

$$\begin{gathered} Q(\mathbf{x}, \zeta) = \Big[\zeta,\ y + A\left[\sin\left(2\pi[\zeta + \alpha]\right) - \sin\left(2\pi[x + \alpha]\right)\right]\Big]^T, \\ A = 0.2, \quad \alpha = 0.25. \end{gathered} \tag{5.3}$$

The desired second-order accuracy with respect to the average node spacing, analogous to that for the linear FEM functions used in the basis construction, is achieved for both uniformly spaced and randomly perturbed nodal locations, which demonstrates the robustness of the representation against both curvature in the mapping and arbitrary node placement. For the perturbed simulations, random displacements were added first to the x-coordinates of each FCI plane and then to the y-coordinates of every node. The magnitude of the displacements was up to 50 % of the uniform spacing in the respective dimension, which is the maximum possible without potentially changing the ordering of the FCI planes or nodes.

In the right column we are approximating

$$\begin{gathered} f(x,y) = x\sin\left(2\pi n[y - ax^2 - bx]\right), \\ n = 3, \quad a = 0.95, \quad b = 0.05, \end{gathered} \tag{5.4}$$

as shown in the bottom right plot of figure 5.1, with fully Dirichlet BCs. Since the y-coordinate is transformed along a quadratic, we use a mapping aligned to this transformation, namely

$$Q(x,\zeta) = \left[\zeta,\ y + a(\zeta^2 - x^2) + b(\zeta - x)\right]^T, \tag{5.5}$$

and the values of a and b were chosen such that $Q\left(\left[0,\ 0\right]^T, 1\right) = \left[1,\ 1\right]^T$.

Again we confirm excellent convergence for both uniformly spaced and randomly perturbed nodes, but since this solution actually has a significant anisotropy, unlike the doubly periodic case, we also test using $16X$ fewer grid divisions in the x-direction, which can be identified as the ζ parameter of the FCIFEM scheme. We can see that almost identical accuracy is reached even with this sixteenfold reduction in the number of solution nodes, clearly demonstrating that a $1:1$ ratio uses far more DoFs than necessary to represent the slow variation in the mapping direction. Alternatively, for the same number of DoFs the FCIFEM scheme allows more than a full order of magnitude reduction in the error.

5.2 Irregular grid and improved integration

The ability to discretise an elliptic operator is tested by simulating the Poisson problem

$$\nabla^2 u = \sin(2\pi x)\sin(2\pi y), \tag{5.6}$$

with figure 5.2 showing the convergence results (the forcing function on the RHS is the same as that visualised in the bottom left plot of figure 5.1 in the previous section). Doubly periodic BCs are still employed, but the values of nodes with $x = 0$ or $y = 0$ coordinates are strongly constrained to zero when solving the resulting linear system in order to ensure the correct unique solution is selected. All simulations use the same sinusoidal mapping given in equation (5.3).

While we still have excellent results when the nodes are uniformly spaced, the order of convergence now deteriorates when the node positions are randomly perturbed, even by just 10 % of the uniform spacing. The order of accuracy can be largely recovered by improving the quadrature, as we demonstrate by increasing the order of the Gauss quadrature used from 3 to 10 in the right plot of figure 5.2, indicating that this problem likely arises due to the non-polynomial basis and arbitrarily overlapping domains of support making it impossible to compute the integrals exactly in the Galerkin formulation. Unfortunately, it can be seen that even this 10th-order quadrature does not fully recover the convergence, and so it is computationally

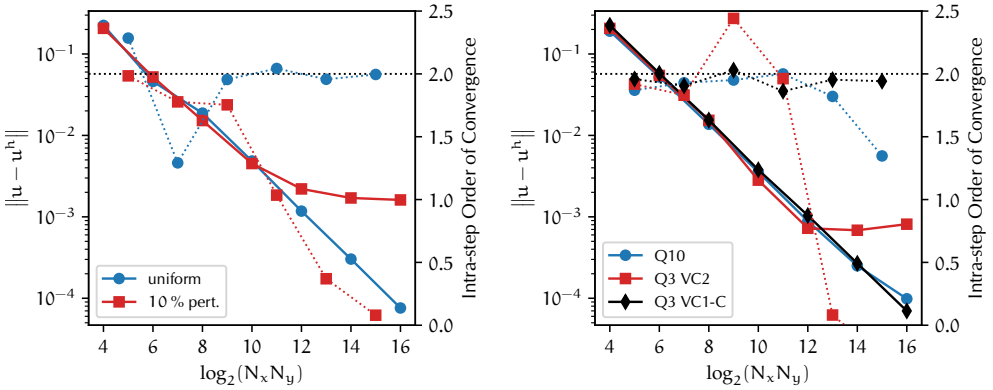


FIGURE 5.2: L^2 norms of error (solid lines, left y-axes) and convergence orders (dotted lines, right y-axes) for Poisson simulation with periodic BCs. Left plot shows loss in accuracy when node locations are randomly perturbed up to 10 % of uniform spacing, using 3rd-order Gauss quadrature. In the right plot, all simulations are 10 % perturbed, Q number is the order of underlying Gauss quadrature, VC means the scheme is variationally consistent [149] with the number indicating whether linear (1) or quadratic (2) constraints are met, and final C denotes our conservation-preserving VCI scheme.

very expensive to simply increase the quadrature order high enough to achieve fully satisfactory results.

As discussed in section 4.3, VCI was investigated according to the work of Chen et al. [149] and found to be an effective tool for improving the accuracy of the method back towards the optimal rate of convergence with less computational effort. Our results in the right plot of figure 5.2 show that Chen's assumed strain method (VC2 in the figure), even when enforcing the quadratic constraints, is less effective at improving the accuracy compared with our conservative reweighting method (VC1-C in the figure) when enforcing only the linear constraints. The linear assumed strain formulation was also tested, but produced little or no improvement for this problem and so is not shown in the figure. The computational time to solve the linear system for the VC1-C was greater than the linear and less than the quadratic assumed strain method, while it was by far the most effective at restoring the convergence order.

5.2.1 PATCH TEST

A useful test for easily determining if a given numerical scheme is providing the desired order of consistency is to use a very simple model problem known as a *patch test*. Essentially one wants to solve a problem where the known solution is a non-trivial polynomial of the desired consistency order. Since the FCIFEM scheme is linear, we will use a linear patch test where the solution is given by the plane

$$u(x, y) = g(x, y) = x + 2y, \tag{5.7}$$

and we note that since this is not a periodic function, we must also use Dirichlet BCs to enforce the function g on the boundary $\Gamma = \delta\Omega$ (a more complete test of essential

		Linear VCI method		
Mapping	Nodes	None	AS	CQR
Straight	uniform	4.49×10^{-12}	4.49×10^{-12}	4.49×10^{-12}
	10 % pert.	3.65×10^{-3}	7.70×10^{-12}	1.14×10^{-11}
Linear	uniform	2.87×10^{-3}	6.81×10^{-12}	6.63×10^{-12}
	10 % pert.	2.05×10^{-3}	2.04×10^{-11}	3.42×10^{-9}
Quadratic	uniform	6.24×10^{-4}	5.39×10^{-4}	8.09×10^{-3}
	10 % pert.	1.33×10^{-3}	5.13×10^{-4}	3.92×10^{-3}

TABLE 5.1: L^2 norms of error for linear patch tests with the FCIFEM scheme showing the effects of mapping, node placement, and linear VCI correction, where AS indicates assumed strain method and CQR indicates our conservative quadrature reweighting scheme. All tests were carried out with $N_x = N_y = 8$ grid divisions in each direction with either uniform or 10 % perturbed node distributions, using either a straight (i.e. coordinate-aligned) mapping, a linear mapping aligned at 45°, or the quadratic mapping from equation (5.11).

boundaries is presented shortly in section 5.4).

Also, since the second-order derivatives of such a linear function are obviously zero, this essentially means we are solving the homogeneous Laplace equation

$$\nabla^2 u = 0, \tag{5.8}$$

where it is then only the BCs which actually enforce the specific planar solution which has been selected.

Since we are largely interested only in whether the numerical scheme is able to exactly represent the solution, we will only present results for a single node distribution to see in what cases we achieve exact (i.e. only floating point error) agreement. The results of these simulations are given in table 5.1 and we can see a few interesting results.

Firstly, we can see that when no special mapping function is in effect (the straight, coordinate-aligned case) and with a uniform node distribution, then the patch test is satisfied exactly even with no VCI correction, which bears similarity to the results in figure 5.2 where a uniform node distribution achieved the expected convergence order without needing correction. However, for the other mappings this does not hold, which indicates that while the patch test can tell us whether a method is truly able to represent polynomials of a given order, passing the patch test exactly is not a truly necessary condition for achieving optimal convergence, as the results in figure 5.2 were obtained with a sinusoidal mapping but still reached optimal convergence for the uniform case without correction. It is perhaps worth noting that this is likely a consequence of the essential BCs, which break the regularity of the uniform node distribution that would otherwise still exist for non-coordinate-aligned mappings

if only periodic BCs were employed, but still highlights a limitation of what such a simple test can tell us.

Secondly, we can see that both the assumed strain method and our conservative quadrature reweighting scheme are able to achieve exact solutions for the coordinate-aligned and angled linear mappings for both uniform and perturbed node distributions, showing their equivalence in meeting the integration constraints.

Lastly, we can see that even with VCI correction none of the simulations using the quadratic mapping achieved exact solutions. This is because the curvature of the mapping means that the basis is actually no longer able to exactly represent linear polynomials in our FCIFEM scheme. Similarly as in the first point though, this indicates that this is also not a truly necessary condition for optimal convergence, as once again optimal convergence orders in figure 5.2 were achieved with a curved sinusoidal mapping (and also later in figure 5.4 with the same quadratic mapping and Dirichlet BCs as used here), indicating that the ability to converge towards a smooth function as in section 5.1 would seem to be sufficient.

5.3 Periodic anisotropic Poisson problem

Following the lead of [117] we will demonstrate the benefit of our basis alignment by simulating a 2D test problem that has a relatively high degree of non-grid-aligned anisotropy. Sticking with a Poisson equation and doubly periodic BCs with constrained nodes as in the previous test problem, we will now use the forcing term

$$\nabla^2 u = \frac{1}{2}\sin\left(2\pi n[y - x]\right)\left[1 + \sin\left(2\pi y\right)\right], \tag{5.9}$$

with $n = 8$ which will result in a solution possessing structures that are largely aligned in the diagonal direction $\begin{bmatrix}1, & 1\end{bmatrix}^T$ and gradients which are significantly smaller parallel to this direction as compared to perpendicular, or when compared to either of the coordinate directions. The normalised analytic solution to this problem is shown in the bottom plot of figure 5.3, clearly showing the strong non-coordinate aligned anisotropy.

Results for these simulations are shown in figure 5.3 where the error for all configurations is seen to converge with second-order accuracy in the asymptotic region. For both uniform and perturbed node placements, one starts with a baseline simulation using a standard grid-aligned basis but which is *unaligned* with respect to the anisotropy, with an equal $1:1$ number of divisions $N_x : N_y$. This is compared to four other simulations which all use a mapping *aligned* to the anisotropy, with ratios of $1:1$, $1:4$, $1:8$, and $1:16$ grid divisions between the x- and y-coordinate directions, as the x-direction corresponds to the ζ-coordinate of the FCIFEM scheme. This means that the larger number of small divisions in the y-direction are capturing the small-scale, large-gradient variations perpendicular to the anisotropy, while large divisions in the x-direction are sufficient to capture the large-scale, small-gradient

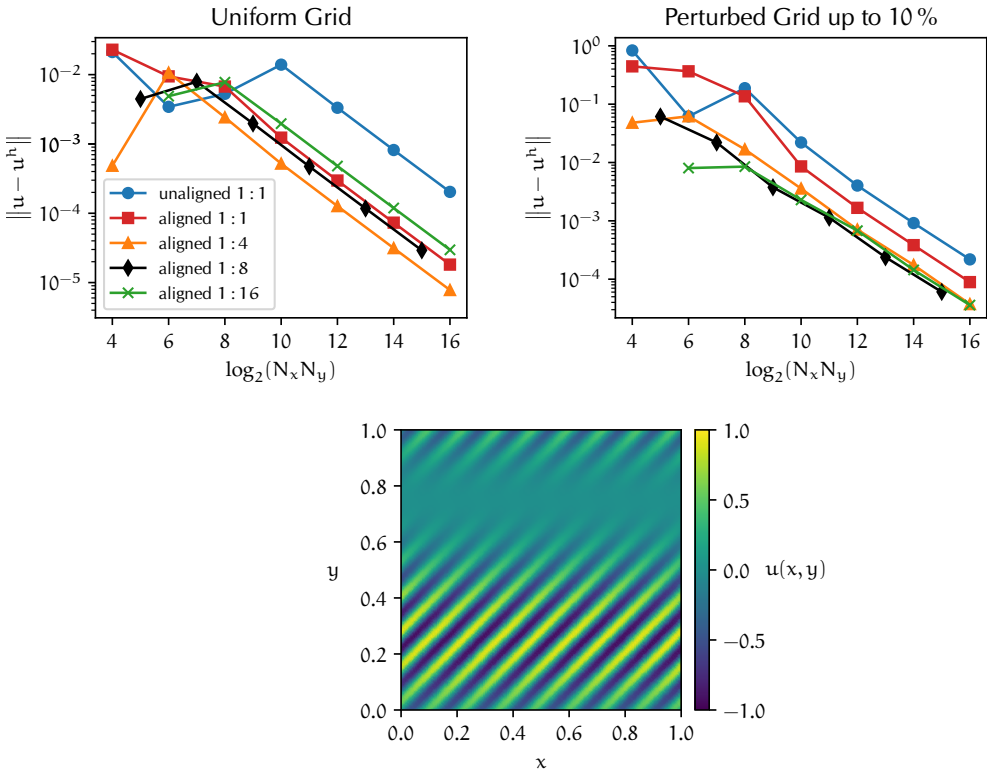


Figure 5.3: Convergence study of Poisson simulations with strong non-coordinate aligned anisotropy, comparing effects of aligning basis to anisotropy and changing $N_x : N_y$ ratio of grid spacings in coordinate directions. *Top row* shows L^2 norms of error for simulations using (top left) a uniform grid distribution of nodes and (top right) the same distribution with nodes perturbed up to 10 % of the uniform spacing. (bottom) Normalised exact solution function.

variation along the anisotropy. We can see that in all cases the alignment of the basis functions to the anisotropy improves the achieved accuracy for a given number of nodes in the simulation.

