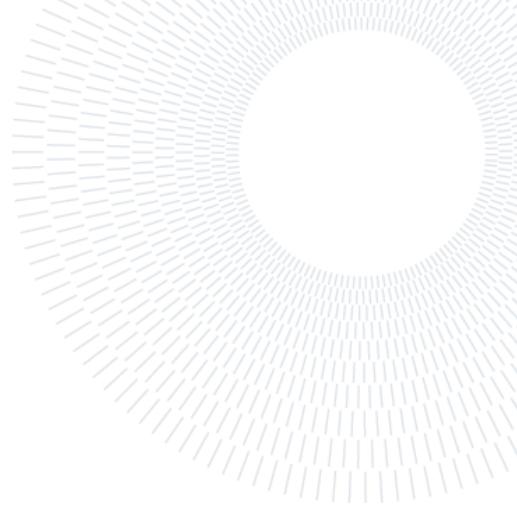




**POLITECNICO
MILANO 1863**

**SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE**



Algorithms and optimizations for global non-linear hybrid fluid-kinetic finite element stellarator simulations

Tesi di Laurea Magistrale in
High Performance Computing Engineering - Ingegneria del Supercalcolo

Luca Venerando Greco, 245265

Abstract: Predictive modeling of stellarator plasmas is crucial for advancing nuclear fusion energy, yet it faces unique computational difficulties. One of the main challenges is accurately simulating the dynamics of specific particle species that are not well captured by fluid models, which necessitates the use of hybrid fluid-kinetic models. The non-axisymmetric geometry of stellarators fundamentally couples the toroidal Fourier modes, in contrast to what happens in tokamaks, requiring different numerical and computational treatment.

This work presents a novel, globally coupled projection scheme inside the JOREK finite element framework. The approach ensures a self-consistent and physically accurate transfer of kinetic markers to the fluid grid, effectively handling the complex 3D mesh by constructing and solving a unified linear system that encompasses all toroidal harmonics simultaneously. To manage the computational complexity of this coupling, the construction of the system's matrix is significantly accelerated using the Fast Fourier Transform (FFT). The efficient localization of millions of particles is made possible by implementing a 3D R-Tree spatial index, which supports this projection and ensures computational tractability at scale.

On realistic Wendelstein 7-X stellarator geometries, the fidelity of the framework is rigorously shown. In sharp contrast to the uncoupled approaches' poor performance, quantitative convergence tests verify that the coupled scheme attains the theoretically anticipated spectral convergence.

This study offers a crucial capability for the predictive analysis and optimization of next-generation stellarator designs by developing a validated, high-fidelity computational tool.

Key-words: stellarator Simulation, Hybrid Fluid-Kinetic Model, Finite Element Method (FEM), Particle-in-Cell (PiC), JOREK, High Performance Computing (HPC)

Contents

1	Introduction	3
2	Background	4
2.1	Motivation	4
2.1.1	Simulation types	4
2.1.2	Reactors types	4
2.2	Governing equations and Numerical Approximations for the fluid	5
2.2.1	The MHD stellarator model	5
2.2.2	Spatial Discretization	6
2.2.3	Bézier Finite Elements	6
2.2.4	Tesselation of the domain	7
2.2.5	Imposing G^1 on the field	8
2.2.6	Field Interpolation	9
2.2.7	Bounding Box of a Bicubic Bézier Element	10
2.2.8	Toroidal Fourier expansion	10
2.2.9	The Finite Element Method	11
2.3	Governing equations and Numerical Approximations for the kinetics	11
2.3.1	Newton-Lorentz equation	11
2.3.2	The Boris Pusher	11
2.3.3	Projecting particles on the Finite Element grid	12
3	Algorithms and verification for PiC methods in the JOREK stellarator model	16
3.1	Preliminary tests	16
3.1.1	Particle Tracing	16
3.1.2	Different Trajectories	17
3.1.3	Particle Projection	18
3.2	Implementation of the coupled harmonics	19
3.2.1	Changes in the matrix system	20
3.3	Coupled projection	21
3.3.1	Gaussian distribution on the W7-A	21
3.3.2	Gaussian distribution on the W7-X	22
3.4	Convergence Tests	23
3.5	FFT acceleration of the matrix construction	23
3.6	The RTree	25
4	Conclusion	27

1. Introduction

Nowadays, nuclear fusion is one of the most important scientific and engineering problems of our time due to the quest for clean, sustainable energy. Fusion reactors promise an almost infinite energy source with major safety and environmental advantages over traditional fission and fossil fuel technologies by exploiting a similar process that powers stars [2]. The confinement of a high-temperature plasma in a magnetic field is the most promising method for accomplishing controlled fusion on Earth. However, high-fidelity computational simulations are required due to the enormous expense and complexity of building and running experimental reactors [40]. These simulations are essential resources for developing new approaches, deciphering experimental data, and investigating plasma phenomena that are challenging or impossible to measure directly.

Fluid and kinetic models are the two main categories into which plasma simulation paradigms fall. Large-scale instabilities and general plasma behavior are effectively captured by fluid models, which are governed by Magneto-HydroDynamics (MHD) equations and treat the plasma as a continuum [24]. Kinetic models, on the other hand, follow the paths of individual particles and offer a more accurate, albeit more computationally expensive, representation of the plasma [25]. The development of hybrid fluid-kinetic approaches has been prompted by the prohibitive computational cost of fully kinetic simulations for reactor-scale plasmas over MHD-relevant timescales. These models create a self-consistent framework by combining the accuracy of a kinetic treatment for certain, important particle populations with the computational efficiency of a fluid description for the bulk plasma [35].

Numerous specialized numerical codes have been created to handle the various physical regimes found in magnetically confined plasmas. Usually, each of these tools is tailored for a particular set of scales, equations, or configurations. In particular, for hybrid MHD-kinetic simulations many codes have been developed, for instance: JOREK [19], NIMROD [39], M3D-C1-K [13, 28], MEGA [41] and XHMGC [43] are some of the leading codes in the field, designed to couple a fluid description of the bulk plasma with a kinetic treatment for specific particle species. Among them, specific interest has sparked in the simulation of stellarator reactors, leading to the development of specific extensions to account for the three-dimensionality of the plasma [7, 33, 47], underscoring the interest and need of the scientific community for this kind of reactor.

The tokamak and the stellarator are the two main device concepts that have emerged in the field of magnetic confinement fusion. Both configurations are actively researched as promising concepts for a fusion reactor with different strengths and weaknesses. The tokamak's intrinsic toroidal symmetry makes its physical model and numerical implementation easier, but the stellarator's non-axisymmetric, three-dimensional magnetic field poses a much greater computational challenge. The need for sophisticated simulation tools that can faithfully capture the unique physics of stellarator designs is growing in importance, triggered in particular by the successful operation of the W7-X. As a result, although there are established hybrid simulation frameworks for tokamaks in codes like JOREK [19], there is still a vital and ongoing research need to extend them to stellarator geometries. The breakdown of toroidal axisymmetry is the main obstacle to applying the available hybrid models to stellarators. This symmetry permits a spectral decomposition of the mesh geometry in the toroidal direction, in which Fourier harmonics are decoupled, in tokamak simulations. The numerical problem is greatly simplified by this decoupling since each harmonic's kinetic particle data can be independently projected onto the fluid grid. These harmonics are intrinsically coupled in a stellarator due to the three-dimensional nature of the geometry. The basis functions utilized for the finite element discretization are functions of all three spatial dimensions since the physical form of the poloidal cross-section varies with the toroidal angle. The uncoupled projection scheme would produce nonphysical results if applied to stellarators and would fail to accurately represent the physical plasma.

By creating, putting into practice, and validating the algorithms necessary for a global non-linear hybrid fluid-kinetic simulation capability within the JOREK stellarator model, this thesis directly tackles this challenge. The design of a novel particle-in-cell projection scheme that completely takes into account the coupling between toroidal harmonics that results from the non-axisymmetric mesh geometry is the main contribution. This approach allows for a precise and self-consistent transfer of information from the kinetic particles to the fluid grid by accurately formulating and solving the global linear system for all harmonics at the same time. These projections are needed for diagnosing the particle dynamics and for a self-consistent coupling between fluid and kinetics. To overcome the significant computational cost of this approach, two key optimizations were implemented. Firstly, the assembly of the global system matrix is accelerated using the Fast Fourier Transform (FFT). Secondly, a three-dimensional R-Tree spatial index is introduced to handle the expensive task of localizing a high number of particles within the complex 3D grid.

The work is structured as follows: the required theoretical background is given in section 2, which also describes the G^1 Bézier finite element framework used in JOREK for the fluid model and the reduced-MHD equations. The kinetic particle-in-cell methodology is then presented, along with an original derivation of the mathematical formulation for the projection scheme and the Boris pusher for particle tracing. The main contributions of this study are described in detail in section 3. The new coupled-harmonic projection algorithm is implemented

after first validating the basic particle tracing in stellarator configurations. Quantitative convergence tests and qualitative projections on realistic stellarator geometries are used to illustrate the precision and effectiveness of this new scheme. Following this validation, the chapter analyzes the computational performance of the key optimizations, presenting a speedup analysis of the FFT-accelerated matrix construction against the direct integration method. We conclude by exploring how the 3D R-Tree for effective particle localization is implemented and examining its performance. This methodical approach paves the way for next-generation, high-fidelity stellarator simulations by validating the new framework as a reliable and accurate tool.

2. Background

The theoretical foundation for comprehending the hybrid simulation approach used in this thesis is laid out in this section. We first describe the motivation behind this study, emphasizing the significance of using the stellarator and tokamak models for high-fidelity plasma simulations. Next, we describe the hybrid model's two main pillars. The reduced MHD equations and the numerical framework for solving them in the JOREK code, which combines G^1 continuous Bézier finite elements for the poloidal plane with a toroidal Fourier expansion, are introduced first in the fluid component. Second, we present the kinetic model, which explains how the Newton-Lorentz equations govern the motion of individual particles and how the Boris pusher algorithm is used to integrate them.

2.1. Motivation

Fusion reactors have been a primary focus of clean energy research for the past half-century, offering the promise of clean, virtually limitless energy without the drawbacks associated with fission energy. The significant engineering challenges and extensive resources required for experimentation have slowed progress, necessitating computer-aided simulations to test and validate hypotheses.

2.1.1 Simulation types

A basic taxonomy of plasma physics simulations was proposed in early literature [4], establishing a clear distinction between kinetic and fluid simulations. Kinetic simulations describe plasma at the particle level, while fluid simulations treat plasma as a single or multiple fluids governed by MHD equations. The fundamental trade-off between these approaches balances computational requirements against accuracy. While fully kinetic simulations can precisely capture particle interactions within plasma, the computational complexity grows exponentially with particle count. On the other hand, fluid-based simulations solve Maxwell's equations on finite element grids, with complexity directly related to grid resolution and the sophistication of the MHD equations employed. In recent years, despite the increase in HPC computational power[45], the inherent approximations in fluid simulations have limited achievable accuracy, driving the development of hybrid simulation approaches. In these hybrid simulations, the fluid solver is combined with a kinetic simulation of particular particle species that cannot be described well in the fluid picture, like supra-thermal ions [5], neutral particles or impurities [27], or runaway electrons [3]. The time evolution of the kinetic simulation provides accurate information about the time evolution of plasma constituents under kinetic assumptions, which is then incorporated into the next fluid time step by coupling terms. A schematic example of the underlying physics is shown in Fig. 1. This creates a continuous feedback loop between the two simulation approaches, allowing them to self-consistently evolve the complete system.

2.1.2 Reactors types

Fusion reactions exploiting magnetic confinement of the plasma are carried out at the moment, mainly, using two different types of reactors, stellarator and tokamak. The latter has seen widespread use in the scientific community for its symmetric shape and ease of design; on the other hand, designs and simulations of stellarator reactors have been limited by the computational power available [17]. Nevertheless, the stellarator reactor offers a number of advantages such as intrinsically steady-state operation, fewer MHD instabilities, and nearly disruption-free operation, i.e., no severe loss of the overall plasma confinement quality [46].

Due to the popularity of tokamak designs, significant progress has been made in developing physically accurate simulations that combine finite element method (FEM) resolution of (reduced) MHD models with kinetic solvers, creating hybrid solutions, e.g., in the JOREK [19] codebase that is of interest to this thesis. While substantial work has been done to develop MHD models for stellarator reactors [31, 32], current simulations lack the crucial kinetic solver component. Integrating such a solver with fluid models would enable more accurate stellarator simulations, potentially accelerating research for the next generation of these less widely studied reactor designs.

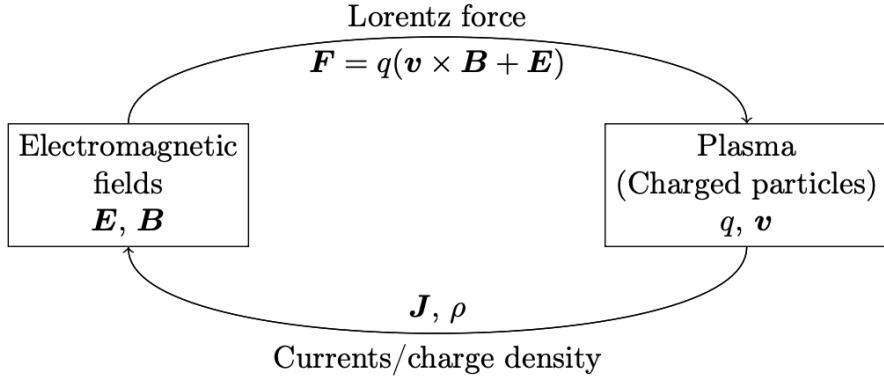


Figure 1: Feedback loop between electromagnetic fields and plasma in kinetic-fluid simulations. The Lorentz force acts on charged particles, influencing their motion and altering currents and charge densities, which in turn affect the electromagnetic fields [23].