5.4 Dirichlet boundaries

For the next test case we will prescribe an analytic solution to the Poisson problem, only now using essential BCs which will enforce a given function g (which is equal to the prescribed solution) on the boundary $\Gamma = \delta\Omega$. Our prescribed solution is given by

$$\begin{aligned} u(x, y) = g(x, y) = x \sin\left(2\pi n[y - ax^2 - bx]\right), \\ n = 3, \quad a = 0.95, \quad b = 0.05. \end{aligned} \tag{5.10}$$

If we let x be the mapping coordinate ζ as described in our FCIFEM scheme, then this problem solution has anisotropic structures aligned along the quadratic mapping given by

$$Q(\mathbf{x}, \zeta) = \left[\zeta,\ y + a(\zeta^2 - x^2) + b(\zeta - x)\right]^T, \tag{5.11}$$

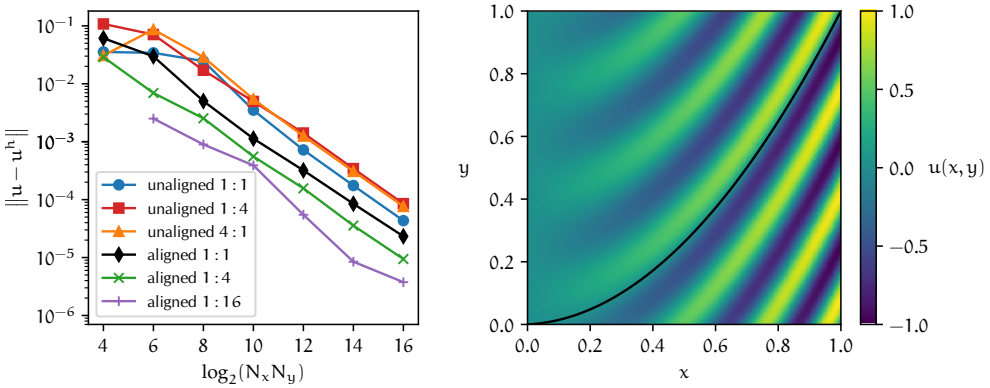


FIGURE 5.4: (left) L^2 norms of error for a Poisson simulation with quadratic anisotropy and Dirichlet BCs, comparing effects of aligning basis to anisotropy and changing $N_x : N_y$ ratio of grid spacings in coordinate directions. (right) Example of well-converged final solution, with the quadratic anisotropy illustrated by the solid line comprising the locus of points which map to the origin.

where $\mathbf{x} = \left[x,\ y\right]^T$. The values of a and b have been selected such that

$$Q\left(\left[0,\ 0\right]^T, 1\right) = \left[1,\ 1\right]^T, \tag{5.12}$$

i.e. the origin point at the bottom-left corner of the domain maps to the top-right corner of the domain. Using an integer value of n makes the solution periodic in the y-direction, such that intermediate testing could be performed with mixed BCs which were Dirichlet in the x-direction only, but still periodic in the y-direction; however, only results for fully Dirichlet BCs are reported here, as no particular differences were observed in the mixed case as compared to earlier doubly periodic simulations. We directly computed the Laplacian of the above prescribed solution in order to provide the associated forcing function in the Poisson equation.

Results for a convergence study are shown in figure 5.4 with essentially second-order accuracy observed for all data series. Simulations were again run in two configurations, either with or without aligning the basis to the anisotropy. For each case, multiple convergence tests were run using different ratios $N_x : N_y$ of the number of grid divisions in coordinate directions, with x identified with ζ as before. All simulations have node spacings perturbed up to 10 % and use the conservative VCI quadrature scheme discussed in section 5.2.

For the unaligned cases, not using a 1 : 1 ratio reduces the solution accuracy because we can clearly see in the right of figure 5.4 that there are large gradients in both the x- and y-coordinate directions, so the grid must be fine enough to sufficiently resolve both directions properly when the basis is grid-aligned. Aligning the basis instead to the quadratic anisotropy decreases the error somewhat on its own, but more importantly it allows for fewer grid divisions to be then used in the x-direction, since this coordinate is now only representing the slow variations along the quadratic mapping. The simulations which use a 1 : 16 ratio of $N_x : N_y$ grid

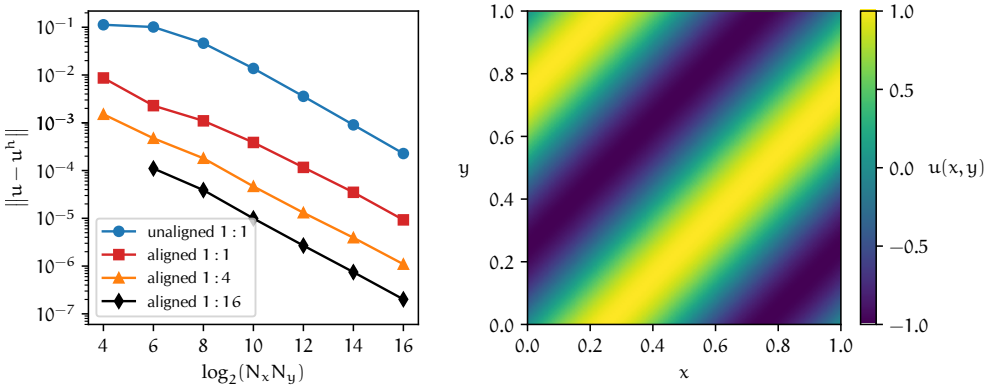


FIGURE 5.5: (left) L^2 norms of error for simulations of an unsteady anisotropic diffusion equation with periodic BCs on a randomly perturbed grid, comparing effects of aligning basis to anisotropy and changing $N_x : N_y$ ratio of grid spacings in coordinate directions. (right) Plot of the sinusoidal IC having zero gradient in the direction along $\left[\cos\left(\frac{\pi}{4}\right), \sin\left(\frac{\pi}{4}\right)\right]^T$, which is aligned with the anisotropic diffusion such that this is also the analytic solution at any time.

divisions allow the same or better level of accuracy to be reached with ~ 4X fewer nodes compared to the base case.

5.5 ANISOTROPIC DIFFUSION

Our final test case solves the unsteady anisotropic diffusion equation given by the initial value problem (IVP)

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial v^2}, \tag{5.13}$$

where

$$v = x\cos(\theta) + y\sin(\theta), \tag{5.14}$$

and we set $\theta = \frac{\pi}{4}$. For our simulations we set the diffusion coefficient D to unity and advance the simulation for 100 time steps with $\Delta t = 0.01$ to a final time of $t_{\text{final}} = 1$ using a Crank-Nicolson time integration scheme and doubly periodic BCs.

By starting the simulation with the initial condition (IC)

$$u_0(x, y) = \sin(2\pi[x - y]), \tag{5.15}$$

as shown in the right plot of figure 5.5, the gradients of u are everywhere exactly zero in the direction of $\hat{v}$ and so the IC is also the analytic solution at any point in time; therefore, any change in u will be due to non-physical numerical diffusion of the method used.

For all of the simulation results, which are shown in figure 5.5, we can see clean second-order convergence with random perturbations of the node locations up to 10 % of the uniform spacing used for all simulations in the figure. The same set of runs was also carried out with a uniform grid but is not shown here, because the error

in the unaligned case was almost identical to the perturbed case, while the errors in the aligned cases were exactly zero. This occurred because the alignment meant that nodes in each plane then mapped perfectly onto the nodes of the neighbouring planes and so there is no spurious numerical diffusion, only floating-point error. By comparison, in the perturbed cases (i.e. those shown in the plot) the alignment greatly reduces the errors by making the diffusion from one plane to another as direct as possible, but since the nodes no longer map directly onto other nodes there is still a spreading effect, as any given node must now interact with multiple nodes on each neighbouring plane.

This spreading effect is also present in all standard FEM (as well as FDM and FVM) schemes, however, and so is not a new problem introduced by our current method. In fact, as discussed above, the ability to align the basis with the diffusion direction greatly reduces the spurious error, and the ability to then use fewer planes in the anisotropic direction as a result of the alignment also decreases the number of node interactions where this error can occur, further reducing its effect. This is clearly demonstrated in the $1:4$ and $1:16$ cases for the $N_x:N_y$ ratio shown in figure 5.5. Thus we would in fact expect our method to generally perform better than standard FEM schemes in terms of spurious numerical diffusion, which is important for accurate simulations in many MCF problems.

It is also worth noting that the results in figure 5.5 were taken without the use of any VCI correction, which was found in other tests (not shown in the figure, but discussed in the following paragraph) to have almost no effect on the diffusion errors. This is likely because this spurious diffusion error is due to inherent spreading of information between imperfectly aligned nodes, and therefore not related to the quadrature error, so the extra effort of correcting the quadrature does not have any benefit for this type of problem.

Finally, this non-stationary IVP case provides us with an opportunity to verify the conservation properties which were elucidated back in section 4.2, and for this error the use and type of VCI correction is important. For all of the simulations which were run either without using any VCI correction or using the conservative quadrature reweighting scheme the weighted change in u was found to be zero to machine precision (order of magnitude 10^{-12} or smaller in all cases). Conversely, when running the same set of simulations but instead using Chen's [149] assumed strain formulation the resulting weighted change in u was found to be between 10^{-3} and 10^{-6}. As these simulations were only run for 100 time steps, this conservation error could easily become quite significant for longer simulations, being already somewhat comparable to the size of the spurious diffusion errors shown in figure 5.5.

CHAPTER 6

A FULLY MESHFREE SCHEME

WHILE EXCELLENT convergence results have already been obtained with the FCIFEM method, the construction proposed in this thesis is fundamentally based on linear FEM schemes. This particular construction of the scheme was proposed to recover the Kronecker delta property and avoid many of the difficulties of fully meshfree schemes in terms of choosing support domain sizes and shapes for the basis functions, selecting weight functions, and ensuring sufficient cover for all parts of the domain, but it is not entirely clear how one might directly extend this construction to higher-order accuracy. Also, as mentioned in section 5.2.1, when there is curvature in the mapping this FCIFEM construction is no longer able to exactly represent linear polynomials, and although the desired second-order convergence was still achieved in all tests it would be preferable to have a method that was truly consistent up to a given polynomial order. (We note that McMillan [117] used a slightly different construction and tested quadratic splines, but did not recover the Kronecker delta property and still left open questions about polynomial consistency with arbitrary mappings.)

Having a high-order method would also be desirable for increasing the computational intensity of algorithms, i.e. increasing the ratio of computations done per unit of memory transferred, which is beneficial in more fully utilising modern hardware which is generally limited more strongly by memory bandwidth and communication while having plenty of FLOPS (floating-point operations per second) to spare. In general for smooth solutions, high-order methods are more efficient than low-order methods for stringent enough accuracy conditions.

Additionally, having a higher-order scheme is necessary for the discretisation of higher-order differential operators, as linear FEM-type schemes can only handle maximally second-order operators by using integration by parts to reduce the derivatives to first order, with all higher-order derivatives of the linear basis functions obviously being zero.

For these reasons we have also carried out very preliminary investigations of a fully meshfree MLS-type scheme for higher-order simulations, which we will refer to analogously as the FCIMLS (flux-coordinate independent moving least squares) method.

6.1 FCIMLS SCHEME

As mentioned in section 3.4, selecting a reasonable size and shape of the compact support for the shape functions plays a large role in determining the success of the simulations, and it can be tricky to define an automatic procedure for ensuring a workable selection in all cases. The most critical condition that must be satisfied, as noted in section 3.3.3, is that every point in the domain must be covered by at least as many non-zero shape functions as there are polynomials in the MLS basis to ensure that the moment matrix is invertible everywhere. However, this must also be balanced against the need to keep the support as compact as possible, which is crucial for computational tractability by keeping the resulting operator matrices sparse and banded. It can also be important for stability and accuracy considerations, as unnecessarily large support sizes are found in our testing to lead to more ill-conditioned operator matrices and increased error as shown later in section 7.3.

We will therefore use essentially the same arrangement of nodes as for our FCIFEM scheme, where all of the interior nodes are placed on FCI planes of constant parallel coordinate. Even though we no longer require any standard FEM mesh on these planes or the domain boundaries this arrangement can still be helpful both for ensuring sufficient domain coverage and reducing the search space for nearby nodes during shape function evaluations. It also reduces the memory required for storing node positions, as only a single value is needed rather than a vector for the parallel coordinate of all nodes on a given FCI plane.

It aids in achieving full domain cover, because it helps to separate the problem between the long parallel and short perpendicular length scales. Much as in the FCIFEM scheme where we had one dimension lower standard FEM meshes on the FCI planes, we now essentially have to solve a standard meshfree coverage problem on each FCI plane, but it is one dimension lower and fully isotropic. The choice of a suitable support size in the anisotropic direction can then be determined much more easily.

For node searching, it should be fairly evident that this set up allows for the search to immediately be narrowed down to those nodes located on FCI planes within the parallel support distance, with only a single comparison needed per plane, rather than one for each node on the plane. If the nodes were simply (quasi)randomly distributed throughout the space many more comparisons would need to be made, whereas we have again reduced the most intensive part of the search to a lower dimensional problem on each FCI plane.

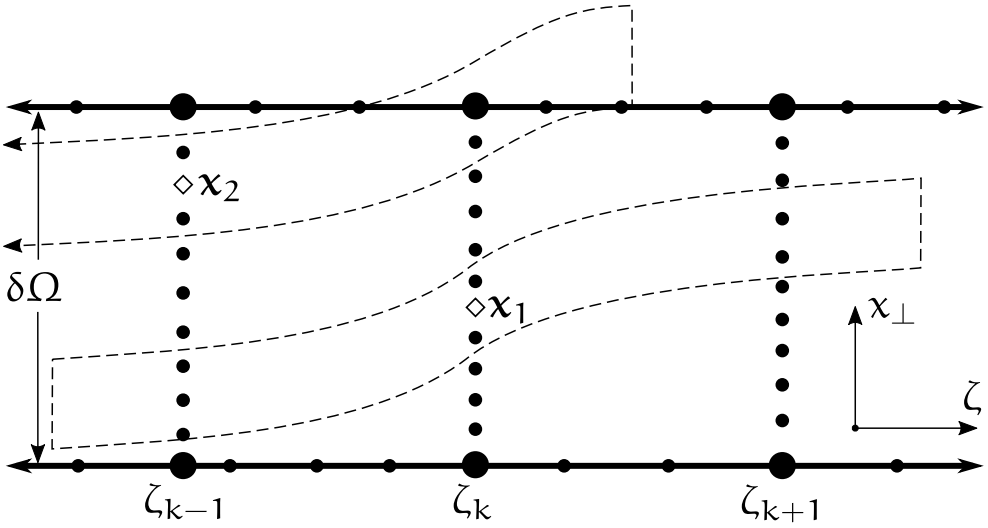


FIGURE 6.1: Illustration of the FCIMLS scheme discretising a 2D domain Ω. Nodes are arranged in FCI planes but without any connectivity necessary between them. Shape functions have box-shaped domains of support, illustrated for two arbitrary nodes labelled x_1 and x_2, elongated in the parallel coordinate direction and sheared by a mapping function $Q : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ in the perpendicular coordinate direction and which can freely extend outside the domain boundary. This example would be for a linear MLS basis, as the support size extends ~1.5X the uniform spacing in each coordinate direction.

6.2 SHAPE FUNCTION CONSTRUCTION

For the preliminary investigation carried in this thesis, we restrict ourselves to box-shaped supports as given in equation (3.26), where the distance in each coordinate direction is considered independently of the others in determining if a given point falls within the support, and which can use a different threshold in each coordinate direction if desired. The difference in our case is that when considering the perpendicular coordinate direction we will first apply the mapping function to map the evaluation point to the same parallel coordinate as the potential node before computing the perpendicular distances, essentially shearing the box support along the mapping direction as illustrated in figure 6.1.

The support sizes must be chosen to ensure invertibility of the moment matrix, so on each FCI plane the support sizes must be made large enough to ensure cover by a minimum number of nearby nodes. For a linear polynomial basis this is similar to the FCIFEM case where having just the nearest neighbours is sufficient, although now one generally wants the support to extend noticeably past the nearest neighbours, in practice about an extra ~0.5X the uniform spacing, also illustrated in figure 6.1. This is because good weight functions will have decayed quite close to zero for some distance near the edges of the support, as shown in figure 3.2, so nodes that are too near the edge of the support may not contribute significantly enough to ensure invertibility.

Then, the support in the parallel coordinate direction is made large enough to extend well past at least one FCI plane in each direction (unless at a domain boundary, although this would not be an issue in a periodic MCF device such as a tokamak or stellarator). It should also be clear from figure 6.1 that the support domains of the shape functions are free to extend past the domain boundaries, with the MLS formalism maintaining the partition of unity of the basis up to the boundary, and the procedures outlined in section 6.3 needed to enforce Dirichlet BCs because of the lack of the Kronecker delta property.

6.2.1 Centred and scaled evaluation

Once an appropriate support domain has been established and the nodes found whose support covers the evaluation point, then the least squares error minimisation procedure outlined in section 3.3.3 determines how to evaluate the shape functions. However, in order to implement this as efficiently as possible into code, there are a couple of particulars to consider. Firstly, an LU factorisation of the moment matrix can be used to avoid direct computation of its inverse and, even more importantly, can be reused in the computations of the shape function derivatives. An excellent overview of this procedure is given in [128] which we will outline and follow below. Secondly, it is preferable that the evaluation of the polynomial basis $\mathbf{p}$ is transformed by first *centring and scaling* its argument [129, 155, 161, 162].

This gives the modified procedure from [128] as follows, starting by writing the shape functions in the form

$$\Phi_I(\mathbf{x}) = \mathbf{c}^\intercal(\mathbf{x}) w_I(\mathbf{x}) \mathbf{p}\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right), \tag{6.1}$$

where the consistency conditions $\mathbf{c}(\mathbf{x})$ can be determined from

$$\mathbf{A}(\mathbf{x})\mathbf{c}(\mathbf{x}) = \mathbf{p}(\mathbf{0}), \tag{6.2}$$

and the moment matrix $\mathbf{A}(\mathbf{x})$ is computed as

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{n} w_I(\mathbf{x}) \mathbf{p}\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right) \mathbf{p}^\intercal\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right). \tag{6.3}$$

It is worth noting that the weight function $w_I(\mathbf{x})$ is also evaluated in a centred and scaled manner, but I have notated it simply as a function of $(\mathbf{x})$ because there is a subtle yet very important difference in the centring compared to that for the polynomial basis. Namely, the weight function is centred in the mapped coordinate system, using the mapping function $Q(\mathbf{x}, \zeta)$ to determine the perpendicular distances as outlined in above with regards to the chosen support, as it is the weight function which actually determines this support in the computation. Conversely, the centring for the polynomial basis evaluation is carried out in the regular coordinate system, as this is required to achieve the exact representation of polynomials in this regular coordinate system. The scaling factor d is the dilation parameter and is generally

the same for both w_I and $\mathbf{p}$, and is simply given by the support size for the given coordinate direction. However, in the case where a different support size d_I is allowed for each node then for the scaling of the polynomial basis the average value of $d = \sum_I d_I / N_n$ should be used such that it is a constant, which is important for consistency [129].

To then efficiently compute the coefficient vector $\mathbf{c}(\mathbf{x})$ we determine the LU factorisation of $\mathbf{A}(\mathbf{x})$ and perform backward substitution as

$$\mathbf{L}\mathbf{U}\mathbf{c}(\mathbf{x}) = \mathbf{p}(\mathbf{0}) \quad \Longrightarrow \quad \mathbf{c}(\mathbf{x}) = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{p}(\mathbf{0}), \tag{6.4}$$

with the shape function then easily following from equation (6.1), and we note that the centring has made this computation significantly cheaper by only requiring evaluation of the basis at the origin of the centred coordinate system, which does not depend on the evaluation point or node, and thus can be computed once and cached.

To perform computation of the first derivatives, we will use a subscript k to indicate partial differentiation with respect to one of the coordinate directions, i.e. $(\cdot)_k = \frac{\partial(\cdot)}{\partial k}$, to give

$$\begin{aligned}\Phi_{I,k}(\mathbf{x}) = \mathbf{c}_k^T(\mathbf{x})\mathbf{p}\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right) w_I(\mathbf{x}) + \\ \mathbf{c}^T(\mathbf{x})\left[\mathbf{p}_k\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right) w_I(\mathbf{x}) + \mathbf{p}\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right) w_{I,k}(\mathbf{x})\right],\end{aligned} \tag{6.5}$$

with

$$\begin{aligned}\mathbf{A}_k(\mathbf{x})\mathbf{c}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{c}_k(\mathbf{x}) &= \mathbf{0}, \\ \mathbf{A}(\mathbf{x})\mathbf{c}_k(\mathbf{x}) &= -\mathbf{A}_k(\mathbf{x})\mathbf{c}(\mathbf{x}), \\ \mathbf{c}_k(\mathbf{x}) &= \mathbf{A}^{-1}(\mathbf{x})[-\mathbf{A}_k(\mathbf{x})\mathbf{c}(\mathbf{x})],\end{aligned} \tag{6.6}$$

where one can reuse the $\mathbf{L}\mathbf{U}$ factorisation of $\mathbf{A}(\mathbf{x})$ which has already been computed, and the derivatives of the moment matrix are found as

$$\begin{aligned}\mathbf{A}_k(\mathbf{x}) = \sum_{I=1}^{n} \Big\{ w_{I,k}(\mathbf{x})\mathbf{p}\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right)\mathbf{p}^T\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right) + \\ w_I(\mathbf{x})\left[\mathbf{p}_k\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right)\mathbf{p}^T\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right) + \mathbf{p}\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right)\mathbf{p}_k^T\left(\frac{\mathbf{x}-\mathbf{x}_I}{d}\right)\right]\Big\}.\end{aligned} \tag{6.7}$$

6.3 Dirichlet boundaries

As noted in section 3.4 the lack of the Kronecker delta property in meshfree Galerkin schemes complicates the enforcement of Dirichlet BCs because one can no longer simply set the nodal coefficients on the boundary to the desired values as in a standard FEM. Instead these boundary coefficients generally remain as DoFs that must be solved for and various methods have been investigated in the literature to impose the essential BCs on the resulting solution. A non-exhaustive and high-level summary of some of these methods is briefly given here.

Perhaps the most obvious way to enforce essential BCs is through *strong form collocation* at the boundary nodes, where the rows of the operator matrix corresponding to such boundary nodes are simply replaced with entries corresponding to the constraint that the sum of non-zero shape functions at the nodal point must equal the desired boundary function, the value of which is overwritten into the corresponding entry of the forcing vector. However, this does not place any constraints on how the solution function behaves between these collocation points, and is inconsistent with the derivation of the weak Galerkin form, so error convergence tends be suboptimal, although more complicated corrections have been developed to help restore convergence [163].

Coupling to a FEM has been exploited by several authors [131, 161, 164, 165] to take advantage of the fact that standard FEMs do possess the Kronecker delta property, by transitioning or blending the meshfree approximation in the interior of the domain with a standard FEM mesh and discretisation existing only on, and possibly very near to, the boundary. Some care must be taken to ensure consistency of the coupled discretisations, and the resulting shape functions tend to have a higher degree of complexity, necessitating more accurate quadrature in and near the transition regions, but it remains a viable option for many problems, and can be particularly attractive if a suitable mesh may already be available for describing the domain geometry in any case.

The *penalty method* [166, 167] is based on a modification of the weak form that is attractive primarily because of its simplicity and ease of implementation; however, it enforces the BCs only approximately. It involves a *penalty parameter* β which can be made larger to enforce the BCs more strongly but at the cost of increasing the condition number of the resulting operator matrix as β increases [129], making the choice of an optimal β sometimes tricky. An improvement on the penalty method that preserves the consistency of the weak form is known as *Nitsche's method* [168] and has been found to allow for optimal convergence of the boundary error with small enough β to keep the operator matrix well-conditioned. The main drawback of this method lies in actually deriving the modifying terms in the weak form, which is not as easily generalised for new problems as other methods.

6.3.1 Method of Lagrange multipliers

The method that we use for our simulations is that used in the original EFG paper [130], namely the method of *Lagrange multipliers*. Like the penalty method, the use of Lagrange multipliers is also based on a modification of the weak form of the problem, although in this case there is no β parameter that must be chosen and the BCs are always accurately enforced. The main drawback to this method is that the Lagrange multipliers must themselves be solved for, so the resulting linear system is larger and no longer positive definite even if the original operator matrix would have been, because zero entries are introduced on the diagonal. This makes the method more

computationally expensive, but it is simple to implement and generally accurate, so it is still useful for 2D simulations such as our test problems, where the size of the final problem matrices is relatively small and the increase due to the Lagrange multipliers is relatively trivial in terms of additional time to solution.

As an example we can consider the Poisson BVP given by

$$-\nabla^2 u(\boldsymbol{x}) = f(\boldsymbol{x}), \tag{6.8}$$

on the domain Ω subject to the Dirichlet BCs $u(\boldsymbol{x}) = g(\boldsymbol{x})$ on the boundaries $\Gamma = \delta\Omega$. Computing the Galerkin solution to this problem is then equivalent to minimising the following functional based on the variational principle [169]

$$\begin{aligned} \Pi(v) &= \langle -\nabla^2 u \mid u \rangle - \langle u \mid f \rangle - \langle f \mid u \rangle, \\ &= \int_\Omega \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} v \, d\Omega - 2 \int_\Omega v f \, d\Omega, \end{aligned} \tag{6.9}$$

where we have initially ignored the Dirichlet boundaries, and the final solution is found as

$$u = \underset{v}{\operatorname{argmin}} \, \Pi(v). \tag{6.10}$$

This minimisation requires finding the extremum of $\Pi(v)$ such that the variation is zero with respect to variations in the argument, i.e. we must solve for

$$\delta\Pi(u) = \frac{\delta\Pi}{\delta v}(u) = 0. \tag{6.11}$$

In order to impose the Dirichlet BCs on the system we add an additional constraint term to the functional as

$$\Pi(v) = \int_\Omega \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} v \, d\Omega - 2 \int_\Omega v f \, d\Omega + \int_\Gamma \lambda C(v) \, d\Gamma, \tag{6.12}$$

where lambda is the Lagrange multiplier and $C(v)$ is a function such that

$$C(v(\boldsymbol{x}))\Big|_{\boldsymbol{x}\in\Gamma} = 0, \tag{6.13}$$

and if we choose $C(v) = 2(v - g)$ then we have the final form of the functional as

$$\Pi(v) = \int_\Omega \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} v \, d\Omega - 2 \int_\Omega v f \, d\Omega + 2 \int_\Gamma \lambda (v - g) \, d\Gamma. \tag{6.14}$$

We can then discretise v with the MLS expansion from equation (3.17) and use the notation $\boldsymbol{v} = \begin{bmatrix} v_1, & v_2, \dots, & v_n \end{bmatrix}^\intercal$ and $\boldsymbol{\Phi} = \begin{bmatrix} \Phi_1, & \Phi_2, \dots, & \Phi_n \end{bmatrix}^\intercal$ as the vectors of nodal coefficients and shape functions, respectively, to give

$$\Pi(\boldsymbol{v}) = \int_\Omega \boldsymbol{\nabla} \left(\boldsymbol{v}^\intercal \boldsymbol{\Phi}\right) \cdot \boldsymbol{\nabla} \left(\boldsymbol{\Phi}^\intercal \boldsymbol{v}\right) d\Omega - 2 \int_\Omega \boldsymbol{v}^\intercal \boldsymbol{\Phi} f \, d\Omega + 2 \int_\Gamma \lambda (\boldsymbol{v}^\intercal \boldsymbol{\Phi} - g) \, d\Gamma, \tag{6.15}$$

with the variation of the functional then given as

$$\delta\Pi = 2 \int_\Omega \left(\boldsymbol{\nabla}\boldsymbol{\Phi} \cdot \boldsymbol{\nabla}\boldsymbol{\Phi}^\intercal\right) \boldsymbol{v} \, d\Omega - 2 \int_\Omega \boldsymbol{\Phi} f \, d\Omega + 2 \int_\Gamma \lambda \boldsymbol{\Phi} \, d\Gamma + 2 \int_\Gamma \delta\lambda (\boldsymbol{v}^\intercal \boldsymbol{\Phi} - g) \, d\Gamma. \tag{6.16}$$

The weak form of the problem is then found by solving for the stationary point $\delta\Pi(\mathbf{u}) = 0$, where the last term is set to zero independently such that the Dirichlet BCs are properly enforced to give the system

$$\int_\Omega (\boldsymbol{\nabla\Phi}\cdot\boldsymbol{\nabla\Phi}^T)\,\mathbf{u}\,d\Omega + \int_\Gamma \lambda\boldsymbol{\Phi}\,d\Gamma = \int_\Omega \boldsymbol{\Phi} f\,d\Omega, \tag{6.17a}$$

$$\int_\Gamma \delta\lambda(\mathbf{u}^T\boldsymbol{\Phi} - g)\,d\Gamma = 0. \tag{6.17b}$$

Comparing equation (6.17a) with the weak form found earlier in equations (3.10) and (3.11) of section 3.1.2, which can be re-written here in our current notation for ease of reading as

$$\int_\Omega (\boldsymbol{\nabla\Phi}\cdot\boldsymbol{\nabla\Phi}^T)\,\mathbf{u}\,d\Omega - \int_\Gamma \boldsymbol{\Phi}\boldsymbol{\nabla}u\cdot\hat{\mathbf{n}}\,d\Gamma = \int_\Omega f\boldsymbol{\Phi}\,d\Omega. \tag{6.18}$$

we can see that the physical interpretation of the Lagrange multiplier is the boundary flux $\lambda = -\boldsymbol{\nabla}u\cdot\hat{\mathbf{n}}$, which appears in the boundary term that arose from the use of integration by parts.