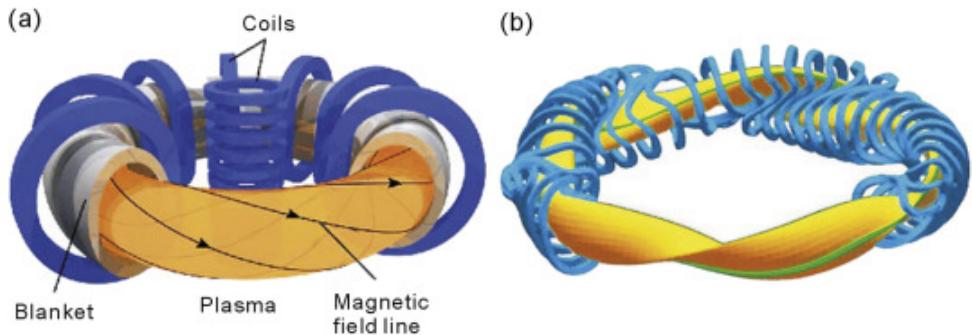


Figure 2: A computer graphic representation of the two different types of reactors [46]. On the left, the tokamak; on the right, the stellarator.

2.2. Governing equations and Numerical Approximations for the fluid

The code is logically separated into two different parts: the fluid part and the kinetic part.

The fluid part provides a starting point for the kinetic module by solving a reduced MHD model of the plasma using the Finite Element Method (FEM).

2.2.1 The MHD stellarator model

Firstly conceptualized in [31] and further implemented in [32], the implemented model aims at handling the MHD equations by not considering the fast magnetosonic waves, since they account for the fastest time scales of the system and are often not relevant to the problems considered.

The set of equations derived from the viscoresistive MHD equations is provided with time evolution equations for the density, velocity, magnetic field, and internal energy, as well as additional definitions and constraints, as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = P,$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho (\mathbf{v} \cdot \nabla) \mathbf{v} + \mathbf{v} P = \mathbf{j} \times \mathbf{B} - \nabla p + \mu \Delta \mathbf{v},$$

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

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot \left[\left(\frac{\rho v^2}{2} + \frac{\gamma p}{\gamma - 1} \right) \mathbf{v} + \frac{p}{\gamma - 1} \frac{D_\perp}{\rho} \nabla_\perp \rho + \frac{\mathbf{E} \times \mathbf{B}}{\mu_0} - \kappa_\perp \nabla_\perp T - \kappa_\parallel \nabla_\parallel T \right] = S_e - \frac{v^2 P}{2}, \quad (1)$$

$$\mathcal{E} = \frac{\rho v^2}{2} + \frac{p}{\gamma - 1} + \frac{B^2}{2\mu_0},$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j},$$

$$\nabla \cdot \mathbf{B} = 0,$$

$$\mathbf{E} = -\mathbf{v} \times \mathbf{B} + \eta \mathbf{j},$$

$$P = \nabla \cdot (D_\perp \nabla_\perp \rho) + S_\rho.$$

These equations, modified with suitable ansatzes, will be solved on a 2D G^1 continuous finite element grid combined with a toroidal Fourier decomposition to provide the electromagnetic field in which the kinetic particles will move through. For a definition of all symbols, refer to [31].

2.2.2 Spatial Discretization

The JOREK code uses G^1 continuous isoparametric Bézier finite elements [19] to discretize the poloidal cross-section of the plasma and a Fourier expansion on the toroidal direction. This allows for achieving the same physical accuracy with a low number of DoFs compared to other types of elements [10].

2.2.3 Bézier Finite Elements

For the finite element method taken into account, Bernstein polynomials are used along each of the two element-local coordinates s and t :

$$B_i^3(s) = \frac{3!}{i! (3-i)!} s^i (1-s)^{3-i} \quad (2)$$

These exhibit some interesting properties such as:

- (B_i^n) is a basis of \mathbb{P}^n ,
- $0 \leq B_i^n(s) \leq 1, \forall s \in [0, 1]$
- (B_i^n) is a partition of unity $\implies \sum_{i=0}^n B_i^n(s) = 1, \forall s \in [0, 1]$

Each element uses two different third-degree Bernstein polynomials defined by:

$$B_{i,j}^3(s, t) = B_i^3(s) B_j^3(t) \quad i, j = 0 \dots 3, \quad (3)$$

Where $0 \leq s, t \leq 1$ denote the local coordinates, which take values 0 and 1 at the four element vertices. These 16 basis functions $B_{i,j}^3(s, t)$ are then used to build a 2D Bézier patch through parametrization of 16 control points $P_{i,j}$, each of them belonging to an appropriate vector space, in our case $P_{i,j} \in \mathbb{R}$.

The position of each point $P(s, t)$ on the patch is defined as:

$$P(s, t) = \sum_{i=0}^3 \sum_{j=0}^3 P_{i,j} B_{i,j}^3(s, t) \quad (4)$$

In this equation, the Bernstein polynomial serves as the weights for the control points. Since the used polynomials form a partition of unity, some properties arise, in particular:

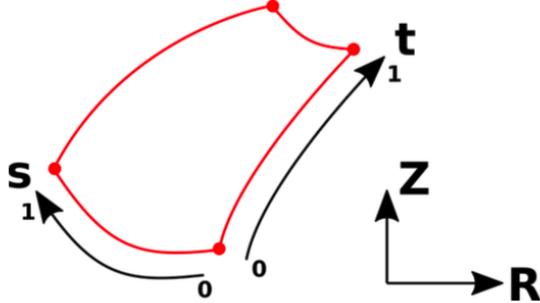


Figure 3: A single 2D isoparametric Bézier element in an axisymmetric R-Z space. The geometry is parameterized by a mapping from the local s-t parent domain using Bernstein polynomial basis functions, which are also used to interpolate field variables like temperature.

A point $P(s, t)$ on a parametric surface is a convex combination of the set of control points $\{P_{i,j}\}$ if it can be written as:

$$P(s, t) = \sum_{i=0}^3 \sum_{j=0}^3 \alpha_{i,j}(s, t) P_{i,j} \quad (5)$$

Where the blending functions $\alpha_{i,j}(s, t)$ must satisfy the following properties for all (s, t) in their domain (e.g., $s, t \in [0, 1]$):

- **Non-negativity:** $\alpha_{i,j}(s, t) \geq 0 \quad \forall i, j$
- **Partition of Unity:** $\sum_{i=0}^3 \sum_{j=0}^3 \alpha_{i,j}(s, t) = 1$

Since the polynomials satisfy both conditions, it is guaranteed that all the points are contained in the convex hull of their control points.

From (4) we can also derive the local derivatives, for instance on $P_{0,0}$:

$$\left. \frac{\partial P}{\partial s} \right|_{(s,t)=(0,0)} = 3(P_{1,0} - P_{0,0}); \quad (6)$$

$$\left. \frac{\partial P}{\partial t} \right|_{(s,t)=(0,0)} = 3(P_{0,1} - P_{0,0}); \quad (7)$$

$$\left. \frac{\partial^2 P}{\partial s \partial t} \right|_{(s,t)=(0,0)} = 9(P_{1,1} + P_{0,0} - P_{0,1} - P_{1,0}). \quad (8)$$

Both variables and space are discretized using these 16 basis functions, giving the iso-parametric characterization to this method.

2.2.4 Tesselation of the domain

In the finite element method, we split the entire domain into different smaller elements, in our case Bézier patches (e.g., Fig. 3). In doing so, we need to take care to satisfy specific conditions imposed by the application's domain-specific requirements.

The first important constraint to satisfy is Geometric continuity between elements (G^0), which simply ensures that there are no gaps between two adjacent elements. Without loss of generality, we consider two adjacent surfaces S and S' which share a common boundary, assuming such boundary lies at the edge $s = 1$ for patch S , and $s = 0$ for patch S' .

The boundary of the first patch is defined by $\{P_{3,0}, P_{3,1}, P_{3,2}, P_{3,3}\}$, while for the second one $\{P'_{0,0}, P'_{0,1}, P'_{0,2}, P'_{0,3}\}$. G^0 is therefore easily achieved by simply enforcing:

$$P_{3,j} = P'_{0,j} \quad \forall j \in 0, 1, 2, 3 \quad (9)$$

Having a continuous surface across the domain is almost trivial, and given for most elements, it becomes more interesting to consider the G^1 continuity between the elements. This must be enforced to avoid sharp, non-physical creases between elements. To ensure such smooth surfaces between elements, the derivatives need to be constrained in some way. G^1 continuity requires satisfaction of the so-called "tangent plane condition" [42]

The tangent plane at any point of the boundary is spanned by the four partial derivatives. For instance, considering the previous shared edge, assuming G^0 continuity, we know that:

$$\frac{\partial P}{\partial t} \Big|_{s=0} = \frac{\partial P'}{\partial t'} \Big|_{s'=0} \quad (10)$$

Leaving us with the need to constrain only the other derivatives:

$$k_1(t) \left(\frac{\partial P}{\partial s} \right) + k_2(t) \left(\frac{\partial P}{\partial t} \right) + k_3(t) \left(\frac{\partial P'}{\partial s'} \right) = 0. \quad (11)$$

In practice, this requirement is equivalent to imposing the collinear alignment of the control points across the boundary [29]:

$$P_{3,j} - P'_{1,j} = k(P_{2,j} - P_{3,j}) \quad \forall j \in 0, 1, 2, 3 \quad (12)$$

In (12) k is a (positive) scaling factor between the magnitudes of the tangent vectors across the boundary. A special case is found for $k = 1$, where the formulation leads to C^1 continuity and the Hermite polynomial. This parameter allows for flexibility in the mesh refinement.

2.2.5 Imposing G^1 on the field

Now, we focus on how the geometrical tools derived so far can be used to interpolate a field. To better understand which information is needed on each node, it is better to shift the perspective from the element to a single vertex.

Following the derivation in (8), we know that the values and derivative in one of the corners only depends on the three closest control points. Given this, we define a local 3×3 grid of control points, denoted $P_{i,j}$ where $(i, j) \in \{-1, 0, 1\}^2$, centered on the shared vertex $P_{0,0}$. This 9-point grid is a subset of the control points from the four adjoining 16-point patches. For example, for one patch, $P_{0,0}$ corresponds to its corner control point, $P_{1,0}$ to its first tangential control point, and $P_{1,1}$ to its first twist control point.

The isoparametric property means that each control point is a vector $P_{i,j} = (R_{i,j}, Z_{i,j}, w_{i,j})$, where (R, Z) are the spatial coordinates and w is the field. We assume the mesh geometry $(R_{i,j}, Z_{i,j})$ is given. The unknowns are the nine scalar values $w_{i,j}$, which constitute the initial DoFs at the node $P_{0,0}$.

With this vertex-centered notation, the G^1 continuity condition (12) can be expressed more generally for the two principal directions s and t :

$$P_{1,j} - P_{0,j} = k_s(P_{0,j} - P_{-1,j}) \quad \forall j \in -1, 0, 1 \quad (13)$$

$$P_{i,1} - P_{i,0} = k_t(P_{i,0} - P_{i,-1}) \quad \forall i \in -1, 0, 1 \quad (14)$$

where k_s and k_t are scalar constants determined by the mesh geometry.

Since our geometry is fixed, the equations can be used directly on $w_{i,j}$ to derive the necessary DoFs. Consider the equations for the s -direction:

$$w_{1,j} - w_{0,j} = k_s(w_{0,j} - w_{-1,j}) \quad \forall j \in -1, 0, 1 \quad (15)$$

As (15) shows, the DoFs are not independent; they must lie on a line in the field space.

We can now define the directed distances along the control polygon legs, from the central node as:

$$d_{u,j} = \text{sign}(j) \cdot \|M_{j,0} - M_{0,0}\| \quad \text{for } j = \pm 1 \quad (16)$$

$$d_{v,i} = \text{sign}(i) \cdot \|M_{0,i} - M_{0,0}\| \quad \text{for } i = \pm 1 \quad (17)$$

where $M_{i,j}$ are the projection on (O, x, y) :

$$M_{i,j} = (R_{i,j}, Z_{i,j}) \quad (18)$$

The values $d_{u,i}$ and $d_{v,j}$ are scalar properties derived from the mesh geometry.

The first two DoFs, after the field value $w_{0,0}$ itself, are $a_{0,0}$ and $b_{0,0}$, defined as:

$$a_{0,0} \equiv \frac{w_{1,0} - w_{0,0}}{d_{u,1}} = \frac{w_{-1,0} - w_{0,0}}{d_{u,-1}} \quad (19)$$

$$b_{0,0} \equiv \frac{w_{0,1} - w_{0,0}}{d_{v,1}} = \frac{w_{0,-1} - w_{0,0}}{d_{v,-1}}. \quad (20)$$

Imposing the equivalence on the DoFs guarantees the required smoothness of the solution.

This handles the derivatives in s and t , reducing the four DoFs related to the direct neighbors of the vertex to only two, but leaves us with the open issue of handling the mixed derivatives at the twist/corner points. For each of the four corners is possible to define:

$$\mathbf{m}_{i,j} \equiv P_{i,j} + P_{0,0} - P_{i,0} - P_{0,j}. \quad (21)$$

This serves the purpose of approximating a discrete mixed derivative $\frac{\partial^2 P}{\partial s \partial t}$; geometrically, this follows the parallelogram rule.