This weak form then gives rise to the augmented matrix equation

$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{q} \end{Bmatrix}, \tag{6.19}$$

where $\mathbf{K}$ and $\mathbf{f}$ are the standard operator matrix and forcing vector computed according to equations (3.7) and (3.11), respectively, $\boldsymbol{\lambda} = \left[\lambda_1,\ \lambda_2, \ldots,\ \lambda_\ell\right]^T$ is the vector of discrete Lagrange multipliers, and $\mathbf{G}$ and $\mathbf{q}$ are computed as

$$\begin{aligned} \mathbf{G}_{IK} &= \int_\Gamma \Phi_I N_K^L\,d\Gamma, \\ \mathbf{q}_K &= \int_\Gamma g N_K^L\,d\Gamma. \end{aligned} \tag{6.20}$$

These discrete Lagrange multipliers come from discretising λ according to its own expansion with an independent set of functions $\{N_K^L(\mathbf{x})\}_{K=1}^{\ell}$

$$\begin{aligned} \lambda(\mathbf{x}) &= \sum_{K=1}^{\ell} \lambda_K N_K^L(\mathbf{x}), \quad \mathbf{x}\in\Gamma, \\ \delta\lambda(\mathbf{x}) &= \sum_{K=1}^{\ell} \delta\lambda_K N_K^L(\mathbf{x}), \quad \mathbf{x}\in\Gamma. \end{aligned} \tag{6.21}$$

There are several common choices for the approximation functions N_K^L in the above expansion, namely:

1. a separate FEM interpolation or meshfree approximation on the boundary;
2. the same meshfree shape functions used for the interior discretisation restricted to the boundary;
3. delta functions $N_K^L(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{x}_K^L)$ to give a point collocation method enforcing the Dirichlet BCs exactly at a set of points $\{\mathbf{x}_K^L\}_{K=1}^{\ell}$.

This last choice is probably the most popular because of its simplicity, and is the choice we will use for that reason also. This gives the final contributions to the $\mathbf{G}$ and $\mathbf{q}$ matrices as

$$\begin{aligned} \mathbf{G}_{IK} &= \Phi_I(\mathbf{x}_K^L), \\ \mathbf{q}_K &= g(\mathbf{x}_K^L). \end{aligned} \tag{6.22}$$

Each of the equations in $\mathbf{Gu} = \mathbf{q}$ is then simply enforcing the condition that the sum of MLS shape functions must equal the BC at the given point $\mathbf{x}_K^L$, which for our simulations we simply take to be the set of nodes on the boundary. This is subtly different from the direct strong form collocation method described earlier in this section, as it maintains the information from the Galerkin discretisation of the operator in the matrix $\mathbf{K}$ and only appends the collocation constraints to the system, rather than overwriting entries of the original matrix. As noted earlier, this increases the size of the system and clearly introduces zeros on the diagonal, but it does maintain the consistency of the discretisation and so does not suffer from the suboptimal convergence observed with direct collocation.

6.4 Error bounds

As mentioned in section 4.4 the error bounds for MLS-based methods, of which this FCIMLS scheme is a direct example, are given by

$$\|u - u^d\| = \mathcal{O}(d^{k+1}), \tag{6.23}$$

where the convergence accuracy in the L^2 norm is one order higher than the polynomial order of the basis with respect to the dilation parameter.

However, it is worth noting that this bound is not entirely sharp in all cases. In particular, in [156] they found that convergence is usually better than this bound for uniform node distributions. Such uniformity is generally disrupted by boundaries, but can still be applicable in the interior of the domain even for non-periodic simulations. This superconvergence effect is clearly evident in the results of section 7.2.

6.5 Summary

In this chapter we explored the extension of our method to a fully meshfree scheme to allow for arbitrary order consistency. The MLS formalism is used to construct the basis functions using box-shaped supports that are warped along the mapping and elongated in the ζ-coordinate to leverage the anisotropy. Nodes are still arranged in planes to aid in ensuring domain cover and searching out nearby nodes. A suitable method for enforcing Dirichlet BCs must be chosen; in this thesis we use the method of Lagrange multipliers for its simplicity, but coupling to a FEM might be particularly worth consideration in future applications. In the next chapter we will show some preliminary simulation results obtained using this scheme.

CHAPTER 7

FCIMLS NUMERICAL RESULTS

SIMILARILY AS in chapter 5, all of the simulations below are also carried out on the 2D unit square domain $\Omega : (0,1) \times (0,1)$, where x and y are respectively associated with the ζ- and $x_\perp$-coordinates of the general FCIMLS scheme outlined above, and may be used interchangeably. All stationary Poisson simulations which use doubly periodic BCs have their nodal values on the x- and y-axes constrained using Lagrange multipliers similarly as was outlined for Dirichlet BCs in order to ensure the problem remains well-posed.

7.1 PATCH TESTS

Starting this time with the simple patch tests, we will now be able to compare both the same linear patch as used previously for the FCIFEM scheme in equation (5.7), as well as the quadratic patch

$$u(x,y) = g(x,y) = 0.5 + 0.1x + 0.8y + 1.2xy + 0.8x^2 + 0.6y^2, \tag{7.1}$$

giving the Poisson equation

$$\nabla^2 u = 2.8, \tag{7.2}$$

where again it is the enforcement of the Dirichlet BCs for g on the boundary $\Gamma = \delta\Omega$ which also enforce the specific quadratic solution we have selected.

Again looking primarily for which simulations achieve exact solutions to machine precision, the results are given in table 7.1 for various types of VCI correction with both the linear and quadratic patches, where the polynomial order of the MLS basis is matched with the order of the patch being simulated. All of the tests were carried out with the curved sinusoidal mapping from equation (5.3) as the use of the full MLS formalism for constructing the shape functions allows for exact representation of the desired polynomial order even for such curved mappings, unlike in the FCIFEM case of section 5.2.1.

VCI	Linear patch	Quadratic patch
None	1.74×10^{-3}	8.52×10^{-4}
AS - linear	6.45×10^{-15}	2.43×10^{-4}
AS - quadratic	6.63×10^{-15}	1.29×10^{-14}
CQR - linear	6.32×10^{-15}	4.53×10^{-4}

TABLE 7.1: L^2 norms of error for linear and quadratic patch tests with the FCIMLS scheme showing effect of VCI correction, where AS indicates assumed strain method and CQR indicates our conservative quadrature reweighting scheme. A linear MLS basis was used for linear patch tests and quadratic MLS basis for quadratic patch tests. All tests were carried out with $N_x = N_y = 8$ grid divisions in each direction with 10 % node perturbation, using the sinusoidal mapping function from equation (5.3) and cubic weight function from equation (3.18).

We can clearly see that in both the cases with no VCI correction or only linear correction for the quadratic patch the solutions are not exact to machine precision, while all simulations with matching- or higher-order VCI correction do achieve machine precision results. This is entirely as expected, and provides an initial verification check on the implementation of the various aspects of the code for the new FCIMLS scheme. It is perhaps worth noting that only results for linear correction with our conservative quadrature reweighting scheme are shown here because of insufficient time for implementing the quadratic correction with Dirichlet BCs, while quadratic reweighting correction is used in some of the following numerical experiments that only use periodic BCs and therefore did not require implementation of the extra boundary integrals required for the quadratic constraints given in equation (4.17c).

7.2 REPRESENTATION ACCURACY

We can then proceed with verifying the representation power of the proposed basis by computing the L^2 projection of a smooth function into our FCIMLS basis function space, with figure 7.1 showing the results of our convergence studies. As mentioned in section 5.1, this simply means we are solving the equation $u(x, y) = f(x, y)$ using the FCIMLS scheme with a standard Bubnov-Galerkin approach. In the left column of figure 7.1 we are approximating

$$f(x, y) = \sin(2\pi x) \sin(2\pi y), \tag{7.3}$$

with doubly periodic BCs and our usual sinusoidal mapping function given earlier in equation (5.3), while in the right column we are approximating

$$\begin{gathered} f(x, y) = x \sin\left(2\pi n[y - ax^2 - bx]\right), \\ n = 3, \quad a = 0.95, \quad b = 0.05, \end{gathered} \tag{7.4}$$

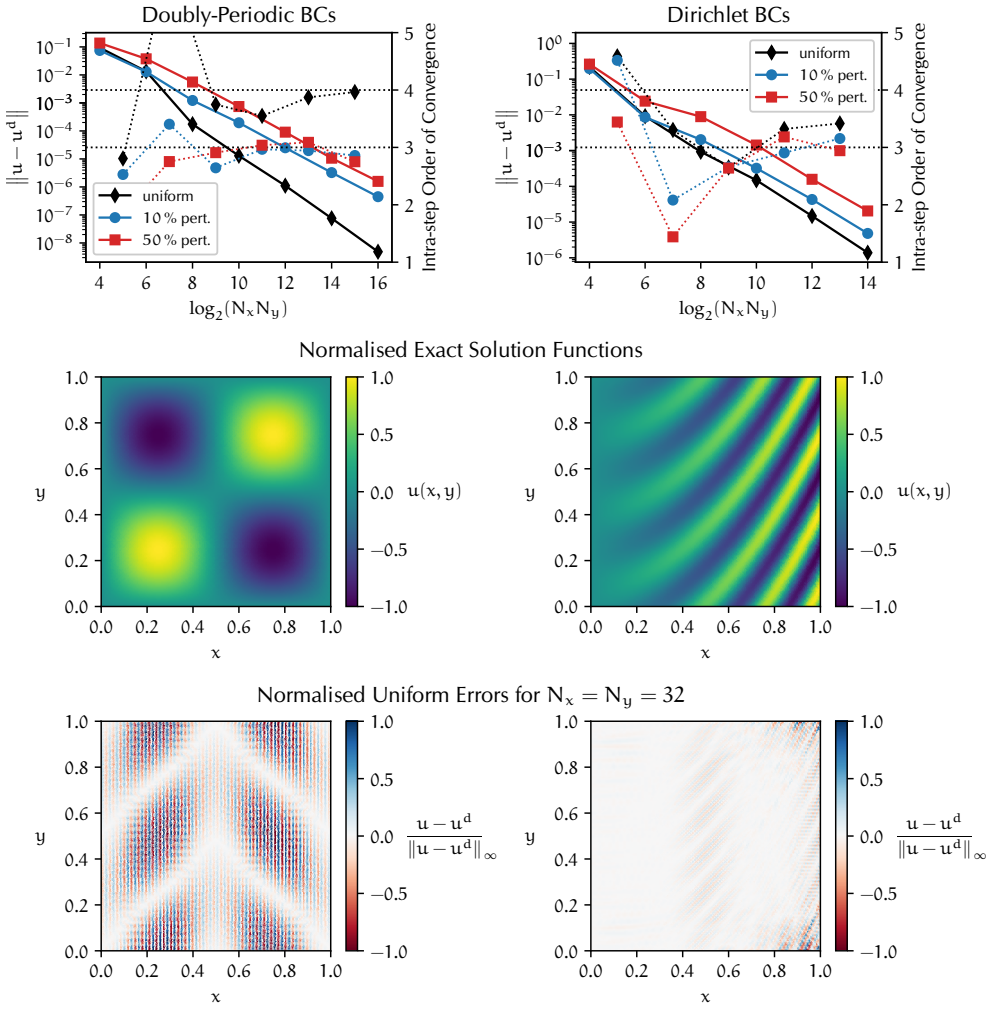


FIGURE 7.1: Convergence study of L^2 projection for FCIMLS, where left column uses periodic BCs and sinusoidal mapping while right column uses Dirichlet BCs with quadratic mapping. *Top row* shows L^2 norms of error (solid lines, left y-axes) and convergence orders (dotted lines, right y-axes) for L^2 projections of smooth functions into the FCIMLS function space. Data for uniform node spacing is shown in black with diamond markers, for node locations randomly perturbed by up to 10 % of the uniform spacing in blue with circle markers, and up to 50 % in red with square markers. *Middle row* shows the normalised analytic functions being approximated. *Bottom row* shows the normalised errors for simulations using a uniform node distribution with $N_x = N_y = 32$ grid divisions in each direction. Localisation of error at the boundaries in the Dirichlet BC case is evident, which deteriorates the superconvergence compared to the periodic BC case.

with fully Dirichlet BCs and using the quadratic mapping of equation (5.11) aligned to the anisotropy. Visualisations of these functions are shown in the middle row of figure 7.1. A $1:1$ ratio of the number of grid divisions $N_x : N_y$ is used for all simulations along with a quadratic MLS basis and the cubic spline weight function from equation (3.18).

For the periodic simulations a support size of $2.5X$ the uniform grid spacing in each direction is used, while the Dirichlet simulations use a larger box of $3X$ the uniform spacing. This is because the periodic BCs mean that the shape function coverage is relatively consistent throughout the entire domain, while at points close to a Dirichlet boundary there is a drop in coverage because there are no nodes outside of the domain and therefore less shape functions within the support of such on- or near-boundary points. This is doubly true near corners in the domain where points are near two such boundaries simultaneously. While not implemented for this thesis due to time constraints, one approach to ameliorating this problem would be to allow variation in support sizes between nodes, and make the supports larger for near- but off-boundary nodes, such that additional shape functions reach to the edges of the domain and maintain sufficient cover there. Additionally, we note that at least for topologically toroidal MCF devices, corners would hopefully not be an insurmountable issue because because the toroidal periodicity means at least that one dimension is always free of Dirichlet boundaries.

The expected third-order error bound from section 6.4 is achieved for all simulations, but, as noted in that section, simulations with uniform node distributions also exhibit superconvergence, as predicted in the work of [156]. For the periodic test fully fourth-order convergence is observed in the uniform case while in the Dirichlet test the uniform case exhibits convergence somewhere between fourth and third order. The reason for this is because the Dirichlet BCs disrupt the uniform ordering and in the bottom plots in figure 7.1 we can see that the error in the Dirichlet case is noticeably larger near the boundaries compared to the domain interior while in the periodic case it is spread throughout the domain. This gives qualitative evidence that the error in the interior of the domain for the Dirichlet case is likely still converging with fourth-order accuracy as in the periodic case while near the boundary this is reduced to the third-order convergence expected from the general error bound for a quadratic MLS basis.

Overall, the tests confirm that at minimum third-order convergence is achievable for our quadratic FCIMLS scheme for all nodal arrangements and with curvature present in the mapping functions.

7.3 Effects of support size

Obviously the single most important consideration in choosing the support size is ensuring sufficient domain coverage, as otherwise the moment matrix is not

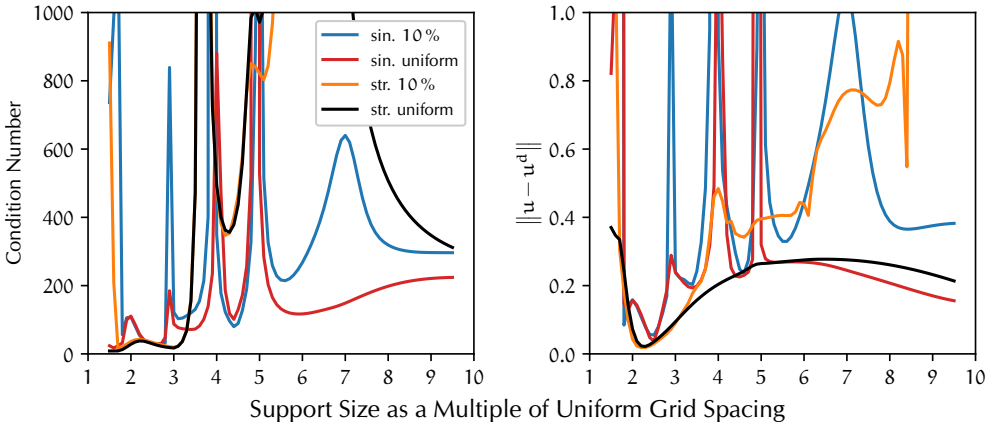


FIGURE 7.2: (left) Condition numbers and (right) L^2 norms of error for Poisson simulations of a smooth function with doubly periodic BCs using $N_x = N_y = 16$ grid divisions in each direction, comparing the use of sinusoidal (sin.) mapping and coordinate-aligned (str.) mapping with both uniform node distributions and nodes perturbed up to 10 % of uniform spacing.

invertible and the simulation will simply fail. However, this essentially only sets a minimum limit on the size, and so it is worth asking what the effects are of variations in the support size above this limit. As mentioned previously, from the view of computational efficiency it is desirable to keep the chosen supports as small as possible, as increasing the size deteriorates the local nature of the shape functions and leads to a denser, and also generally less-banded, operator matrix.

Therefore, a series of simulations were carried out solving a Poisson problem using the forcing function visualised in the left plot of the middle row in figure 7.1 to give

$$\nabla^2 u = \sin(2\pi x)\sin(2\pi y), \tag{7.5}$$

with doubly periodic BCs and comparing both the equation (5.3) sinusoidal mapping with a coordinate-aligned mapping, and uniform versus 10 % perturbed node distributions. All simulations were performed using a quadratic MLS basis and the quintic weight function from equation (3.20). The data from these simulations is presented in figure 7.2 with the left graph giving the condition numbers of the resulting operator matrices and the right graph giving the L^2 norms of the error for the simulations. All values are plotted as functions of the support size, which is normalised to the uniform grid spacing and is equal in both directions of the box. A normalised support size of 1.5 is the minimum limit for sufficient cover, with all smaller values causing the simulations to fail, and so this is the size which all the data series in the figure start from.

We note that the simulations all used $N_x = N_y = 16$ grid divisions in both coordinate directions, and so a normalised support size of 8 corresponds to the size at which the supports would cover the full domain, and thus larger sizes would actually have the shape functions wrapping back onto themselves at the edges of their support.

Such sizes would of course be entirely impractical because of the aforementioned computational efficiency concerns, but it is still interesting to note that at no point do even such large sizes cause the simulations to simply fail, although both the condition numbers and error norms are significantly larger than the corresponding values near the optimal support sizes.

There is a fair bit of variation in the behaviour of the different data series, but some general features can still be gleaned. Both condition numbers and error norms have minima at sizes < 3 and trend higher as the size increases above this point. There are also spikes in many of the data series that correspond to integer sizes, which are the points at which the nodes of an additional FCI plane would be just at the edge of the supports. This could imply that at least in the parallel coordinate direction it may be preferable to avoid integer support sizes, while in the perpendicular directions this should be less of an issue for general unstructured node arrangements, as there would not be similar size thresholds where such large step changes occur in the number of nodes covering given areas.

As well, the error norms (and condition numbers for the perturbed cases) are noticeably suboptimal near the minimal support size of 1.5, and one would anyway want to have some buffer from this threshold to avoid the risk of falling below it and having the simulation fail. Thus, at least for problems with periodic BCs, a size around 2.5 seems relatively optimal for avoiding the problems near integer sizes whilst simultaneously staying close to the true optimal points for the condition numbers and error norms. As noted previously, Dirichlet BCs may require larger support sizes near the boundaries to ensure sufficient coverage there.

7.4 Periodic anisotropic Poisson problem

We can then test the new FCIMLS scheme with the Poisson problem from [117] using the forcing term given in equation (5.9) but reproduced here for ease of reading as

$$\nabla^2 u = \frac{1}{2} \sin\left(2\pi n[y - x]\right)\left[1 + \sin\left(2\pi y\right)\right], \tag{7.6}$$

with $n = 8$ to create non-grid-aligned anisotropy along the $\begin{bmatrix}1, & 1\end{bmatrix}^T$ diagonal direction. The normalised analytic solution to this problem is shown in the bottom plot of figure 7.3, clearly showing the strong non-coordinate aligned anisotropy. As previously, we use doubly periodic BCs, a quadratic MLS basis, and the quintic weight function from equation (3.20).

Results for these simulations are shown in figure 7.3 where the error convergence for all configurations is seen to cluster around third-order accuracy in the asymptotic region and with all simulations using a node distribution perturbed by 10 %. One starts with a baseline simulation using a standard grid-aligned basis but which is *unaligned* with respect to the anisotropy, with an equal $1:1$ number of divisions $N_x : N_y$. This is compared to four other simulations which all use a mapping *aligned*

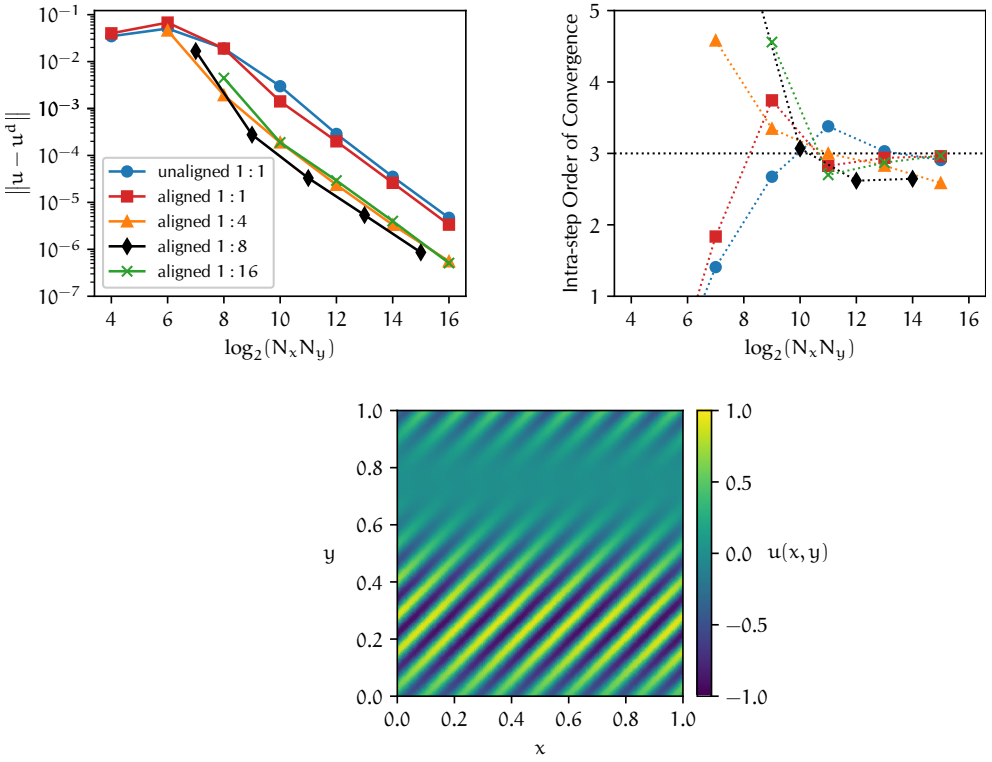


Figure 7.3: Convergence study of Poisson simulations with strong non-coordinate aligned anisotropy, comparing effects of aligning basis to anisotropy and changing $N_x : N_y$ ratio of grid spacings in coordinate directions. (top left) L^2 norms of error and (top right) convergence orders, with all simulations using a grid distribution of nodes perturbed up to 10 % of the uniform spacing. (bottom) Normalised exact solution function.

to the anisotropy, with ratios of 1 : 1, 1 : 4, 1 : 8, and 1 : 16 grid divisions between the x- and y-coordinate directions, as the x-direction corresponds to the ζ-coordinate of the FCIMLS scheme. As with the FCIFEM we can see that the larger number of small divisions in the y-direction are effectively capturing the small-scale, large-gradient variations perpendicular to the anisotropy, while large divisions in the x-direction sufficiently capture the large-scale, small-gradient variation along the anisotropy. In all cases the alignment of the basis functions to the anisotropy improves the achieved accuracy for a given number of nodes in the simulation, but it is the reduction in the required number of DoFs for a given level of accuracy which is particularly valuable, with the most optimal $N_x : N_y$ ratio being 1 : 8, as expected, since this matches the $n = 8$ anisotropy in the wavenumbers defined in the forcing term.

7.5 Summary

In this chapter we show preliminary simulation results using the FCIMLS scheme. Successful patch tests confirm the consistency inherent in using the MLS formalism for both linear and quadratic patches and now also with curved mappings, in

contrast to the FCIFEM scheme. L^2 projection tests verify the expected error convergence for simple representations of smooth functions, including the theoretically predicted superconvergence in specific cases. A quick parameter scan over support sizes demonstrated the sometimes finicky nature of selecting such parameters in meshfree methods, where future applications might need to investigate heuristics for robustly determining good values. A periodic Poisson simulation with moderate anisotropy provides initial validation of the scheme for reducing the number of DoFs required for a given level of accuracy, but further work implementation work would be needed for a robust set-up to test a similar problem with Dirichlet BCs.

CHAPTER 8

Conclusion

Two related methods have been presented for representing fields containing structures highly elongated along a known directional field, but which need not be aligned to any coordinate direction of the simulation. The directional information is encoded by a mapping function, as proposed for previous FCI methods based on FDMs, but which is then used here to define basis functions providing a high degree of anisotropy. In the first FCIFEM scheme the basis functions are constructed in a manner so as to be analogous with linear FEM hat functions. In the second FCIMLS scheme the meshfree technique of MLS is employed to allow computation of bases providing arbitrarily high-order polynomial consistency.

These highly anisotropic bases are then used in a Galerkin formalism to discretise PDEs of interest, with modifications from the standard FEM setup to incorporate concepts from meshfree methods which allow arbitrary overlap of the basis functions between distantly separated planes of nodes. The use of integration by parts during the discretisation is shown to provide a local flux conservation property for the methods, which should also be applicable to other meshfree methods using similar Galerkin constructions.

Also similarly to other meshfree schemes, accurate quadrature is found to be problematic for non-uniformly distributed nodes, and so a reweighting scheme for the quadrature points is demonstrated which provides excellent recovery of the expected convergence orders of both FCIFEM and FCIMLS schemes while preserving the local conservation property of the methods.

Numerical testing results were presented which illustrate the ability of both schemes to represent smooth functions with various mappings and with either periodic or Dirichlet BCs. Results demonstrating the ability to discretise differential operators were also presented, using both periodic and Dirichlet BCs for the FCIFEM scheme but only periodic BCs for the FCIMLS scheme due to time constraints

preventing the inclusion of Dirichlet BCs results for the FCIMLS scheme before the writing of this thesis.

The expected second-order convergence for the linear basis in the FCIFEM scheme and third-order convergence for the quadratic basis in the FCIMLS scheme were consistently achieved in all test problems. Superconvergence to fourth-order accuracy in specific cases for the FCIMLS scheme was also observed, which is as predicted by the theory.

For toy problems with moderate anisotropy a significant reduction is achieved in the number of DoFs required to reach a given level of accuracy. This provides proof of concept for greatly reducing the computational resources required for problems such as tokamak simulations where the degree of anisotropy is often several orders of magnitude.

Extension of the code for 3D simulations was not carried out due to time constraints and because it was not the primary focus of this work, but we would refer the interested reader to the 3D example of the preceding work by McMillan [117] the results of which would be expected to hold quite well for the schemes proposed here, given the excellent convergence results obtained and agreement of the 2D test problems.