It is possible to prove that:

$$\mathbf{m}_{i,j} = \frac{d_{u,i}}{d_{u,k}} \mathbf{m}_{k,j}, \quad \mathbf{m}_{k,j} = \frac{d_{v,j}}{d_{v,l}} \mathbf{m}_{k,l}, \quad \forall i, j, k, l = \pm 1, \quad (22)$$

Yielding the existence at the corner of a vector independent from the local twist points:

$$\frac{1}{d_{u,i} d_{v,j}} \mathbf{m}_{i,j} = \frac{1}{d_{u,k} d_{v,l}} \mathbf{m}_{k,l} = \vec{\Gamma}, \quad \forall i, j, k, l = \pm 1. \quad (23)$$

Therefore, calling $\gamma_{0,0}$ the third coordinate of $\vec{\Gamma}$, we obtain:

$$\psi_{i,j} + \psi_{0,0} - \psi_{i,0} - \psi_{0,j} = \gamma_{0,0} d_{u,i} d_{v,j} \quad \forall i, j = \pm 1. \quad (24)$$

Consequently, only four degrees of freedom ($\psi_{0,0}, \alpha_{0,0}, \beta_{0,0}, \gamma_{0,0}$), instead of nine, are necessary for each fluid quantity to determine all the control points associated with vertex $P_{0,0}$:

$$\begin{cases} \psi_{i,0} = \psi_{0,0} + \alpha_{0,0} d_{u,i} \\ \psi_{0,j} = \psi_{0,0} + \beta_{0,0} d_{v,j} \\ \psi_{i,j} = \psi_{i,0} + \psi_{0,j} - \psi_{0,0} + \gamma_{0,0} d_{u,i} d_{v,j} \end{cases} \quad \forall i, j = \pm 1. \quad (25)$$

2.2.6 Field Interpolation

A basis of normalized vectors is defined at a vertex $P_{0,0}$. This basis consists of three vectors, $\mathbf{U}_{0,0}$, $\mathbf{V}_{0,0}$, and $\mathbf{W}_{0,0}$:

$$\mathbf{U}_{0,0} = \begin{pmatrix} u_{0,0,x} \\ u_{0,0,y} \\ \alpha_{0,0} \end{pmatrix}, \quad \mathbf{V}_{0,0} = \begin{pmatrix} v_{0,0,x} \\ v_{0,0,y} \\ \beta_{0,0} \end{pmatrix}, \quad \mathbf{W}_{0,0} = \begin{pmatrix} w_{0,0,x} \\ w_{0,0,y} \\ \gamma_{0,0} \end{pmatrix} \quad (26)$$

where the spatial subvectors $\tilde{u}_{0,0}$, $\tilde{v}_{0,0}$, and $\tilde{w}_{0,0}$ (from Fig. 3.1(b)) are given by:

$$\begin{cases} \tilde{u}_{0,0} = \frac{M_{1,0} - M_{0,0}}{d_{u,1}} \\ \tilde{v}_{0,0} = \frac{M_{0,1} - M_{0,0}}{d_{v,1}} \\ \tilde{w}_{0,0} = \frac{M_{1,1} + M_{0,0} - M_{0,1} - M_{1,0}}{d_{u,1} d_{v,1}} \end{cases} \quad (27)$$

The basis is written at each vertex, the Bézier parameterization of $\mathbf{P} = (x, y, \psi)$ on element E_K can be rewritten as:

$$\mathbf{P}(s, t) = \sum_{k=1}^4 \tilde{\mathbf{P}}_{i_k}(s, t) = \sum_{i=1}^4 \begin{pmatrix} \tilde{x}_{i_k}(s, t) \\ \tilde{y}_{i_k}(s, t) \\ \psi_{i_k}(s, t) \end{pmatrix}, \quad (28)$$

where index i denotes vertex i . The four components $\tilde{\mathbf{P}}_i$ are given by:

$$\left\{ \begin{array}{l} \tilde{\mathbf{P}}_{i_1}(s, t) = (1-s)^2(1-t)^2[(1+2s)(1+2t)\mathbf{P}_{i_1} + 3s(1+2t)d_{u,i_1}\mathbf{U}_{i_1} \\ \quad + 3t(1+2s)d_{v,i_1}\mathbf{V}_{i_1} + 9std_{u,i_1}d_{v,i_1}\mathbf{W}_{i_1}], \\ \tilde{\mathbf{P}}_{i_2}(s, t) = s^2(1-t)^2[(3-2s)(1+2t)\mathbf{P}_{i_2} + 3(1-s)(1+2t)d_{u,i_2}\mathbf{U}_{i_2} \\ \quad + 3t(3-2s)d_{v,i_2}\mathbf{V}_{i_2} + 9(1-s)td_{u,i_2}d_{v,i_2}\mathbf{W}_{i_2}], \\ \tilde{\mathbf{P}}_{i_3}(s, t) = s^2t^2[(3-2s)(3-2t)\mathbf{P}_{i_3} + 3(1-s)(3-2t)d_{u,i_3}\mathbf{U}_{i_3} \\ \quad + 3(3-2s)(1-t)d_{v,i_3}\mathbf{V}_{i_3} + 9(1-s)(1-t)d_{u,i_3}d_{v,i_3}\mathbf{W}_{i_3}], \\ \tilde{\mathbf{P}}_{i_4}(s, t) = (1-s)^2t^2[(1+2s)(3-2t)\mathbf{P}_{i_4} + 3s(3-2t)d_{u,i_4}\mathbf{U}_{i_4} \\ \quad + 3(1+2s)(1-t)d_{v,i_4}\mathbf{V}_{i_4} + 9s(1-t)d_{u,i_4}d_{v,i_4}\mathbf{W}_{i_4}], \end{array} \right. \quad (29)$$

Each of the $(\tilde{\mathbf{P}}_i)$ can be set into the general form:

$$\tilde{\mathbf{P}}_i(s, t) = b_{i,1|K}(s, t)\mathbf{P}_i + b_{i,2|K}(s, t)\mathbf{U}_i + b_{i,3|K}(s, t)\mathbf{V}_i + b_{i,4|K}(s, t)\mathbf{W}_i, \quad (30)$$

$b_{i,j}^{(K)}(s, t)$ are the Bézier basis functions.

Elements with a higher order of continuity are also available in the JOREK code, although not considered in this work. A more detailed discussion on geometrical continuity of Bézier patches can be found in [34].

2.2.7 Bounding Box of a Bicubic Bézier Element

As we will see in section 3.6, for some algorithms it is important to know the bounding box of each element. The bounding box of an element is defined as the smallest, axis-aligned rectangle that contains all the points belonging to an element.

Let's start by considering the physical coordinates (R, Z) , at a specific toroidal angle ϕ they are a function of the local coordinates $(s, t) \in [0, 1]^2$:

$$\mathbf{x}(s, t) = \begin{pmatrix} R(s, t) \\ Z(s, t) \end{pmatrix} = \sum_{i=0}^3 \sum_{j=0}^3 \mathbf{P}_{i,j}(\phi) B_i^3(s) B_j^3(t) \quad (31)$$

Finding the extreme points of $R(s, t)$ and $Z(s, t)$ would require solving $\nabla R(s, t) = 0$ and $\nabla Z(s, t) = 0$ over the domain $(s, t) \in [0, 1]^2$, an expensive problem to solve for each element. A robust and efficient approximation consists of finding the extrema on the boundaries of the element. This assumption holds for the meshes taken into consideration, and is enforced at the start of the simulation by assuring that the Jacobian determinant does not change sign inside the element.

The search for extrema is restricted to the four edges of the element: $s = 0, s = 1, t = 0$, and $t = 1$. Consider the coordinate R along the edge $t = 0$, for $s \in [0, 1]$. The function $R(s, t = 0)$ simplifies to a 1D cubic Bézier curve:

$$R(s, t = 0) = \sum_{i=0}^3 R_{i,0}(\phi) B_i^3(s) \quad (32)$$

This is a cubic polynomial in s . To find its local extrema for $s \in (0, 1)$, we must find the roots of its derivative with respect to s :

$$\frac{d}{ds} R(s, t = 0) = \frac{d}{ds} \left(\sum_{i=0}^3 R_{i,0}(\phi) B_i^3(s) \right) = 0 \quad (33)$$

Since the Bernstein polynomials $B_i^3(s)$ are cubic in s , their derivative is quadratic. This results in a simple quadratic equation of the form $as^2 + bs + c = 0$, which can be efficiently solved for each boundary, for each dimension.

By taking the minimum and maximum along each boundary, we get the corners of the bounding box for each element.

2.2.8 Toroidal Fourier expansion

To account for the 3D nature of the problem, we need to discretize along the toroidal direction, especially in non-axisymmetric devices such as stellarators. Every physical or geometric quantity can be expressed in a Fourier series of the toroidal angle Φ . Since everything we are interested in is real-valued, we use a basis consisting of real trigonometric functions. Any quantity X is expressed as:

$$X(s, t, \Phi) = \sum_{l=1}^{n_{\text{tor}}} X_l(s, t) Z_l(\Phi) \quad (34)$$

where X_l are the real-valued Fourier coefficients (which depend on the poloidal coordinates), and the basis functions $Z_l(\Phi)$ are defined in [37] as:

$$Z_l(\Phi) = \begin{cases} 1 & \text{if } l = 1 \quad (\text{for the } n = 0 \text{ mode}) \\ \cos(n_{\text{period}} \frac{l}{2} \Phi) & \text{if } l > 1 \text{ is even} \\ \sin(n_{\text{period}} \frac{l-1}{2} \Phi) & \text{if } l > 1 \text{ is odd} \end{cases} \quad (35)$$

Here, n_{period} denotes the field-periodicity of the configuration, allowing for representations of geometries that complete their pattern multiple times in one full 2π turn.

2.2.9 The Finite Element Method

Now, we have all the tools necessary to build the numerical framework to solve the governing equations that describe the plasma.

To accomplish this task, we use the Finite Element Method [48]. By subdividing the domain into smaller subdomains, looking for a suitable coefficient, we will be able to numerically approximate the function.

In our case, the poloidal domain \mathcal{D} is partitioned into a mesh of non-overlapping Bézier patch finite elements \mathcal{D}_K , such that:

$$\mathcal{D} = \bigcup_K \mathcal{D}_K \quad (36)$$

For each element K , we associate a set of nodes where the degrees of freedom are defined. In addition to the vertices, each element carries four additional control points that influence the shape of the basis functions within the patch. The local basis functions are $b_{i,j}^{(K)}(s,t)$, where (s,t) are local coordinates within the element \mathcal{D}_K , and the indices i,j represent the nodes and control points associated with it.

The solution within each element is expressed as a linear combination of these local basis functions:

$$\psi^{(K)}(s,t) = \sum_{i,j} \psi_{i,j}^{(K)} b_{i,j}^{(K)}(s,t) \quad (37)$$

As previously noted, the solution needs to be three-dimensional, so we take that into account by introducing the Fourier expansion along the toroidal angle Φ :

$$\psi^{(K)}(s,t,\Phi) = \sum_{l=1}^{n_{\text{tor}}} \sum_{i,j} \psi_{i,j,l}^{(K)} b_{i,j}^{(K)}(s,t) Z_l(\Phi) \quad (38)$$

Where:

- $Z_l(\Phi)$ are the toroidal basis functions
- $\psi_{i,j,l}^{(K)}$ are the local degrees of freedom associated with the l -th toroidal mode and the (i,j) -th local basis function within element K ,
- $b_{i,j}^{(K)}(s,t)$ are the Bézier basis functions defined locally over element \mathcal{D}_K .

2.3. Governing equations and Numerical Approximations for the kinetics

In this section, we are going to provide a general overview of the kinetics equations and the numerical approximations considered, as well as their stability. For a detailed description, see [12].

2.3.1 Newton-Lorentz equation

We study the motion of charged particles in a given electromagnetic field, according to Newton's equations:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}, \quad (39)$$

$$m \frac{d\mathbf{v}}{dt} = q[\mathbf{E}(\mathbf{x}) + \mathbf{v} \times \mathbf{B}(\mathbf{x})], \quad (40)$$

Where \mathbf{x} (\mathbf{v}) is the position (velocity) of the particle with mass m and charge q , while \mathbf{E} (\mathbf{B}) is the given electric (magnetic) field.

The electromagnetic field is always considered given, as it is calculated by solving the MHD equations before starting the kinetic solver, and is considered a static field during the particle pushing.

2.3.2 The Boris Pusher

For the case under study, we chose to employ an explicit time stepping scheme called the Boris pusher, an integration scheme with second order accuracy and comparably light computational requirements.

This algorithm is considered a leap-frog type of algorithm, since it employs staggered updates of position and velocity in time. In particular, velocities are updated at time $t = t^{n+1/2} = (n + \frac{1}{2})\Delta t$ and position are updated like $t = t^n = n\Delta t$.

The following discussion stems from [12], which presents a detailed analysis of the numerical properties of such algorithms, out of scope for the discussion carried out in this work.

The algorithm separates the two different equations, the electric and the magnetic parts of the Lorentz force, by performing the following three steps:

A half-step update (time step $\Delta t/2$) with the contribution of only the electric field:

$$\mathbf{v}^- = \mathbf{v}^{n-1/2} + \frac{q}{m} \mathbf{E}(\mathbf{x}^n) \frac{\Delta t}{2}. \quad (41)$$

Perform a full step (time step Δt) following the magnetic field:

$$\mathbf{v}' = \mathbf{v}^- + f^{n,\Delta t} \mathbf{v}^- \times \mathbf{B}(\mathbf{x}^n), \quad (42)$$

$$\mathbf{v}^+ = \mathbf{v}^- + \frac{2f^{n,\Delta t}}{1 + (f^{n,\Delta t})^2 \mathbf{B}(\mathbf{x}^n)^2} \mathbf{v}' \times \mathbf{B}(\mathbf{x}^n), \quad (43)$$

with

$$f^{n,\Delta t} = \frac{\tan(\frac{q}{m} \frac{\Delta t}{2} |\mathbf{B}(\mathbf{x}^n)|)}{|\mathbf{B}(\mathbf{x}^n)|}. \quad (44)$$

Perform another half-step update (time step $\Delta t/2$) with acceleration only due to the electric field:

$$\mathbf{v}^{n+1/2} = \mathbf{v}^+ + \frac{q}{m} \mathbf{E}(\mathbf{x}^n) \frac{\Delta t}{2}. \quad (45)$$

Once the velocity update is completed, the position can be updated by

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2} \Delta t. \quad (46)$$

This algorithm is second-order accurate and time reversible [12]. Of course, as for any leap-frog algorithm, the accuracy also depends on the choice of the initial step that needs to be provided to relate the (known) particle position and velocity at time $t = 0$ to the velocity $\mathbf{v}^{-1/2}$ (or $\mathbf{v}^{1/2}$) that the algorithm needs to get started.

2.3.3 Projecting particles on the Finite Element grid

To couple the kinetic particles with the fluid part, it is necessary to translate the discrete information from the particles into a continuous field compatible with our Finite Element formulation of the MHD equations. We employ a projection using the previously established methodologies. The goal is to find a continuous field, p , that is equivalent in a weighted-average sense to the source data from the particles, X .

This equivalence is formally expressed in the weak form, which states that the two quantities are identical when integrated against any valid test function v :

$$\int_V p v dV = \int_V X v dV \quad (47)$$

Here, p is the projected field to be solved for (e.g., mass density, parallel velocity), and X represents the corresponding moment of the particle distribution function, 1 for number density, L_z for radiative energy loss, m for mass density, v_{\parallel} for parallel velocity, E for energy et cetera [11].

Usually, in these particle methods, we introduce a shape function $S(x)$ describing the shape of the particle in space. Various shape functions can be chosen and consequently give rise to different types of smoothing effects on the particle; by including these shape functions into the formulation, we get:

$$\int_V p v dV = \int_V \left(\sum_{i=1}^N S(\mathbf{x} - \mathbf{x}_i) w_i X \right) v dV \quad (48)$$

The term w_i indicates the weight we give to each particle in the simulation. This is done to create macroparticles, whose evolution is tied together but which carry different amounts of physical properties.

Matrix construction

Let's now look at the single element to construct the matrix representing the linear system of equations that needs to be solved for the projections. Within each element \mathcal{D}_K , the field p , following the formulation introduced in (38), is approximated as:

$$p^{(K)}(s, t, \Phi) = \sum_{l=1}^{n_{\text{tor}}} \sum_{i,j} p_{i,j,l}^{(K)} b_{i,j}^{(K)}(s, t) Z_l(\Phi) \quad (49)$$

Similarly, the test function v is expressed as:

$$v^{(K)}(s, t, \Phi) = \sum_{m=1}^{n_{\text{tor}}} \sum_{k, \ell} v_{k, \ell, m}^{(K)} b_{k, \ell}^{(K)}(s, t) Z_m(\Phi) \quad (50)$$

The mapping from local coordinates (s, t, Φ) to physical cylindrical coordinates (R, Z, ϕ) is given by:

$$(s, t, \Phi) \mapsto \mathbf{x}(s, t, \Phi) = (R(s, t, \Phi), Z(s, t, \Phi), \phi = \Phi) \quad (51)$$

The Jacobian of the 2D poloidal mapping $(s, t) \mapsto (R, Z)$ is:

$$J_{\text{pol}}^{(K)}(s, t, \Phi) = \det \begin{bmatrix} \frac{\partial R}{\partial s} & \frac{\partial R}{\partial t} \\ \frac{\partial Z}{\partial s} & \frac{\partial Z}{\partial t} \end{bmatrix} \quad (52)$$

Note that $J_{\text{pol}}^{(K)}$ depends in general on all three coordinates s , t , and Φ due to the dependence of R and Z . The physical volume element $dV = R dR dZ d\phi$ transforms to the local coordinates as:

$$dV = R(s, t, \Phi) J_{\text{pol}}^{(K)}(s, t, \Phi) ds dt d\Phi =: \mathcal{J}^{(K)}(s, t, \Phi) ds dt d\Phi \quad (53)$$

where $\mathcal{J}^{(K)} = R \cdot J_{\text{pol}}^{(K)}$ is the determinant of the full 3D coordinate transformation Jacobian. The element-wise contribution to the left-hand side of equation (48) is:

$$\int_{\mathcal{D}_K} \int_0^{2\pi} p^{(K)}(s, t, \Phi) v^{(K)}(s, t, \Phi) \mathcal{J}^{(K)}(s, t, \Phi) d\Phi ds dt \quad (54)$$

Substituting the basis function expansions for $p^{(K)}$ and $v^{(K)}$ gives the element mass matrix contribution. The equation for an arbitrary test function v is satisfied if it holds for each basis function. This leads to a linear system where the element mass matrix $M^{(K)}$ couples the degrees of freedom $p_{i,j,l}^{(K)}$. A single entry of this matrix, which couples the poloidal basis functions indexed by (i, j) and (k, ℓ) and the toroidal modes l and m , is given by:

$$M_{ij,k\ell,lm}^{(K)} = \int_0^{2\pi} \left[\int_{\mathcal{D}_K} b_{i,j}^{(K)}(s, t) b_{k,\ell}^{(K)}(s, t) \mathcal{J}^{(K)}(s, t, \Phi) ds dt \right] Z_l(\Phi) Z_m(\Phi) d\Phi \quad (55)$$

To evaluate this integral numerically, we can conceptually separate the integration over the poloidal plane from the integration over the toroidal angle. Let's define an intermediate function, $F_{ij,k\ell}(\Phi)$, which represents the result of the poloidal integral for a given toroidal angle Φ :

$$F_{ij,k\ell}(\Phi) := \int_{\mathcal{D}_K} b_{i,j}^{(K)}(s, t) b_{k,\ell}^{(K)}(s, t) \mathcal{J}^{(K)}(s, t, \Phi) ds dt \quad (56)$$

The integral over the 2D reference element \mathcal{D}_K is approximated using a fourth-order Gaussian quadrature rule. This is constructed as the product of two 1D rules, leading to a double summation over the quadrature points (s_{ms}, t_{mt}) with their corresponding 1D weights ω_{ms} and ω_{mt} :

$$F_{ij,k\ell}(\Phi) \approx \sum_{ms=1}^{n_{\text{gp}}} \sum_{mt=1}^{n_{\text{gp}}} \omega_{ms} \omega_{mt} b_{i,j}^{(K)}(s_{ms}, t_{mt}) b_{k,\ell}^{(K)}(s_{ms}, t_{mt}) \mathcal{J}^{(K)}(s_{ms}, t_{mt}, \Phi) \quad (57)$$

With this intermediate function, the expression for the mass matrix entry simplifies to a single integral over the toroidal angle:

$$M_{ij,k\ell,lm}^{(K)} = \int_0^{2\pi} F_{ij,k\ell}(\Phi) Z_l(\Phi) Z_m(\Phi) d\Phi \quad (58)$$

Since the integral in (58) is integrating a periodic function, it can be evaluated using a periodic Gaussian quadrature, which is exact with N points for trigonometric polynomials of degree $2N - 2$ [6]. We therefore sample the function in n_{plane} equidistant points along the toroidal direction.

$$M_{ij,k\ell,lm}^{(K)} = \frac{2\pi}{n_{\text{plane}}} \sum_{p=1}^{n_{\text{plane}}} F_{ij,k\ell}(\Phi_p) Z_l(\Phi_p) Z_m(\Phi_p) \quad (59)$$

Uncoupled Harmonics in an Axisymmetric Geometry

The coupling between different toroidal harmonics on the left-hand side comes directly from the geometry of the mesh, see eq. (55). The term responsible for coupling the toroidal modes l and m is the full 3D Jacobian determinant, $\mathcal{J}^{(K)}(s, t, \Phi) = R(s, t, \Phi) J_{\text{pol}}^{(K)}(s, t, \Phi)$. If this Jacobian is a function of the toroidal angle Φ , the integral over Φ will generally be different from zero for $l \neq m$.

An important simplification happens if the mesh grid is axisymmetric. In an axisymmetric grid, the poloidal section is the same, independently of toroidal angle Φ . This means the mapping from the local coordinates (s, t) to the physical poloidal coordinates (R, Z) is independent of Φ :

$$(s, t) \mapsto \mathbf{x}_{\text{pol}}(s, t) = (R(s, t), Z(s, t)) \quad (\text{independent of } \Phi) \quad (60)$$

Therefore, the derivatives are also independent of Φ . This directly changes the Jacobian in the poloidal section by providing independence from the toroidal direction.

With this simplification, the mass matrix integral can be rearranged, as the term in the square brackets is no longer a function of Φ :

$$M_{ij,k\ell,lm}^{(K)} = \left[\int_{\mathcal{D}_K} b_{i,j}^{(K)}(s, t) b_{k,\ell}^{(K)}(s, t) \mathcal{J}^{(K)}(s, t) ds dt \right] \left[\int_0^{2\pi} Z_l(\Phi) Z_m(\Phi) d\Phi \right] \quad (61)$$

The basis set $\{Z_l(\Phi)\}$ is constructed from sines, cosines, and a constant term, which are orthogonal over a period. This means the integral is zero unless the two basis functions are the same:

$$\int_0^{2\pi} Z_l(\Phi) Z_m(\Phi) d\Phi = C_l \delta_{lm} \quad (62)$$

where C_l is the normalization constant given by

$$C_l = \begin{cases} 2\pi & \text{if } l = 1 \\ \pi & \text{if } l > 1 \end{cases} \quad (63)$$

and δ_{lm} is the Kronecker delta.

The appearance of the delta means that the matrix entry $M_{ij,k\ell,lm}^{(K)}$ is different from zero only if the toroidal mode indices are identical ($l = m$). Therefore, for an axisymmetric grid like the already existing projections for tokamak cases in the code, the global mass matrix is perfectly block-diagonal when rearranged to cluster the harmonics contribution together.

Each harmonic is decoupled, allowing the full linear system to be solved as a set of smaller, independent systems. A similar scheme is employed in the iterative solver of the fluid as an effective and efficient preconditioner for the fluid matrix [20, 21].

This is not the case for stellarator scenarios considered in this work.

Fast Fourier Transform for the evaluation of the toroidal integral

For stellarator simulations, a high number of harmonics is usually needed to accurately represent solutions on the plasma, given the intrinsic coupling of harmonics due to the non-axisymmetric geometry; the matrix assembly becomes a bottleneck for simulations. However, there is a computationally more efficient algorithm, used in the fluid part, to evaluate the toroidal integral by using the Fast Fourier Transform [9].

Let's define the integer wavenumber ν :

$$\nu(l) = \lfloor l/2 \rfloor \implies \nu(1) = 0, \nu(2) = \nu(3) = 1, \nu(4) = \nu(5) = 2, \dots \quad (64)$$

Such that the couple of \sin and \cos with the same periodicity can be addressed by the same wavenumber.

With these identities, every real basis function can be rewritten as

$$Z_{2q}(\Phi) = \cos(q\Phi) = \frac{1}{2}(e^{iq\Phi} + e^{-iq\Phi}), \quad Z_{2q+1}(\Phi) = \sin(q\Phi) = \frac{1}{2i}(e^{iq\Phi} - e^{-iq\Phi}). \quad (65)$$

Therefore, any product $Z_l Z_m$ can be written as a linear combination of two complex exponentials:

$$Z_l(\Phi) Z_m(\Phi) = \frac{1}{2} \left(e^{i[\nu(l)-\nu(m)]\Phi} + \sigma_{lm} e^{i[\nu(l)+\nu(m)]\Phi} \right) \quad (66)$$

where the coefficient σ_{lm} depends only on the parity of the two indices:

$$\sigma_{lm} = \begin{cases} +1 & l \text{ and } m \text{ have the same parity } [\cos \times \cos \text{ or } \sin \times \sin], \\ -1 & l \text{ and } m \text{ have opposite parity } [\cos \times \sin \text{ or } \sin \times \cos]. \end{cases} \quad (67)$$

The two exponential factors carry wavenumbers

$$r_1 := \nu(l) - \nu(m), \quad r_2 := \nu(l) + \nu(m), \quad (68)$$

both of which obey $|r_{1,2}| < n_{\text{plane}}/2$ in practice, because we strictly choose $n_{\text{plane}} \geq 2n_{\text{tor}}$ in the simulations. Inserting (66) into (59) and distributing the summation gives:

$$\begin{aligned} M_{ij,k\ell,lm}^{(K)} &= \frac{\pi}{n_{\text{plane}}} \sum_{p=1}^{n_{\text{plane}}} F_{ij,k\ell}(\Phi_p) \left[e^{(ir_1\Phi_p)} + \sigma_{lm} e^{(ir_2\Phi_p)} \right] \\ &= \frac{\pi}{n_{\text{plane}}} \left[\sum_{p=1}^{n_{\text{plane}}} F_{ij,k\ell}(\Phi_p) e^{(ir_1\Phi_p)} + \sigma_{lm} \sum_{p=1}^{n_{\text{plane}}} F_{ij,k\ell}(\Phi_p) e^{(ir_2\Phi_p)} \right] \end{aligned} \quad (69)$$

We can now define the complex discrete Fourier coefficients:

$$\widehat{F}_{ij,k\ell}^r := \sum_{p=1}^{n_{\text{plane}}} F_{ij,k\ell}(\Phi_p) e^{-i r \Phi_p}, \quad r = 1, \dots, n_{\text{plane}}. \quad (70)$$

and by substituting them into (70) we obtain:

$$M_{ij,k\ell,lm}^{(K)} = \frac{\pi}{n_{\text{plane}}} \left[\widehat{F}_{ij,k\ell}^{-r_1} + \sigma_{lm} \widehat{F}_{ij,k\ell}^{-r_2} \right] \quad (71)$$

Since all the coefficients can be obtained from a single FFT computation, the asymptotic complexity is greatly reduced.