In particular, a big question for MCF simulations when moving to 3D is what effect the strong magnetic shear would have, as this can cause serious issues for truly field-aligned meshes; a flux tube that starts out with a regular cross-section becomes progressively more sheared and warped as you travel around the torus, which also leads to questions of how to reconnect with the original mesh after complete revolutions.

Conversely, for our proposed schemes (and other FCI-based methods) the planes are independent of each other, so the shear does not accumulate but rather is reset at each new plane. Thus, the basis functions themselves are only warped over the distance to the nearest neighbouring planes or length of their parallel support, which is quite small compared to the full toroidal domain length, so the overall distortion effect should also be comparatively small and non-problematic.

8.1 Directions for future investigation

As mentioned, the implementation and testing of the FCIMLS scheme for achieving higher-order accuracy is still very much in its preliminary stages, and further work would be needed to more fully characterise the properties of the proposed method, particularly for properly and efficiently incorporating Dirichlet BCs and testing the quadrature reweighting correction.

An initial implementation of the Dirichlet BCs was coded using the method of Lagrange multipliers described in section 6.3, and used for the L^2 projection and patch tests in sections 7.1 and 7.2, but during testing it proved difficult to replicate the

test problem used in section 5.4. This was largely due to the bottom right corner of the simulation where the shape of the mapping made it extremely difficult to ensure adequate coverage as the spacing between nodal planes was increased. In a future implementation I would expect blending with FEM at Dirichlet boundaries to likely be a more practical solution for two reasons: firstly, it would allow one to simply use a standard FEM scheme to deal with such small problem areas or other fine features of the boundary geometry where the anisotropic assumptions are anyway violated, and secondly, extra MLS nodes could be embedded within the FEM layer to ameliorate the problem of decreased nodes being present as you approach the boundary, as noted in section 7.2.

For both schemes, implementation of the described methods into a proper 3D code framework in order to test more reactor relevant problems and confirm the ability to leverage the highly anisotropic bases for reducing computational cost would be an important avenue for future research.

One additional complexity that must be addressed by future implementations of these method in 3D will be locating which element of the unstructured FEM mesh on the FCI planes or boundaries a given point is located in, or which FCIMLS nodes have non-zero support there. These operations would be needed when mapping from quadrature or evaluation points in the interior of the domain, where one will then need to determine which element contains the mapped point or which FCIMLS nodes contribute non-zero basis functions at the original point. Further investigation would be required to determine efficient algorithms for carrying out this search step. Other implementation challenges would largely be similar as those already encountered anyway in either standard FEM schemes or FDM-based FCI schemes.

Additionally, this work serves largely as a proof of concept that meshfree ideas could be utilised to avoid meshing difficulties when dealing with strong anisotropy, particularly in MCF simulations. It also elucidates and begins to address some of the initial challenges that arise in doing so. However, it does not purport to provide the optimal scheme for such simulations, and future investigations and collaborations with researchers from the meshfree modelling community would be valuable for determining other techniques and ideas which could be beneficial and used to build on and refine the methods proposed here.

Testing of both schemes on truly turbulent simulations with energy cascades would be necessary to validate that the method is fit for its intended purpose. How to deal with concerns such as overshoot oscillations in advection-diffusion type problems would also merit specific investigation, as they occur in many plasma models of interest. For example, techniques such as upwinding for FEM methods [170–175] might be adaptable to provide the necessary local dissipation.

Bibliography

1. Maloney, S. A. & McMillan, B. F. A partially mesh-free scheme for representing anisotropic spatial variations along field lines: Conservation, quadrature, and the delta property. *Computer Physics Communications* **284,** 108629. doi:10.1016/j.cpc.2022.108629 (Mar. 2023).
2. Lee, H. *et al. Synthesis Report of the IPCC Sixth Assessment Report (AR6)* tech. rep. (The Intergovernmental Panel on Climate Change, 2023). www.ipcc.ch/report/sixth-assessment-report-cycle/.
3. Bilgen, S. Structure and environmental impact of global energy consumption. *Renewable and Sustainable Energy Reviews* **38,** 890–902. doi:10.1016/j.rser.2014.07.004 (Oct. 2014).
4. Mitrašinović, A. M. Photovoltaics advancements for transition from renewable to clean energy. *Energy* **237,** 121510. doi:10.1016/j.energy.2021.121510 (Dec. 2021).
5. Darwish, A. S. & Al-Dabbagh, R. Wind energy state of the art: present and future technology advancements. *Renewable Energy and Environmental Sustainability* **5,** 7. doi:10.1051/rees/2020003 (Apr. 2020).
6. Schmietendorf, K., Peinke, J. & Kamps, O. The impact of turbulent renewable energy production on power grid stability and quality. *The European Physical Journal B* **90,** 222. doi:10.1140/epjb/e2017-80352-8 (Nov. 2017).
7. Frisch, W. & Gros, G. Improving the safety of future nuclear fission power plants. *Fusion Engineering and Design* **56-57,** 83–93. doi:10.1016/S0920-3796(01)00238-1 (Oct. 2001).
8. Marques, J. Evolution of nuclear fission reactors: Third generation and beyond. *Energy Conversion and Management* **51,** 1774–1780. doi:10.1016/j.enconman.2009.12.043 (Sept. 2010).

9. Kelly, J. E. Generation IV International Forum: A decade of progress through international cooperation. *Progress in Nuclear Energy* **77,** 240–246. doi:10.1016/j.pnucene.2014.02.010 (Nov. 2014).

10. Canadian Nuclear Safety Commission. *Facts about tritium* Nov. 2021. https://nuclearsafety.gc.ca/eng/resources/fact-sheets/tritium.cfm.

11. Raeder, J. Report on the European Safety and Environmental Assessment of Fusion Power (SEAFP). *Fusion Engineering and Design* **29,** 121–140. doi:10.1016/0920-3796(95)80014-0 (Mar. 1995).

12. Pfalzner, S. *An Introduction to Inertial Confinement Fusion* 1st ed. doi:10.1201/9781420011845 (CRC Press, Mar. 2006).

13. Stacey, W. M. *Fusion: An Introduction to the Physics and Technology of Magnetic Confinement Fusion* 2nd ed. ISBN: 978-3-527-40967-9 (John Wiley & Sons, 2010).

14. Lawson, J. D. Some Criteria for a Power Producing Thermonuclear Reactor. *Proceedings of the Physical Society. Section B* **70,** 6–10. doi:10.1088/0370-1301/70/1/303 (Jan. 1957).

15. Ongena, J., Koch, R., Wolf, R. & Zohm, H. Magnetic-confinement fusion. *Nature Physics* **12,** 398–410. doi:10.1038/nphys3745 (May 2016).

16. Sykes, A. Progress on spherical tokamaks. *Plasma Physics and Controlled Fusion* **36,** B93–B106. doi:10.1088/0741-3335/36/12B/007 (Dec. 1994).

17. Windridge, M. Smaller and quicker with spherical tokamaks and high-temperature superconductors. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **377,** 20170438. doi:10.1098/rsta.2017.0438 (Mar. 2019).

18. Spitzer, L. The Stellarator Concept. *Physics of Fluids* **1,** 253. doi:10.1063/1.1705883 (July 1958).

19. Xu, Y. A general comparison between tokamak and stellarator plasmas. *Matter and Radiation at Extremes* **1,** 192–200. doi:10.1016/j.mre.2016.07.001 (July 2016).

20. Abu-Shawareb, H. *et al.* Lawson Criterion for Ignition Exceeded in an Inertial Fusion Experiment. *Physical Review Letters* **129,** 075001. doi:10.1103/PhysRevLett.129.075001 (Aug. 2022).

21. Seaton, A. G. & Arber, T. D. Laser-plasma instabilities in long scale-length plasmas relevant to shock-ignition. *Physics of Plasmas* **27,** 082704. doi:10.1063/5.0010920 (Aug. 2020).

22. Chen, F. F. *Introduction to Plasma Physics and Controlled Fusion* 3rd ed. doi:10.1007/978-3-319-22309-4 (Springer International Publishing, Cham, Dec. 2016).

23. Gurnett, D. A. & Bhattacharjee, A. *Introduction to Plasma Physics: With Space, Laboratory and Astrophysical Applications* 2nd ed. doi:10.1017/9781139226059 (Cambridge University Press, Feb. 2017).

24. Baumjohann, W. & Treumann, R. A. *Basic Space Plasma Physics* 1st ed. doi:10.1142/p015 (Imperial College Press, London, Sept. 1997).

25. Aymar, R., Barabaschi, P. & Shimomura, Y. The ITER design. *Plasma Physics and Controlled Fusion* **44,** 519–565. doi:10.1088/0741-3335/44/5/304 (May 2002).

26. Landau, L. D. in *Collected Papers of L.D. Landau* (ed Ter Haar, D.) 1st ed., 445–460 (Pergamon Press, Oxford, Jan. 1965). ISBN: 9781483152707. https://shop.elsevier.com/books/collected-papers-of-ld-landau/ter-haar/978-0-08-010586-4.

27. Liewer, P. C. Measurements of microturbulence in tokamaks and comparisons with theories of turbulence and anomalous transport. *Nuclear Fusion* **25,** 543–621. doi:10.1088/0029-5515/25/5/004 (May 1985).

28. Horton, W. Drift waves and transport. *Reviews of Modern Physics* **71,** 735–778. doi:10.1103/RevModPhys.71.735 (Apr. 1999).

29. Garbet, X., Idomura, Y., Villard, L. & Watanabe, T. Gyrokinetic simulations of turbulent transport. *Nuclear Fusion* **50,** 043002. doi:10.1088/0029-5515/50/4/043002 (Apr. 2010).

30. Ottaviani, M. An alternative approach to field-aligned coordinates for plasma turbulence simulations. *Physics Letters A* **375,** 1677–1685. doi:10.1016/j.physleta.2011.02.069 (Apr. 2011).

31. Braginskii, S. I. in *Review of Plasma Physics* (ed Leontovich, M. A.) 205–311 (Consultants Bureau, New York, 1965).

32. Hender, T. *et al.* Chapter 3: MHD stability, operational limits and disruptions. *Nuclear Fusion* **47,** S128–S202. doi:10.1088/0029-5515/47/6/S03 (June 2007).

33. D'haeseleer, W. D., Hitchon, W. N. G., Callen, J. D. & Shohet, J. L. *Flux Coordinates and Magnetic Field Structure* 1st ed. (eds Glowinski, R. *et al.*) doi:10.1007/978-3-642-75595-8 (Springer, Berlin, Heidelberg, 1991).

34. Zhang, F. *et al.* Mesh generation for confined fusion plasma simulation. *Engineering with Computers* **32,** 285–293. doi:10.1007/s00366-015-0417-y (Apr. 2016).

35. Hamada, S. Notes on Megneto-Hydrodynamic Equilibrium. *Progress of Theoretical Physics* **22,** 145–146. doi:10.1143/PTP.22.145 (July 1959).

36. Hamada, S. Hydromagnetic equilibria and their proper coordinates. *Nuclear Fusion* **2,** 23–37. doi:10.1088/0029-5515/2/1-2/005 (Jan. 1962).

37. Greene, J. M. & Johnson, J. L. Stability Criterion for Arbitrary Hydromagnetic Equilibria. *Physics of Fluids* **5,** 510. doi:10.1063/1.1706651 (May 1962).

38. Boozer, A. H. Guiding center drift equations. *Physics of Fluids* **23,** 904. doi:10.1063/1.863080 (May 1980).

39. Boozer, A. H. Plasma equilibrium with rational magnetic surfaces. *Physics of Fluids* **24,** 1999. doi:10.1063/1.863297 (Nov. 1981).

40. Poincaré, H. Mémoire sur les courbes définies par une équation différentielle (I). *Journal de Mathématiques Pures et Appliquées, Série 3* **7,** 375–422. http://www.numdam.org/item/JMPA_1881_3_7__375_0/ (1881).

41. Poincaré, H. Mémoire sur les courbes définies par une équation différentielle (II). *Journal de Mathématiques Pures et Appliquées, Série 3* **8,** 251–296. http://www.numdam.org/item/JMPA_1882_3_8__251_0/ (1882).

42. Poincaré, H. Sur les courbes définies par les équations différentielles (III). *Journal de Mathématiques Pures et Appliquées, Série 4* **1,** 167–244. http://www.numdam.org/item?id=JMPA_1885_4_1__167_0 (1885).

43. Poincaré, H. Sur les courbes définies par les équations différentielles (IV). *Journal de Mathématiques Pures et Appliquées, Série 4* **2,** 151–217. http://www.numdam.org/item/JMPA_1886_4_2__151_0/ (1886).

44. Hopf, H. Abbildungsklassen n-dimensionaler Mannigfaltigkeiten. *Mathematische Annalen* **96,** 209–224. doi:10.1007/BF01209163 (Dec. 1927).

45. Hopf, H. Vektorfelder in n-dimensionalen Mannigfaltigkeiten. *Mathematische Annalen* **96,** 225–249. doi:10.1007/BF01209164 (Dec. 1927).

46. Brouwer, L. E. J. Über Abbildung von Mannigfaltigkeiten. *Mathematische Annalen* **71,** 97–115. doi:10.1007/BF01456931 (Mar. 1911).

47. Hudson, S. R., Startsev, E. & Feibush, E. A new class of magnetic confinement device in the shape of a knot. *Physics of Plasmas* **21,** 010705. doi:10.1063/1.4863844 (Jan. 2014).

48. Hazeltine, R. D. & Meiss, J. D. *Plasma Confinement* ISBN: 9780486151038. https://store.doverpublications.com/0486151034.html (Dover Publications, Inc., Mineola, N.Y., 2003).

49. Boozer, A. H. Physics of magnetically confined plasmas. *Reviews of Modern Physics* **76,** 1071–1141. doi:10.1103/RevModPhys.76.1071 (Jan. 2005).

50. Wesson, J. *Tokamaks* 4th ed. ISBN: 9780199592234. https://global.oup.com/academic/product/tokamaks-9780199592234 (Oxford University Press, Dec. 2011).

51. Hariri, F. *FENICIA: a generic plasma simulation code using a flux-independent field-aligned coordinate approach* PhD thesis (Aix-Marseille University, Marseille, France, Nov. 2013). http://www.theses.fr/2013AIXM4760.

52. Hariri, F. & Ottaviani, M. A flux-coordinate independent field-aligned approach to plasma turbulence simulations. *Computer Physics Communications* **184,** 2419–2429. doi:10.1016/j.cpc.2013.06.005 (Nov. 2013).