When using the direct summation, for each degree of freedom on the poloidal section, we have to evaluate the function in n_{plane} different points for each couple l, m of toroidal harmonics:

$$\text{cost}_{\text{direct}} = n_{\text{tor}}^2 \times O(n_{\text{plane}}) = O(n_{\text{tor}}^2 n_{\text{plane}}). \quad (72)$$

On the other hand, using the FFT acceleration, we compute all the coefficients in a single FFT pass of complexity $O(n_{\text{plane}} \log(n_{\text{plane}}))$. Once all the coefficients are obtained, we construct the matrix by carefully adding the relevant coefficients for each couple following (66), resulting in a final complexity of:

$$\text{cost}_{\text{FFT}} = O(n_{\text{plane}} \log n_{\text{plane}} + n_{\text{tor}}^2). \quad (73)$$

In the direct method, the expensive $O(n_{\text{plane}})$ summation is repeated N_{tor}^2 times. In the FFT method, the expensive part, with a slightly higher cost of $O(n_{\text{plane}} \log n_{\text{plane}})$, is performed only once. The subsequent assembly is extremely fast.

Right hand side

We consider now the right-hand side of the weak formulation from (48):

$$\int_V \left(\sum_{i=1}^N S(\mathbf{x} - \mathbf{x}_i) w_i X(\mathbf{x}) \right) v(\mathbf{x}) dV \quad (74)$$

The element-wise contribution is obtained by substituting the expansion for v and the transformation for dV :

$$\sum_{m=1}^{n_{\text{tor}}} \sum_{k,\ell} v_{k,\ell,m}^{(K)} \int_{\mathcal{D}_K} \int_0^{2\pi} \left(\sum_{i=1}^N S(\mathbf{x}(s,t,\Phi) - \mathbf{x}_i) w_i X(\mathbf{x}(s,t,\Phi)) \right) b_{k,\ell}^{(K)}(s,t) Z_m(\Phi) \mathcal{J}^{(K)}(s,t,\Phi) d\Phi ds dt \quad (75)$$

We define the right-hand-side vector coefficients $L_{k,\ell,m}^{(K)}$ from the integral term as:

$$L_{k,\ell,m}^{(K)} = \int_{\mathcal{D}_K} \int_0^{2\pi} \left(\sum_{i=1}^N S(\mathbf{x}(s,t,\Phi) - \mathbf{x}_i) w_i X(\mathbf{x}_i) \right) b_{k,\ell}^{(K)}(s,t) Z_m(\Phi) \mathcal{J}^{(K)}(s,t,\Phi) d\Phi ds dt \quad (76)$$

where for brevity we use $X(\mathbf{x}_i)$ as the value of the moment for particle i .

The evaluation of this integral depends on the choice of the shape function S . The simplest and most computationally efficient choice is the Dirac delta function, $S(\mathbf{x} - \mathbf{x}_i) = \delta(\mathbf{x} - \mathbf{x}_i)$. Using this choice, the integral in equation (76) can be greatly simplified.

The Dirac delta function transforms between coordinate systems according to the rule:

$$\delta(\mathbf{x} - \mathbf{x}_i) = \frac{\delta(s - s_i)\delta(t - t_i)\delta(\Phi - \Phi_i)}{|\mathcal{J}^{(K)}(s_i, t_i, \Phi_i)|} \quad (77)$$

where (s_i, t_i, Φ_i) are the local coordinates of the particle at position \mathbf{x}_i . This transformation is valid only for particles located within the element K ; otherwise, the delta function is zero over the integration domain $\mathcal{D}_K \times [0, 2\pi]$.

Substituting this into the expression for $L_{k,\ell,m}^{(K)}$ and swapping the sum and integral, we get:

$$L_{k,\ell,m}^{(K)} = \sum_{i \in K} \int_{\mathcal{D}_K} \int_0^{2\pi} \frac{\delta(s - s_i)\delta(t - t_i)\delta(\Phi - \Phi_i)}{|\mathcal{J}^{(K)}(s_i, t_i, \Phi_i)|} w_i X(\mathbf{x}_i) b_{k,\ell}^{(K)}(s, t) Z_m(\Phi) \mathcal{J}^{(K)}(s, t, \Phi) d\Phi ds dt \quad (78)$$

The summation is now restricted to particles $i \in K$ that reside within the element K . By the sifting property of the delta function, the integral collapses to an evaluation of the integrand at the particle's local coordinates (s_i, t_i, Φ_i) :

$$L_{k,\ell,m}^{(K)} = \sum_{i \in K} w_i X(\mathbf{x}_i) b_{k,\ell}^{(K)}(s_i, t_i) Z_m(\Phi_i) \frac{\mathcal{J}^{(K)}(s_i, t_i, \Phi_i)}{|\mathcal{J}^{(K)}(s_i, t_i, \Phi_i)|} \quad (79)$$

Assuming the element mapping is not inverted, the Jacobian determinant $\mathcal{J}^{(K)}$ is positive, and the fraction equals one. This reduces the computationally expensive triple integral to a simple sum over the particles within the element:

$$L_{k,\ell,m}^{(K)} = \sum_{i \in K} w_i X(\mathbf{x}_i) b_{k,\ell}^{(K)}(s_i, t_i) Z_m(\Phi_i) \quad (80)$$

This formulation provides a significant computational advantage, as it avoids numerical quadrature and only requires finding which particles reside in which element and evaluating the basis functions at the particle locations.

So, the element-wise right-hand side of the weak form becomes:

$$\sum_{m=1}^{n_{\text{tor}}} \sum_{k,\ell} v_{k,\ell,m}^{(K)} \left(\sum_{i \in K} w_i X(\mathbf{x}_i) b_{k,\ell}^{(K)}(s_i, t_i) Z_m(\Phi_i) \right) \quad (81)$$

3. Algorithms and verification for PiC methods in the JOREK stellarator model

The new particle-in-cell (PiC) capabilities within the JOREK stellarator model are implemented and verified in this section after the theoretical underpinnings were established in the previous section. Here, the development and verification of the algorithms needed for a fully functional hybrid simulation are the main priorities. In order to verify that the basic particle tracing operates as anticipated in both tokamak and stellarator configurations, we start by presenting initial tests. Expanding upon this, we describe the main contribution of this work: the application of a fully coupled harmonic projection scheme, which is necessary to accurately depict fields in stellarator geometries that are not axisymmetric. First, qualitative projections onto realistic stellarator meshes are used to illustrate the efficacy of this approach, which is then quantitatively validated through a series of convergence tests. Lastly, we implement and evaluate a spatial R-Tree data structure to tackle the problem of effectively locating particles within the 3D mesh.

3.1. Preliminary tests

3.1.1 Particle Tracing

In order to produce a working hybrid solver for the stellarator model, it is crucial to have consistent kinetic simulation results with the existing code base. For this purpose, three simulations have been set up and tested in which particles are pushed in static fields: a tokamak configuration in a stellarator model, a tokamak configuration in a tokamak model, and a stellarator configuration in a stellarator model. The first and the second should have an agreeing result, while the third should showcase the non-axisymmetric behavior intrinsic to the stellarator geometry.

The stellarator model used is the model 180, while the tokamak model used is 600

As expected, the first two configurations yield the same result within numerical tolerance as seen in Fig. 4. In contrast, the tracing on the stellarator case follows the characteristic geometry of the stellarator (Fig. 5).

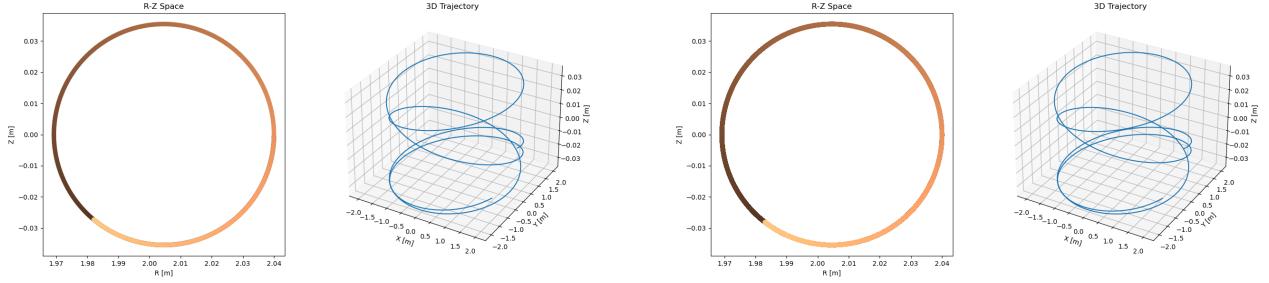


Figure 4: Comparison of particle trajectories between stellarator and tokamak models in tokamak configuration.

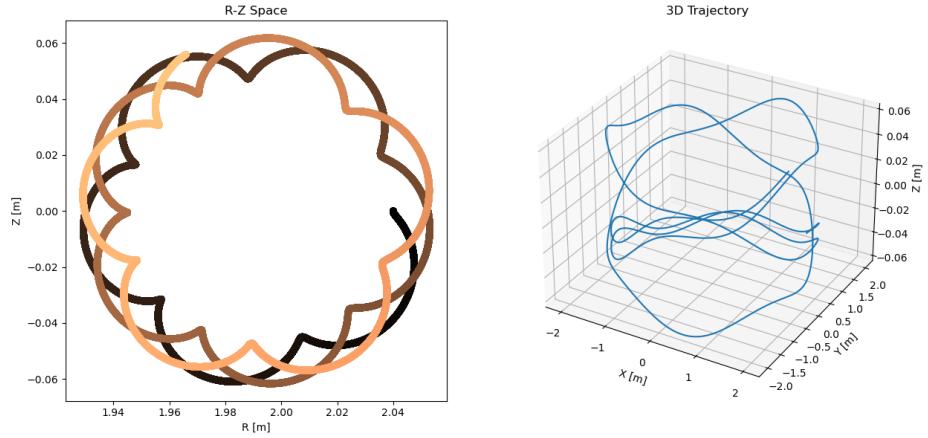


Figure 5: Particle Tracing in a stellarator case using the stellarator model

3.1.2 Different Trajectories

In toroidal fusion plasmas like tokamaks, neoclassical transport arises from collisions in the toroidally inhomogeneous magnetic field, which is fundamentally different from classical transport in a uniform field. In a simple cylindrical model (“classical” limit), cross-field diffusion is set by the random walk of gyrocenters (step size, the Larmor radius) [18, 44]. However, the tokamak’s toroidal geometry makes the field stronger on the inboard side, so particles with small parallel velocity become magnetically trapped and bounce between mirror points on the outboard side. These trapped particles execute banana-shaped orbits in the poloidal plane. [18].

By choosing an appropriate initial position and velocity for some particles, we should be able to reproduce banana and trapped orbits in the tokamak. These serve as a first step experiment to check if the stellarator code in JOREK is suitable to reproduce the effects of neoclassical transport and consequently see if the results from the pure tokamak model are in agreement.

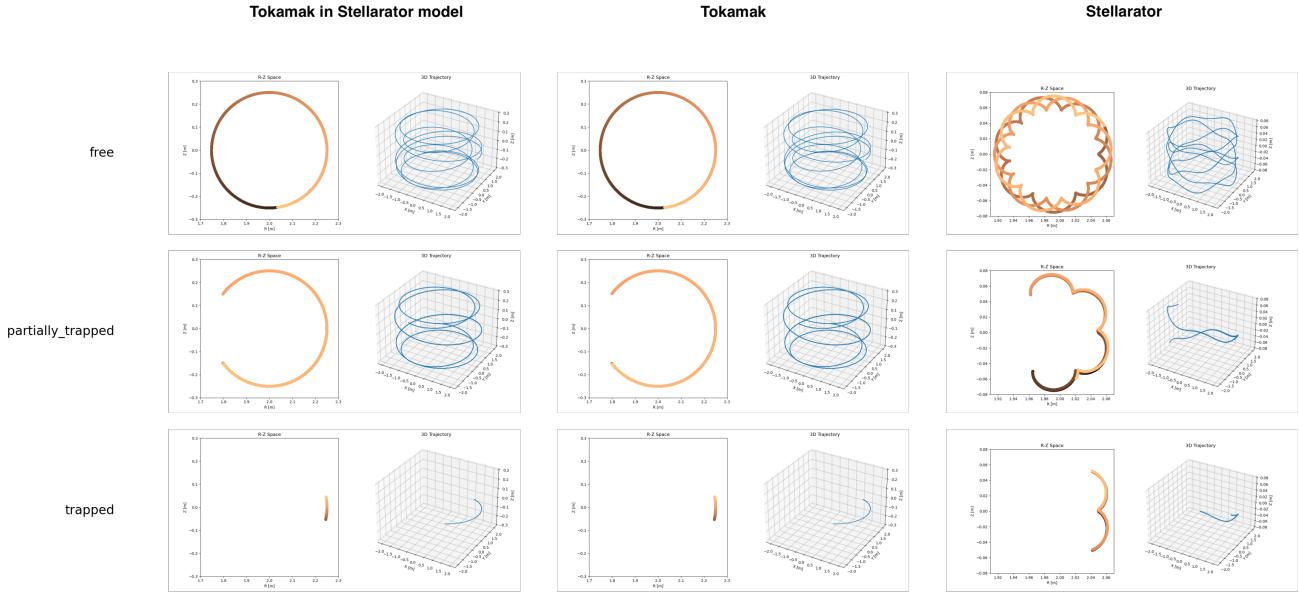


Figure 6: The three types of particle trajectories in the tested cases

As expected, the results show perfectly matching results between the models, assuring compatibility for particle pushing, see Fig. 6.

3.1.3 Particle Projection

While the pusher allows for particles to follow the field calculated on the fluid timestepping, at some point, it is necessary to project the particles on the fluid mesh to retrieve useful properties on the particle moving through the plasma. To achieve this, in the axis-symmetric tokamak, each harmonic is retrieved by solving a separate linear system.

As this is the complementary step for kinetic simulations, more tests have been performed to ensure agreement in the different models. To achieve this, a special three-dimensional Gaussian particle distribution has been employed spanning across the two poloidal dimensions and on the toroidal dimension.

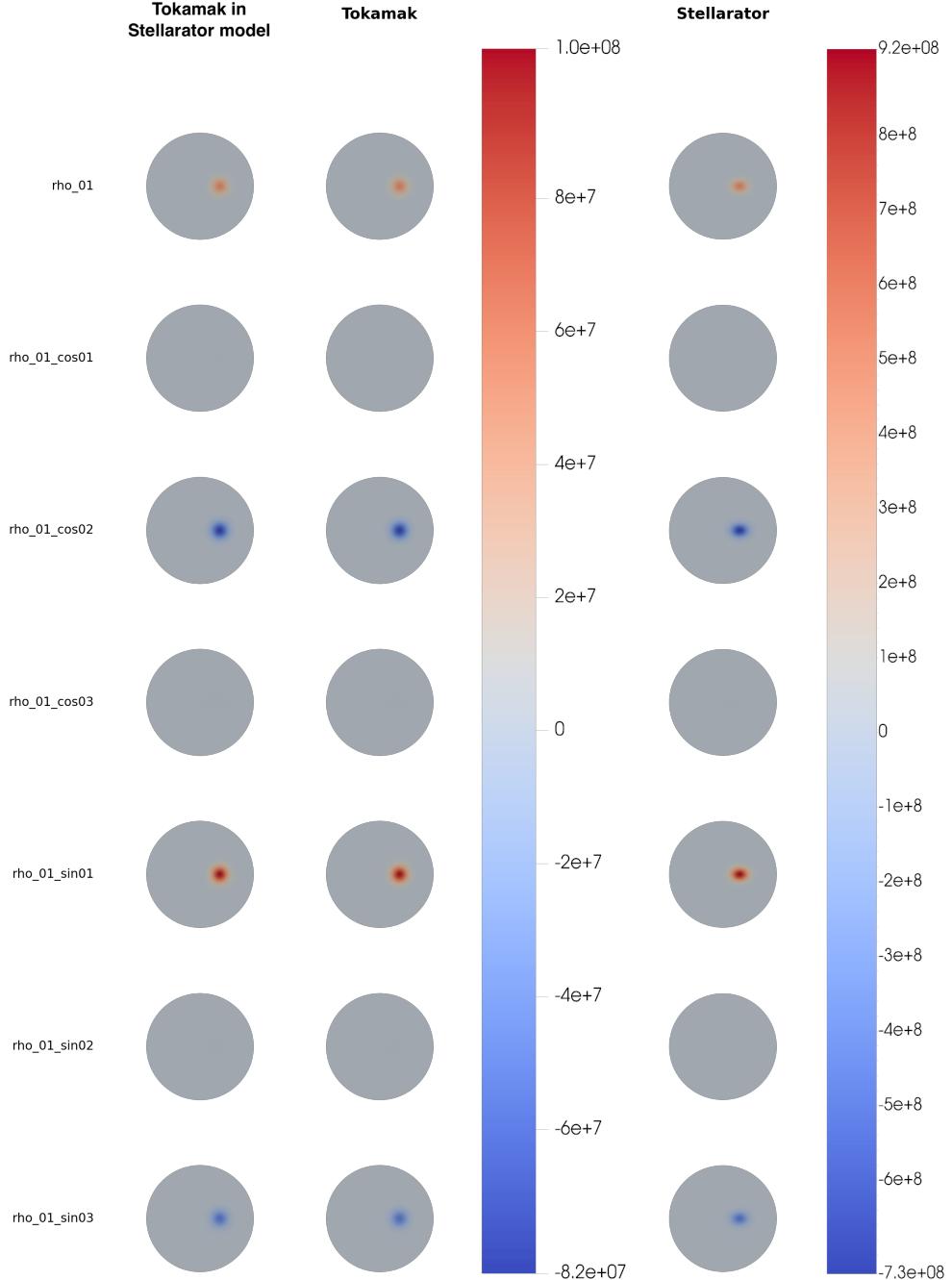


Figure 7: Projection of three-dimensional Gaussian distribution

The result of the aforementioned distribution when projected is a two-dimensional Gaussian, decomposed in its harmonics (Fig. 7).

As expected, the results in the two tokamak-like cases are sound across models. Here though we can have a first appreciation of the problem revolving around projection in the stellarator case when using the uncoupled harmonics system, since the geometry is non-axisymmetrical we cannot treat each harmonic separately and solving their respective linear system, we have to couple all the harmonics together and account for the three dimensional variation of the grid along the toroidal direction in the Jacobian.

3.2. Implementation of the coupled harmonics

The mathematical derivation done in (58) highlights the system for which we are trying to solve; this required some significant modifications to the code, as the infrastructure heavily relied on splitting the system among

the harmonics.

3.2.1 Changes in the matrix system

Before jumping directly to the results, it is important to verify that our mathematical assumptions and conclusions are well reflected in the practical solution implemented.

The first concrete difference we should be able to see is a Jacobian changing along the toroidal angle:

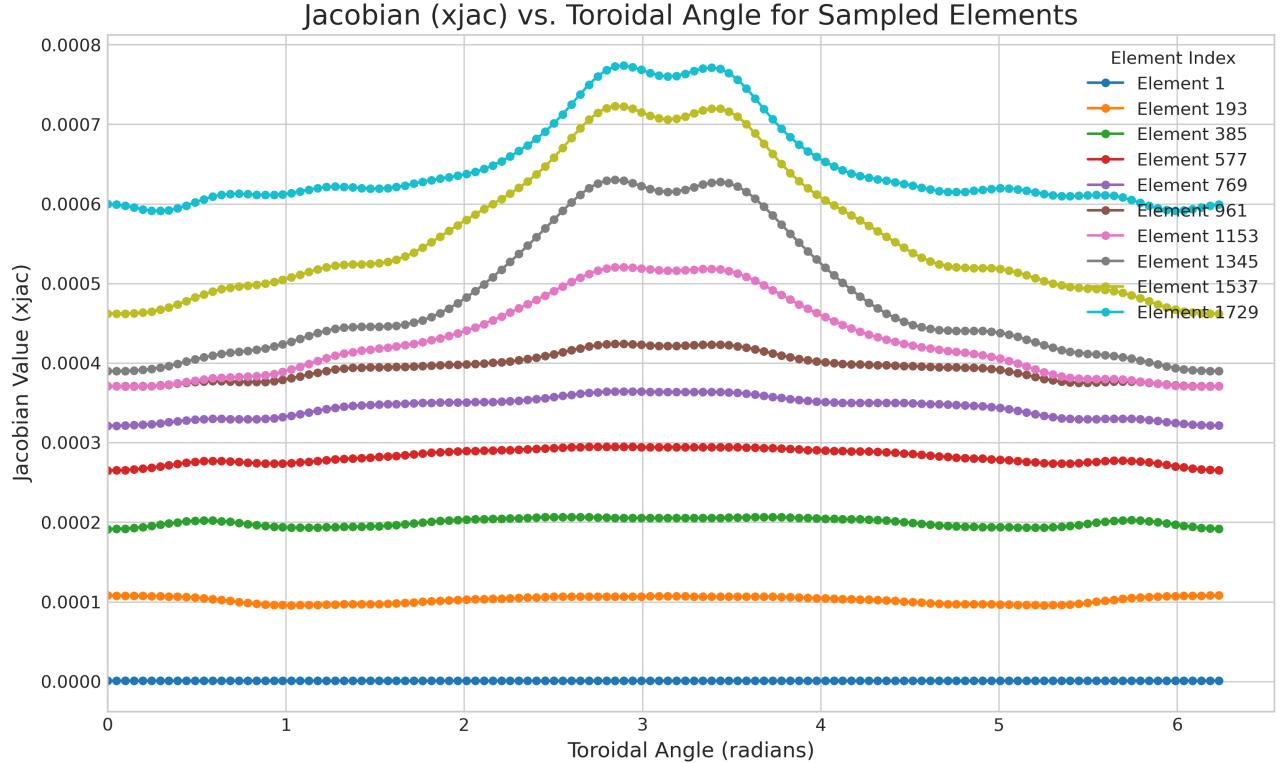
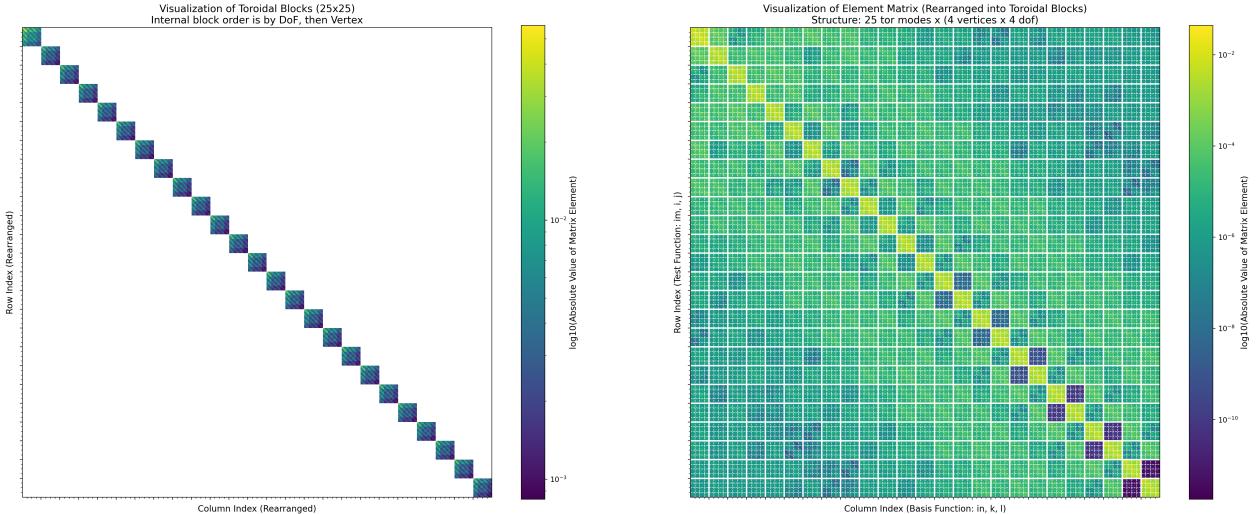


Figure 8: Dependence of the Jacobian on the toroidal angle for several mesh elements. The plot shows a stronger variation for the innermost elements due to their higher geometric distortion near the mesh axis.

As evident from Fig. 8, the Jacobian now presents a dependence on the toroidal angle. Interestingly, we can also observe a stronger influence on the Jacobian in the innermost elements. This is due to the higher distortions towards the mesh axis of the elements.

This dependence on the three-dimensional Jacobian also allows us to create a meaningful system to solve, considering all the harmonics, which is immediately evident from Fig. 9:



(a) Sparsity pattern of a system matrix where the toroidal modes are uncoupled. Each colored block on the diagonal represents the physics within a single toroidal mode number, n . The zero-valued off-diagonal blocks indicate no coupling between different modes.

(b) Sparsity pattern of a system matrix with coupling between toroidal modes. The non-zero off-diagonal blocks represent the interaction between different modes (e.g., coupling between modes n and $n \pm 1$)

Figure 9: Comparison of system matrix structures for a physical model with (a) uncoupled and (b) coupled toroidal modes. The uncoupled case results in a simple block-diagonal matrix, while coupling introduces off-diagonal terms.

3.3. Coupled projection

With all the modifications in place, we can now have a qualitative appreciation of the coupled harmonics by revising the projection proposed in Fig. 7 in the coupled scenario.

Before being able to appropriately represent our complex geometry, it was necessary to take into account the non-axisymmetric geometry of the stellarator. As previously discussed, the assumption on the geometry influenced not only the underlying math of the system but also several support diagnostics along with the code, in particular, the vtk export resulting from projecting particles used the 2D basis to represent the poloidal section without taking into account the toroidal angle; for this reason, an extension to 3D geometries has been developed.

Although the geometry is represented continuously over the toroidal angle $\phi \in [0, 2\pi]$, for clarity, this work will primarily show results at a few discrete toroidal planes

3.3.1 Gaussian distribution on the W7-A

The first test case considered is the Wendelstein 7-A (W7-A)[30], a classical stellarator. While its geometry is non-axisymmetric, it is not subject to drastic deformation. The three-dimensional variation provides a suitable benchmark to initially validate the coupled harmonic projection scheme.

For these tests, we employ a three-dimensional Gaussian of the form:

$$f(R, Z, \phi) = \exp \left(- \left(\frac{(R - R_0)^2}{2\sigma_R^2} + \frac{(Z - Z_0)^2}{2\sigma_Z^2} + \frac{(\phi - \phi_0)^2}{2\sigma_\phi^2} \right) \right) \quad (82)$$

For the Gaussian projected onto W7-A, the parameters chosen are:

$$\begin{aligned} R_0 &= 2.02, & Z_0 &= 0, & \phi_0 &= \frac{\pi}{2} \\ \sigma_R &= \sigma_Z = 0.01, & \sigma_\phi &= \frac{\pi}{2} \end{aligned}$$

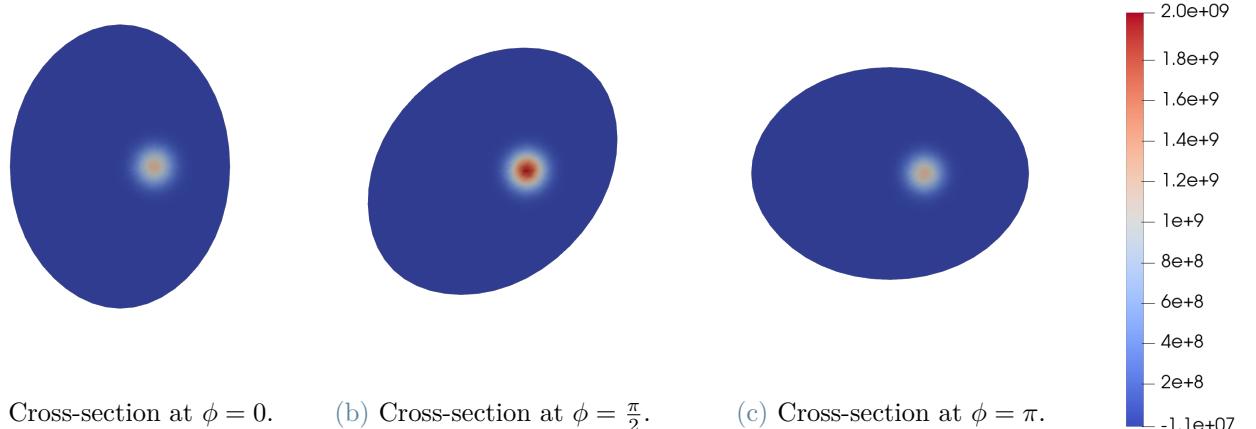


Figure 10: Visualization of the W7-A stellarator’s non-axisymmetric geometry. The poloidal cross-section is shown at three different toroidal angles.