53. Scott, B. Global consistency for thin flux tube treatments of toroidal geometry. *Physics of Plasmas* **5,** 2334–2339. doi:10.1063/1.872907 (June 1998).

54. Cowley, S. C., Kulsrud, R. M. & Sudan, R. Considerations of ion-temperature-gradient-driven turbulence. *Physics of Fluids B: Plasma Physics* **3,** 2767–2782. doi:10.1063/1.859913 (Oct. 1991).

55. Hammett, G. W., Beer, M. A., Dorland, W., Cowley, S. C. & Smith, S. A. Developments in the gyrofluid approach to Tokamak turbulence simulations. *Plasma Physics and Controlled Fusion* **35,** 973–985. doi:10.1088/0741-3335/35/8/006 (Aug. 1993).

56. Dimits, A. M. Fluid simulations of tokamak turbulence in quasiballooning coordinates. *Physical Review E* **48,** 4070–4079. doi:10.1103/PhysRevE.48.4070 (Nov. 1993).

57. Beer, M. A., Cowley, S. C. & Hammett, G. W. Field-aligned coordinates for nonlinear simulations of tokamak turbulence. *Physics of Plasmas* **2,** 2687–2700. doi:10.1063/1.871232 (July 1995).

58. Kim, J. Y. & Wakatani, M. Radial Structure of High-Mode-Number Toroidal Modes in General Equilibrium Profiles. *Physical Review Letters* **73,** 2200–2203. doi:10.1103/PhysRevLett.73.2200 (Oct. 1994).

59. Connor, J. W., Astie, R. J. H. & Taylor, J. B. High mode number stability of an axisymmetric toroidal plasma. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* **365,** 1–17. doi:10.1098/rspa.1979.0001 (Feb. 1979).

60. Scott, B. Shifted metric procedure for flux tube treatments of toroidal geometry: Avoiding grid deformation. *Physics of Plasmas* **8,** 447–458. doi:10.1063/1.1335832 (Feb. 2001).

61. Dannert, T. & Jenko, F. Gyrokinetic simulation of collisionless trapped-electron mode turbulence. *Physics of Plasmas* **12,** 072309. doi:10.1063/1.1947447 (July 2005).

62. Jenko, F., Dorland, W., Kotschenreuther, M. & Rogers, B. N. Electron temperature gradient driven turbulence. *Physics of Plasmas* **7,** 1904–1910. doi:10.1063/1.874014 (May 2000).

63. Xanthopoulos, P. & Jenko, F. Clebsch-type coordinates for nonlinear gyrokinetics in generic toroidal configurations. *Physics of Plasmas* **13,** 092301. doi:10.1063/1.2338818 (Sept. 2006).

64. Ball, J. & Brunner, S. A non-twisting flux tube for local gyrokinetic simulations. *Plasma Physics and Controlled Fusion* **63,** 064008. doi:10.1088/1361-6587/abf8f4 (June 2021).

65. Görler, T. *et al.* The global version of the gyrokinetic turbulence code GENE. *Journal of Computational Physics* **230,** 7053–7071. doi:10.1016/j.jcp.2011.05.034 (Aug. 2011).

66. Lapillonne, X. *Local and Global Eulerian Gyrokinetic Simulations of Microturbulence in Realistic Geometry with Applications to the TCV Tokamak* PhD thesis (École Polytechnique Fédérale de Lausanne, Lausanne, Apr. 2010). doi:10.5075/epfl-thesis-4684.

67. Merlo, G. *Flux-tube and global grid-based gyrokinetic simulations of plasma microturbulence and comparisons with experimental TCV measurements* PhD thesis (École Polytechnique Fédérale de Lausanne, Lausanne, July 2016). doi:10.5075/epfl-thesis-7065.

68. Pan, Q., Told, D., Shi, E. L., Hammett, G. W. & Jenko, F. Full-f version of GENE for turbulence in open-field-line systems. *Physics of Plasmas* **25,** 062303. doi:10.1063/1.5008895 (June 2018).

69. Michels, D., Stegmeir, A., Ulbl, P., Jarema, D. & Jenko, F. GENE-X: A full-f gyrokinetic turbulence code based on the flux-coordinate independent approach. *Computer Physics Communications* **264,** 107986. doi:10.1016/j.cpc.2021.107986 (July 2021).

70. Rognlien, T. D. *et al. Advances in Understanding Tokamak Edge/Scrape-Off Layer Transport* in *Proceedings of the 23rd IAEA Fusion Energy Conference* (IAEA, Daejeon, Oct. 2010), 1–8. www-naweb.iaea.org/napc/physics/FEC/FEC2010/html/node272.htm.

71. Dorf, M. A. & Dorr, M. Continuum gyrokinetic simulations of edge plasmas in single-null geometries. *Physics of Plasmas* **28,** 032508. doi:10.1063/5.0039169 (Mar. 2021).

72. Huysmans, G. & Czarny, O. MHD stability in X-point geometry: simulation of ELMs. *Nuclear Fusion* **47,** 659–666. doi:10.1088/0029-5515/47/7/016 (July 2007).

73. Czarny, O. & Huysmans, G. Bézier surfaces and finite elements for MHD simulations. *Journal of Computational Physics* **227,** 7423–7445. doi:10.1016/j.jcp.2008.04.001 (Aug. 2008).

74. Hoelzl, M. *et al.* The JOREK non-linear extended MHD code and applications to large-scale instabilities and their control in magnetically confined fusion plasmas. *Nuclear Fusion* **61,** 065001. doi:10.1088/1741-4326/abf99f (June 2021).

75. Nikulsin, N. *et al.* JOREK3D: An extension of the JOREK nonlinear MHD code to stellarators. *Physics of Plasmas* **29.** doi:10.1063/5.0087104 (June 2022).

76. Jolliet, S. *et al.* A global collisionless PIC code in magnetic coordinates. *Computer Physics Communications* **177,** 409–425. doi:10.1016/j.cpc.2007.04.006 (Sept. 2007).

77. Lanti, E. *et al.* Orb5: A global electromagnetic gyrokinetic code using the PIC approach in toroidal geometry. *Computer Physics Communications* **251,** 107072. doi:10.1016/j.cpc.2019.107072 (June 2020).

78. McMillan, B. F. *et al.* Long global gyrokinetic simulations: Source terms and particle noise control. *Physics of Plasmas* **15,** 052308. doi:10.1063/1.2921792 (May 2008).

79. Vernay, T. *et al.* Synergy between ion temperature gradient turbulence and neoclassical processes in global gyrokinetic particle-in-cell simulations. *Physics of Plasmas* **19,** 042301. doi:10.1063/1.3699189 (Apr. 2012).

80. Sonnendrücker, E., Wacher, A., Hatzky, R. & Kleiber, R. A split control variate scheme for PIC simulations with collisions. *Journal of Computational Physics* **295,** 402–419. doi:10.1016/j.jcp.2015.04.004 (Aug. 2015).

81. Ku, S., Chang, C. & Diamond, P. Full-f gyrokinetic particle simulation of centrally heated global ITG turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry. *Nuclear Fusion* **49,** 115021. doi:10.1088/0029–5515/49/11/115021 (Nov. 2009).

82. Ku, S., Hager, R., Chang, C., Kwon, J. & Parker, S. A new hybrid-Lagrangian numerical scheme for gyrokinetic simulation of tokamak edge plasma. *Journal of Computational Physics* **315,** 467–475. doi:10.1016/j.jcp.2016.03.062 (June 2016).

83. Hager, R. *et al.* A fully non-linear multi-species Fokker–Planck–Landau collision operator for simulation of fusion plasma. *Journal of Computational Physics* **315,** 644–660. doi:10.1016/j.jcp.2016.03.064 (June 2016).

84. Hager, R. *et al.* Verification of long wavelength electromagnetic modes with a gyrokinetic-fluid hybrid model in the XGC code. *Physics of Plasmas* **24,** 054508. doi:10.1063/1.4983320 (May 2017).

85. Fasoli, A. *et al.* Computational challenges in magnetic-confinement fusion physics. *Nature Physics* **12,** 411–423. doi:10.1038/nphys3744 (May 2016).

86. Courant, R., Friedrichs, K. & Lewy, H. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen* **100,** 32–74. doi:10.1007/BF01448839 (Dec. 1928).

87. Grandgirard, V. *et al.* A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation. *Journal of Computational Physics* **217,** 395–423. doi:10.1016/j.jcp.2006.01.023 (Sept. 2006).

88. Grandgirard, V. *et al.* A 5D gyrokinetic full-f global semi-Lagrangian code for flux-driven ion turbulence simulations. *Computer Physics Communications* **207,** 35–68. doi:10.1016/j.cpc.2016.05.007 (Oct. 2016).

89. Kwon, J.-M., Yi, D., Piao, X. & Kim, P. Development of semi-Lagrangian gyrokinetic code for full-f turbulence simulation in general tokamak geometry. *Journal of Computational Physics* **283,** 518–540. doi:10.1016/j.jcp.2014.12.017 (Feb. 2015).

90. Latu, G., Mehrenberger, M., Güçlü, Y., Ottaviani, M. & Sonnendrücker, E. Field-Aligned Interpolation for Semi-Lagrangian Gyrokinetic Simulations. *Journal of Scientific Computing* **74,** 1601–1650. doi:10.1007/s10915–017–0509–5 (Mar. 2018).

91. Hölzl, M. *Diffusive Heat Transport across Magnetic Islands and Stochastic Layers in Tokamaks* PhD thesis (Technische Universität München, Garching, Jan. 2010). https://hdl.handle.net/11858/00–001M–0000–0026–EF89–9.

92. Hariri, F., Hill, P., Ottaviani, M. & Sarazin, Y. The flux-coordinate independent approach applied to X-point geometries. *Physics of Plasmas* **21.** doi:10.1063/1.4892405 (Aug. 2014).

93. Hill, P., Hariri, F. & Ottaviani, M. The effect of magnetic islands on Ion Temperature Gradient turbulence driven transport. *Physics of Plasmas* **22,** 042308. doi:10.1063/1.4919031 (Apr. 2015).

94. Stegmeir, A., Coster, D., Maj, O. & Lackner, K. Numerical Methods for 3D Tokamak Simulations Using a Flux-Surface Independent Grid. *Contributions to Plasma Physics* **54,** 549–554. doi:10.1002/ctpp.201410041 (June 2014).

95. Stegmeir, A. K. *GRILLIX: A 3D turbulence code for magnetic fusion devices based on a field line map* PhD thesis (Technische Universität München, Münich, Jan. 2015). https://mediatum.ub.tum.de/1231836.

96. Stegmeir, A., Coster, D., Maj, O., Hallatschek, K. & Lackner, K. The field line map approach for simulations of magnetically confined plasmas. *Computer Physics Communications* **198,** 139–153. doi:10.1016/j.cpc.2015.09.016 (Jan. 2016).

97. Stegmeir, A. *et al.* Advances in the flux-coordinate independent approach. *Computer Physics Communications* **213,** 111–121. doi:10.1016/j.cpc.2016.12.014 (Apr. 2017).

98. Stegmeir, A. *et al.* GRILLIX: a 3D turbulence code based on the flux-coordinate independent approach. *Plasma Physics and Controlled Fusion* **60,** 035005. doi:10.1088/1361–6587/aaa373 (Mar. 2018).

99. Stegmeir, A. *et al.* Global turbulence simulations of the tokamak edge region with GRILLIX. *Physics of Plasmas* **26.** doi:10.1063/1.5089864 (May 2019).

100. Stegmeir, A., Body, T. & Zholobenko, W. Analysis of locally-aligned and non-aligned discretisation schemes for reactor-scale tokamak edge turbulence simulations. *Computer Physics Communications* **290,** 108801. doi:10.1016/j.cpc.2023.108801 (Sept. 2023).

101. Ross, A. *Extension of GRILLIX: Towards a global fluid turbulence code for realistic magnetic geometries* PhD thesis (Technische Universität München, Münich, Oct. 2018). https://mediatum.ub.tum.de/1452529.

102. Ross, A., Stegmeir, A. & Coster, D. Effect of the Boussinesq approximation: Turbulence studies with GRILLIX in slab geometry. *Contributions to Plasma Physics* **58,** 478–483. doi:10.1002/ctpp.201700154 (July 2018).

103. Ross, A. *et al.* On the nature of blob propagation and generation in the large plasma device: Global GRILLIX studies. *Physics of Plasmas* **26.** doi:10.1063/1.5095712 (Oct. 2019).

104. Body, T., Stegmeir, A., Zholobenko, W., Coster, D. & Jenko, F. Treatment of advanced divertor configurations in the flux-coordinate independent turbulence code GRILLIX. *Contributions to Plasma Physics* **60,** e201900139. doi:10.1002/ctpp.201900139 (Dec. 2019).

105. Zholobenko, W. *et al.* Thermal dynamics in the flux-coordinate independent turbulence code GRILLIX. *Contributions to Plasma Physics* **60,** e201900131. doi:10.1002/ctpp.201900131 (Dec. 2019).

106. Zholobenko, W. *et al.* Corrigendum: Thermal dynamics in the flux-coordinate independent turbulence code GRILLIX. *Contributions to Plasma Physics* **60,** e202000056. doi:10.1002/ctpp.202000056 (June 2020).

107. Dudson, B., Umansky, M., Xu, X., Snyder, P. & Wilson, H. BOUT++: A framework for parallel plasma fluid simulations. *Computer Physics Communications* **180,** 1467–1480. doi:10.1016/j.cpc.2009.03.008 (Sept. 2009).

108. Dudson, B. D. *et al.* BOUT++: Recent and current developments. *Journal of Plasma Physics* **81,** 365810104. doi:10.1017/S0022377814000816 (Jan. 2015).

109. Shanahan, B. W., Hill, P. & Dudson, B. D. Towards nonaxisymmetry; initial results using the Flux Coordinate Independent method in BOUT++. *Journal of Physics: Conference Series* **775,** 012012. doi:10.1088/1742-6596/775/1/012012 (Nov. 2016).

110. Dudson, B. D. *et al.* Verification of BOUT++ by the method of manufactured solutions. *Physics of Plasmas* **23,** 062303. doi:10.1063/1.4953429 (June 2016).

111. Shanahan, B. W., Dudson, B. D. & Hill, P. Fluid simulations of plasma filaments in stellarator geometries with BSTING. *Plasma Physics and Controlled Fusion* **61.** doi:10.1088/1361-6587/aaed7d (Feb. 2019).

112. Hill, P., Shanahan, B. W. & Dudson, B. D. Dirichlet boundary conditions for arbitrary-shaped boundaries in stellarator-like magnetic fields for the Flux-Coordinate Independent method. *Computer Physics Communications* **213,** 9–18. doi:10.1016/j.cpc.2016.11.004 (Apr. 2017).

113. Held, M., Wiesenberger, M. & Stegmeir, A. Three discontinuous Galerkin schemes for the anisotropic heat conduction equation on non-aligned grids. *Computer Physics Communications* **199,** 29–39. doi:10.1016/j.cpc.2015.10.009 (Feb. 2016).

114. Held, M. *Full-F gyro-fluid modelling of the tokamak edge and scrape-off layer* PhD thesis (University of Innsbruck, Innsbruck, Austria, Nov. 2016). https://diglib.uibk.ac.at/urn:nbn:at:at-ubi:1-6853.

115. Wiesenberger, M. *et al.* Reproducibility, accuracy and performance of the Feltor code and library on parallel computer architectures. *Computer Physics Communications* **238,** 145–156. doi:10.1016/j.cpc.2018.12.006 (May 2019).

116. Wiesenberger, M. & Held, M. A finite volume flux coordinate independent approach. *Computer Physics Communications,* 108838. doi:10.1016/j.cpc.2023.108838 (July 2023).

117. McMillan, B. F. A partially mesh-free scheme for representing anisotropic spatial variations along field lines. *Computer Physics Communications* **212,** 39–46. doi:10.1016/j.cpc.2016.10.004 (Mar. 2017).

118. Rao, S. S. *The Finite Element Method in Engineering* 6th ed. doi:10.1016/C2016-0-01493-6 (Butterworth-Heinemann, Kidlington, 2018).

119. Zienkiewicz, O. C., Taylor, R. L. & Zhu, J. Z. *The Finite Element Method: its Basis and Fundamentals* 7th ed. doi:10.1016/C2009-0-24909-9 (Butterworth-Heinemann, Kidlington, 2013).

120. Shu, C.-W. & Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics* **77,** 439–471. doi:10.1016/0021-9991(88)90177-5 (Aug. 1988).

121. Shu, C.-W. Total-Variation-Diminishing Time Discretizations. *SIAM Journal on Scientific and Statistical Computing* **9,** 1073–1084. doi:10.1137/0909073 (Nov. 1988).

122. Shu, C.-W. & Osher, S. in *Upwind and High-Resolution Schemes* (eds Hussaini, M. Y., van Leer, B. & Van Rosendale, J.) 328–374 (Springer, Berlin, Heidelberg, 1989). doi:10.1007/978-3-642-60543-7_14.

123. Lallemand, M.-H. *Dissipative properties of Runge-Kutta schemes with upwind spatial approximation for the Euler equations* tech. rep. (INRIA, 1990), RR–1173. https://hal.inria.fr/inria-00075385.

124. Crank, J. & Nicolson, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society* **43,** 50–67. doi:10.1017/S0305004100023197 (Jan. 1947).

125. Chen, J.-S., Hillman, M. C. & Chi, S.-W. Meshfree Methods: Progress Made after 20 Years. *Journal of Engineering Mechanics* **143,** 04017001. doi:10.1061/(ASCE)EM.1943-7889.0001176 (Apr. 2017).

126. Lancaster, P. & Salkauskas, K. Surfaces Generated by Moving Least Squares Methods. *Mathematics of Computation* **37,** 141. doi:10.2307/2007507 (July 1981).

127. Nayroles, B., Touzot, G. & Villon, P. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics* **10,** 307–318. doi:10.1007/BF00364252 (Sept. 1992).

128. Nguyen, V. P., Rabczuk, T., Bordas, S. & Duflot, M. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation* **79,** 763–813. doi:10.1016/j.matcom.2008.01.003 (Dec. 2008).

129. Huerta, A., Belytschko, T., Fernández-Méndez, S. & Rabczuk, T. in *Encyclopedia of Computational Mechanics* (eds Stein, E., de Borst, R. & Hughes, T. J.) 1st ed., 279–309 (John Wiley & Sons, 2004). doi:10.1002/0470091355.ecm005.

130. Belytschko, T., Lu, Y. Y. & Gu, L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* **37,** 229–256. doi:10.1002/nme.1620370205 (Jan. 1994).

131. Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. & Krysl, P. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* **139,** 3–47. doi:10.1016/S0045-7825(96)01078-X (Dec. 1996).

132. Hu, H.-Y., Lai, C.-K. & Chen, J.-S. A study on convergence and complexity of reproducing kernel collocation method. *Interaction and multiscale mechanics* **2,** 295–319. doi:10.12989/imm.2009.2.3.295 (Sept. 2009).

133. Grad, H. & Rubin, H. *Hydromagnetic Equilibria and Force-Free Fields* in *Proceedings of the 2nd United Nations International Conference on the Peaceful Uses of Atomic Energy, Volume 31* **31** (United Nations Publications, Geneva, Sept. 1958), 190–197. https://digitallibrary.un.org/record/3808553.

134. Shafranov, V. D. On Magnetohydrodynamical Equilibrium Configurations. *Soviet physics, JETP* **6,** 545–554. http://jetp.ras.ru/cgi-bin/e/index/e/6/3/p545?a=list (Mar. 1958).

135. Lüst, R. & Schlüter, A. Axialsymmetrische magnetohydrodynamische Gleichgewichtskonfigurationen. *Zeitschrift für Naturforschung A* **12,** 850–854. doi:10.1515/zna-1957-1014 (Oct. 1957).

136. Lütjens, H., Bondeson, A. & Roy, A. Axisymmetric MHD equilibrium solver with bicubic Hermite elements. *Computer Physics Communications* **69,** 287–298. doi:10.1016/0010-4655(92)90167-W (Mar. 1992).

137. Lütjens, H., Bondeson, A. & Sauter, O. The CHEASE code for toroidal MHD equilibria. *Computer Physics Communications* **97,** 219–260. doi:10.1016/0010-4655(96)00046-X (Sept. 1996).

138. Hughes, T. J. R., Engel, G., Mazzei, L. & Larson, M. G. The Continuous Galerkin Method Is Locally Conservative. *Journal of Computational Physics* **163,** 467–488. doi:10.1006/jcph.2000.6577 (Sept. 2000).

139. Beissel, S. & Belytschko, T. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* **139,** 49–74. doi:10.1016/S0045-7825(96)01079-1 (Dec. 1996).

140. Dolbow, J. & Belytschko, T. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics* **23,** 219–230. doi:10.1007/s004660050403 (Apr. 1999).

141. Atluri, S. N. & Zhu, T. New concepts in meshless methods. *International Journal for Numerical Methods in Engineering* **47,** 537–556. doi:10.1002/(SICI)1097-0207(20000110/30)47:1/3<537::AID-NME783>3.0.CO;2-E (Jan. 2000).

142. De, S. & Bathe, K. J. The method of finite spheres. *Computational Mechanics* **25,** 329–345. doi:10.1007/s004660050481 (Apr. 2000).

143. De, S. & Bathe, K.-J. The method of finite spheres with improved numerical integration. *Computers & Structures* **79,** 2183–2196. doi:10.1016/S0045-7949(01)00124-9 (Sept. 2001).

144. Chen, J.-S., Wu, C.-T., Yoon, S. & You, Y. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* **50,** 435–466. doi:10.1002/1097-0207(20010120)50:2<435::AID-NME32>3.0.CO;2-A (Jan. 2001).

145. Chen, J.-S., Yoon, S. & Wu, C.-T. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* **53,** 2587–2615. doi:10.1002/nme.338 (Apr. 2002).

146. Carpinteri, A., Ferro, G. & Ventura, G. The partition of unity quadrature in meshless methods. *International Journal for Numerical Methods in Engineering* **54,** 987–1006. doi:10.1002/nme.455 (July 2002).

147. Babuška, I., Banerjee, U., Osborn, J. E. & Li, Q. Quadrature for meshless methods. *International Journal for Numerical Methods in Engineering* **76,** 1434–1470. doi:10.1002/nme.2367 (Nov. 2008).

148. Chen, J.-S., Chi, S.-W. & Hu, H.-Y. Recent developments in stabilized Galerkin and collocation meshfree methods. *Computer Assisted Mechanics and Engineering Sciences* **18,** 3–21. ISSN: 1232308X. https://cames.ippt.pan.pl/index.php/cames/article/view/117 (2011).

149. Chen, J.-S., Hillman, M. C. & Rüter, M. An arbitrary order variationally consistent integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering* **95,** 387–418. doi:10.1002/nme.4512 (Aug. 2013).

150. De Arantes e Oliveira, E. R. Theoretical foundations of the finite element method. *International Journal of Solids and Structures* **4,** 929–952. doi:10.1016/0020-7683(68)90014-0 (Oct. 1968).

151. Babuška, I. Error-bounds for finite element method. *Numerische Mathematik* **16,** 322–333. doi:10.1007/BF02165003 (Jan. 1971).

152. Strang, G. & Fix, G. *An Analysis of the Finite Element Method* 1st ed. ISBN: 0-13-032946-0. https://archive.org/details/analysisoffinite0000stra (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973).

153. Hughes, T. J. R. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis* 1st ed. ISBN: 0-13-317025-X. https://store.doverpublications.com/0486411818.html (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1987).

154. Li, S. *Moving Least Square Reproducing Kernel Method* PhD thesis (Northwestern University, Evanston, Illinois, June 1997). ISBN: 978-0-591-40724-2. https://www.proquest.com/docview/304360505.

155. Liu, W.-K., Li, S. & Belytschko, T. Moving least-square reproducing kernel methods (I) Methodology and convergence. *Computer Methods in Applied Mechanics and Engineering* **143,** 113–154. doi:10.1016/S0045-7825(96)01132-2 (Apr. 1997).

156. Li, S. & Liu, W. K. Moving least-square reproducing kernel method Part II: Fourier analysis. *Computer Methods in Applied Mechanics and Engineering* **139,** 159–193. doi:10.1016/S0045-7825(96)01082-1 (Dec. 1996).

157. Duarte, C. A. & Oden, J. T. H-p clouds—an h-p meshless method. *Numerical Methods for Partial Differential Equations* **12,** 673–705. doi:10.1002/(SICI)1098-2426(199611)12:6<673::AID-NUM3>3.0.CO;2-P (Nov. 1996).

158. Duarte, C. & Oden, J. An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* **139,** 237–262. doi:10.1016/S0045-7825(96)01085-7 (Dec. 1996).

159. Han, W. & Meng, X. Error analysis of the reproducing kernel particle method. *Computer Methods in Applied Mechanics and Engineering* **190,** 6157–6181. doi:10.1016/S0045-7825(01)00214-6 (Sept. 2001).

160. Chen, J.-S., Han, W., You, Y. & Meng, X. A reproducing kernel method with nodal interpolation property. *International Journal for Numerical Methods in Engineering* **56,** 935–960. doi:10.1002/nme.592 (Feb. 2003).

161. Huerta, A. & Fernández-Méndez, S. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering* **48,** 1615–1636. doi:10.1002/1097-0207(20000820)48:11<1615::AID-NME883>3.0.CO;2-S (Aug. 2000).

162. Li, S. & Liu, W. K. Meshfree and particle methods and their applications. *Applied Mechanics Reviews* **55,** 1–34. doi:10.1115/1.1431547 (Jan. 2002).

163. Wagner, G. J. & Liu, W. K. Application of essential boundary conditions in mesh-free methods: A corrected collocation method. *International Journal for Numerical Methods in Engineering* **47,** 1367–1379. doi:10.1002/(SICI)1097-0207(20000320)47:8<1367::AID-NME822>3.0.CO;2-Y (Mar. 2000).

164. Belytschko, T., Organ, D. & Krongauz, Y. A coupled finite element-element-free Galerkin method. *Computational Mechanics* **17,** 186–195. doi:10.1007/BF00364080 (May 1995).

165. Wagner, G. J. & Liu, W. K. Hierarchical enrichment for bridging scales and mesh-free boundary conditions. *International Journal for Numerical Methods in Engineering* **50,** 507–524. doi:10.1002/1097-0207(20010130)50:3<507::AID-NME33>3.0.CO;2-B (Jan. 2001).

166. Zhu, T. & Atluri, S. N. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. *Computational Mechanics* **21,** 211–222. doi:10.1007/s004660050296 (Apr. 1998).

167. Atluri, S. N. & Zhu, T. A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* **22,** 117–127. doi:10.1007/s004660050346 (Aug. 1998).

168. Nitsche, J. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* **36,** 9–15. doi:10.1007/BF02995904 (July 1971).

169. Zhou, P.-b. *Numerical Analysis of Electromagnetic Fields* doi:10.1007/978-3-642-50319-1 (Springer, Berlin, Heidelberg, 1993).

170. Heinrich, J. C., Huyakorn, P. S., Zienkiewicz, O. C. & Mitchell, A. R. An 'upwind' finite element scheme for two-dimensional convective transport equation. *International Journal for Numerical Methods in Engineering* **11,** 131–143. doi:10.1002/nme.1620110113 (1977).

171. Hughes, T. J. R. A simple scheme for developing 'upwind' finite elements. *International Journal for Numerical Methods in Engineering* **12,** 1359–1365. doi:10.1002/nme.1620120904 (1978).

172. Payre, G., De Broissia, M. & Bazinet, J. An 'upwind' finite element method via numerical integration. *International Journal for Numerical Methods in Engineering* **18,** 381–396. doi:10.1002/nme.1620180306 (Mar. 1982).

173. Brooks, A. N. & Hughes, T. J. R. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* **32,** 199–259. doi:10.1016/0045-7825(82)90071-8 (Sept. 1982).

174. Rice, J. & Schnipke, R. A monotone streamline upwind finite element method for convection-dominated flows. *Computer Methods in Applied Mechanics and Engineering* **48,** 313–327. doi:10.1016/S0045-7825(85)80005-0 (Apr. 1985).

175. Bank, R. E., Bürgler, J. F., Fichtner, W. & Smith, R. K. Some upwinding techniques for finite element approximations of convection-diffusion equations. *Numerische Mathematik* **58,** 185–202. doi:10.1007/BF01385618 (Dec. 1990).