As shown in Fig. 10, even for relatively simple non-axisymmetric geometries, the toroidal dependency of the mesh already has a noticeable influence on the solution; taking into account the full Jacobian, such deformation is correctly accounted for and the shape of the Gaussian is recovered correctly.

3.3.2 Gaussian distribution on the W7-X

While W7-A provides a suitable starting point, it is crucial to move toward more realistic and contemporary geometries, where a strong dependence on the toroidal direction greatly deforms the mesh shape. For these reasons, further tests have been conducted on the geometry of the W7-X [26], the latest stellarator reactor located at the Max Planck Institute for Plasma Physics in Greifswald.

W7-X is a quasi-isodynamic stellarator, and the geometry of the reactor closely resembles the geometries of SQuIDs [15], a promising future reactor design for which, in recent years, great interest has developed.

Due to the highly shaped mesh, the Gaussian distribution cannot be initialized in the same RZ position like in the W7-A case; for this reason, suitable positions are chosen to showcase the most deformed geometries.

The Results shown in Fig. 11 use a 3D Gaussian with parameters: $Z_0 = 0, \sigma_R = \sigma_Z = 0.45, \sigma_\phi = \frac{\pi}{16}$

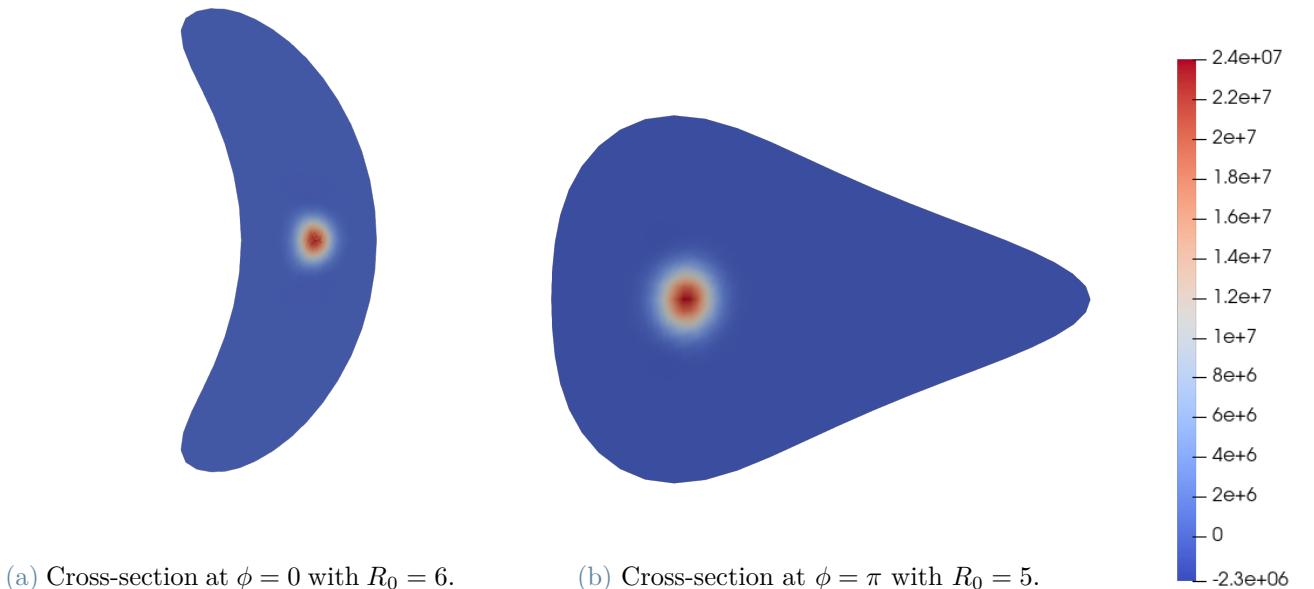


Figure 11: Projection of a Gaussian distribution onto the W7-X geometry. The significant deformation of the poloidal cross-section between the toroidal angles $\phi = 0$ and $\phi = \pi$ showcases the capability of the coupled projection scheme on a highly shaped, realistic geometry.

3.4. Convergence Tests

While the qualitative analysis shows promising results, it is vital to present quantitative studies as evidence of the scheme's accuracy. For this, a range of convergence tests has been performed by projecting some known analytical functions onto the W7-X grid. The idea is to measure the change of error as we increase the number of toroidal harmonics used to compute the solution. With the uncoupled projection, no convergence is expected since the geometrical coupling of harmonics is neglected. In contrast, for the coupled scheme, we expect an exponential, spectral [8], convergence.

We choose the L^2 norm between the original analytical function and the numerical solution from the projection on the FEM grid. By increasing the resolution of the solution, through the increase in the number of toroidal harmonics, we expect this error to decrease faster than the uncoupled counterpart. These tests have been performed on three different functions and are reported below:

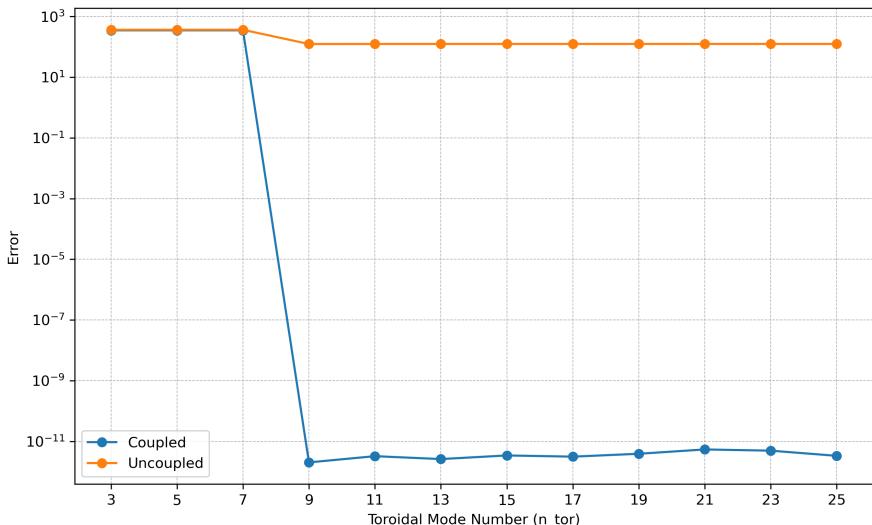


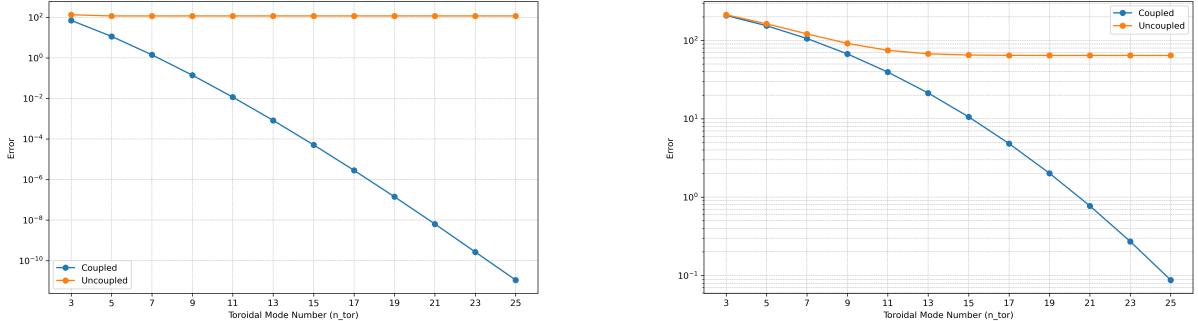
Figure 12: Performance of the coupled vs. uncoupled schemes for the function $\cos^2(2\phi)$. The uncoupled system fails to converge, while the coupled system shows a sharp convergence. The error decays precisely at $n_{\text{tor}} = 9$, which corresponds to a maximum mode number of $M = 4$ in our basis set ($n_{\text{tor}} = 2M + 1$), capturing the dominant harmonic in the function's analytical form. The remaining error at higher resolution is close to the machine epsilon for double-precision floating points.

The first function under consideration is a $\cos^2(2\phi)$, whose Fourier expansion has a periodicity of 4 ($\frac{1}{2} + \frac{1}{2}\cos(4x\phi)$). The behavior is perfectly in line with the theoretical assumptions: the sharp increase in accuracy is explained by the introduction of the harmonic with the proper periodicity into the spectral decomposition (Fig. 12). This pathological case highlights how the uncoupled system is fundamentally unable to approximate the function to a meaningful level of accuracy.

In order to validate the method against more realistic distribution scenarios, tests were performed using Gaussian functions (Fig. 13). The first scenario presents the perfect candidate for the convergence study. By using a periodic and smooth (C^∞) function such as $e^{\cos(\phi)}$, we can appreciate the spectral convergence of the method. The second case, a standard Gaussian on the toroidal direction of equation: $\exp\left(-\frac{(\phi-\pi)^2}{2(0.3)^2}\right)$, demonstrates how the uncoupled system can almost approximate the results from the coupled scenario for a small number of harmonics, usually this is not the case for production cases as the complex physics needs a high number of harmonics to be properly represented.

3.5. FFT acceleration of the matrix construction

As described in sec. 2.3.3, the integral on the toroidal direction can be efficiently evaluated by means of the FFT algorithm. This strategy was already employed in the construction of the matrix in the fluid time stepping, but since the toroidal dependency is missing in the tokamak case, the strategy had to be implemented from scratch for the projection system.



(a) Convergence for a periodized Gaussian function. The function is smooth and periodic.

(b) Convergence for a non-periodic Gaussian function. The convergence rate is slower, and the uncoupled approximation is good enough for a small n_{tor} number.

Figure 13: Convergence tests for two Gaussian functions projected on the W7-X grid.

The general strategy implemented was to collect all the toroidally independent parts of the integral together, and then, depending on whether the user wants to run the code with or without the FFT, either add the direct integration contribution or compute the coefficients via FFT. For the FFT computation, we leveraged the FFTW library [14].

The actual system to be solved for is slightly different from the one presented earlier. In order to get a smoother projection of the kinetic quantities, it is sometimes useful to use some filters applied to the derivatives of the solution. For these terms, a separate FFT needs to be performed; thus, two different sets of FFTs are performed separately and later reassembled together in one single matrix per element.

An excellent way to demonstrate the effectiveness of this implementation is to compare the performance with and without the FFT acceleration. Fig. 14 shows the average wall time required to build the projection matrix for both the direct integration (non-FFT) and the FFT-based methods alongside the speedup, plotted against the number of toroidal harmonics used (n_{tor}).

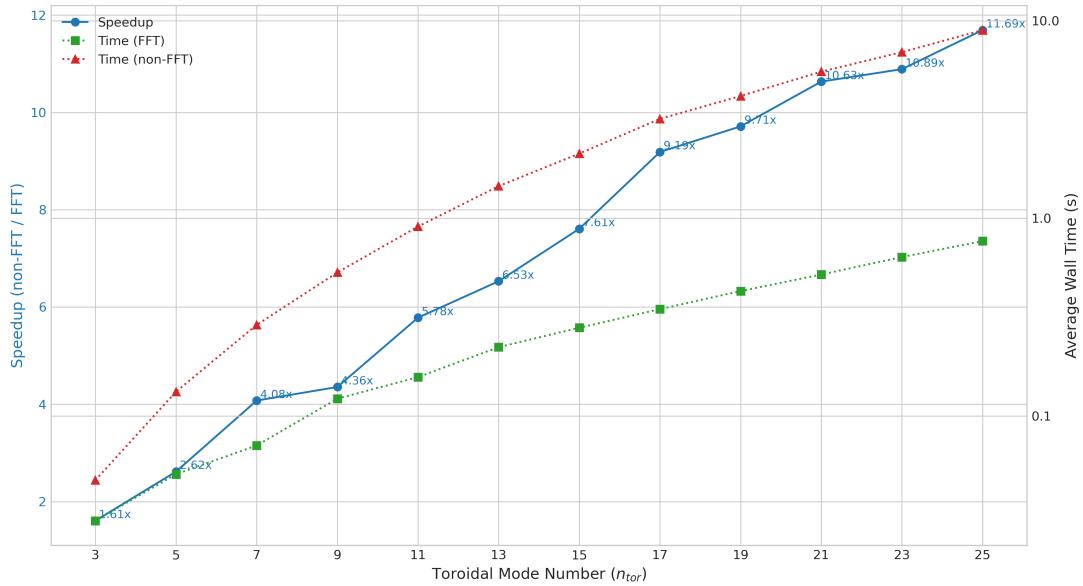


Figure 14: Comparison of wall time for the matrix assembly with (FFT) and without (non-FFT) considering $n_{plane} = 2n_{tor}$ the Fast Fourier Transform acceleration, as a function of the number of toroidal harmonics (n_{tor}). The speedup, calculated as the ratio of non-FFT time to FFT time, demonstrates a significant performance gain, aligned with the theoretical linear speedup with respect to n_{tor} .

As can be noted in Fig. 14, while the number of toroidal harmonics increases, the computational cost of the

direct integration method (represented by red triangles) increases quickly. The wall time for the FFT-based method (green squares), on the other hand, shows a considerably softer increase. Performance is significantly improved as a result of this scaling difference. The blue line on the plot represents the speedup, which is the ratio of the non-FFT time to the FFT time. With n_{tor} , it grows almost linearly, and for $n_{tor} = 25$, it achieves a notable speedup.

3.6. The RTree

A crucial missing step for a complete working simulation lies in the association of each particle with the corresponding element. This step is needed since each particle lives in the real space $RZ\phi$, but the finite element formulation requires an element-wise location of the particles to project the relevant quantities.

This problem can be reduced to a classical spatial indexing problem, for which many solutions have been developed over the years, depending on the specific application requirements.

For the problem at stake, since the mesh naturally partitions the space into regions, a region proposal algorithm is required. For these reasons, the RTree [16] algorithm and data structure were the most natural choice and are already used in the 2D tokamak applications of the code.

The underlying idea being that the bounding boxes in three dimensions of each element must cover the entire element, this is achieved by finding the minimum and maximum points of the element along two different poloidal section located at ϕ_1, ϕ_2 and defining the bounding box as $([R_{min}, R_{max}], [Z_{min}, Z_{max}], [\phi_1, \phi_2])$.

To find the 2D minimum bounding box at a given poloidal section, we can look for the points at which the derivative is zero along each of the four element sides, as briefly explained in sec. 2.2.7.

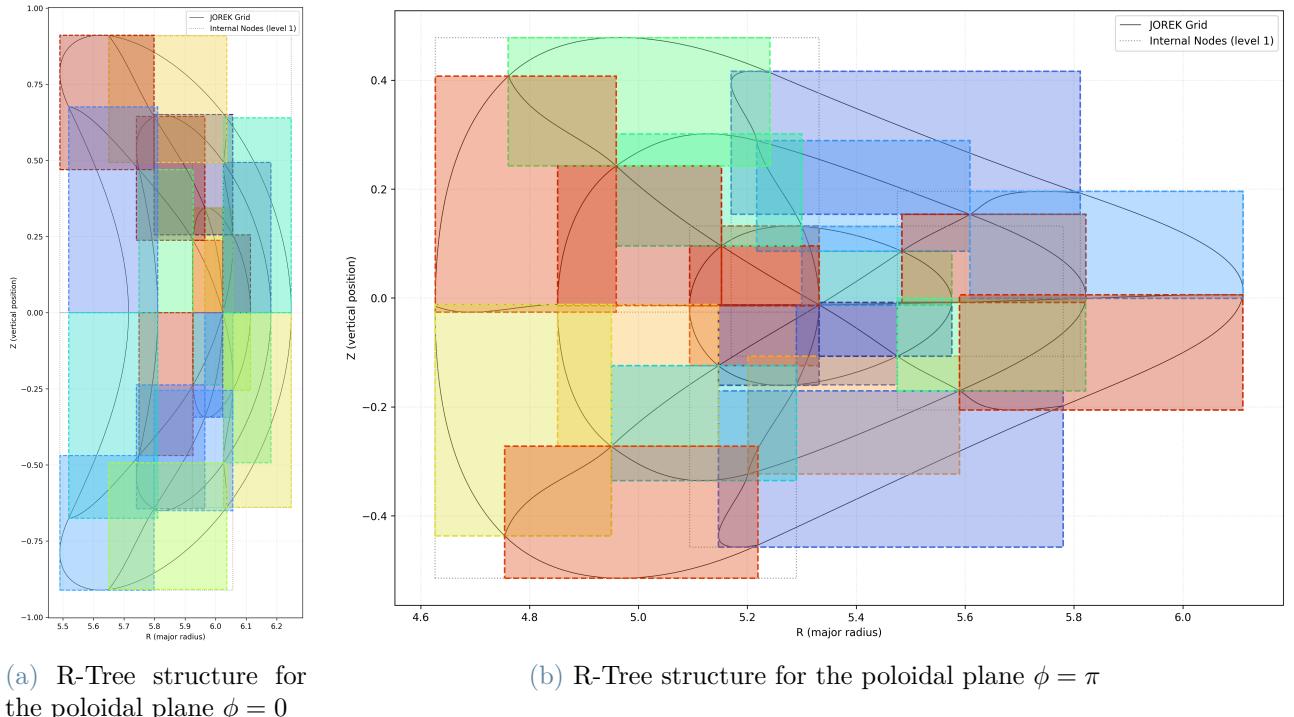


Figure 15: Visualization of the 2D R-Tree spatial indexes for the W7-X grid at two different toroidal angles. Each rectangle represents the 2D bounding box of a single mesh element at that specific poloidal cross-section. The significant change in the bounding box layout between (a) and (b) illustrates the strong geometric variation along the toroidal direction.

With the minimum of the element in the poloidal section (Fig. 15), we chose to consider the width of the bounding box in the toroidal direction as the distance between two different poloidal sections, thus creating our three-dimensional bounding box (Fig. 16).

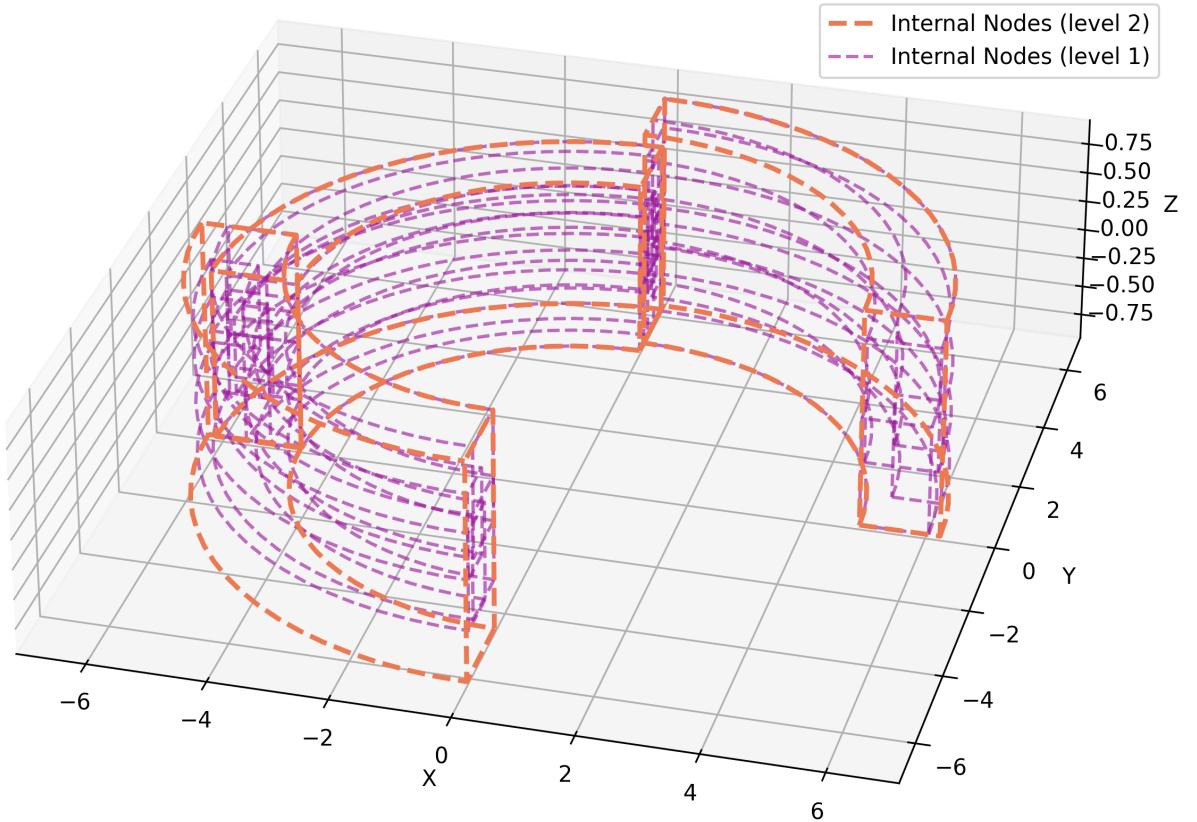


Figure 16: The 3D R-Tree data structure used for efficient particle lookup in the W7-X stellarator geometry. The image shows the 3D bounding boxes that form the tree’s nodes, creating a hierarchical map of the simulation domain. This spatial partitioning is a critical component for associating particles with their host finite elements in the non-axisymmetric grid.

The introduction of the third dimension on both the RTree algorithm and the local coordinate search introduces a significant overhead. This overhead should still be bounded by a factor proportional to the number of particles, for the association step between the particles and elements, and a constant, for the element creation.

As we can see in Fig. 17, we have a worst-case scenario of 80x speed-down for the tree creation. This is a favorable result, as a naive slicing of the toroidal direction into $n_{planes} = 128$ might suggest a 128x slowdown. The improved performance is due to the R-Tree’s hierarchical structure, which efficiently groups elements and prunes the search space, avoiding a linear scaling with the number of poloidal planes.

As per the particles association itself, adding a third dimension was bound to be computationally expensive; on average, we measure a 3x speed down, for which a small amount is attributable to the imperfect discretizations of the toroidal direction requiring, on average, 1.65x more elements to lookup.

Performance & Metrics: Coupled vs. Uncoupled

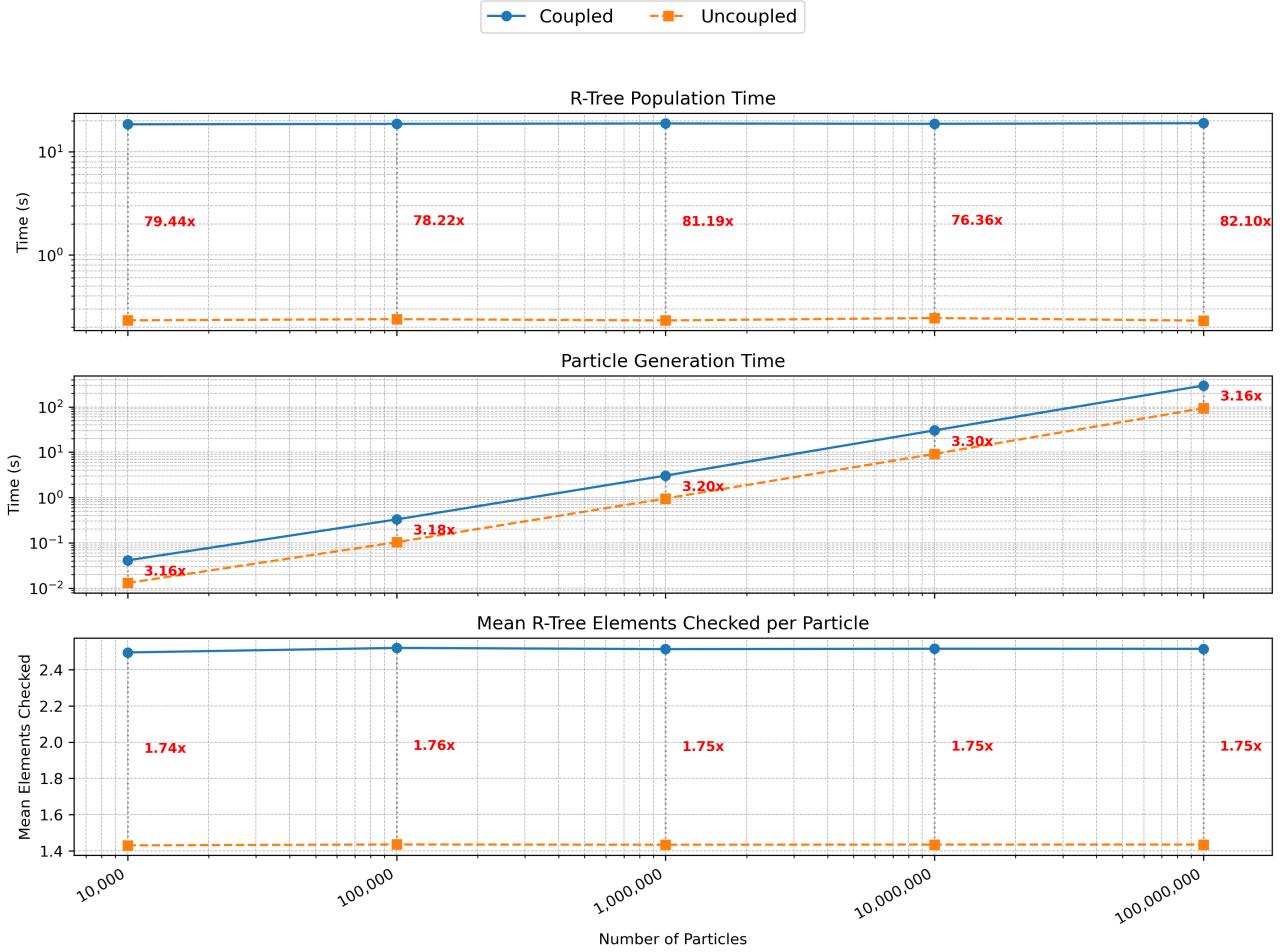


Figure 17: Performance comparison of the R-Tree implementation for element creation and particle association. The plot highlights the computational overhead introduced by the third dimension.

4. Conclusion

The development of strong, predictive simulation capabilities is essential to the search for commercially viable fusion energy, particularly for the geometrically challenging stellarator concept. A fundamental challenge in this field is the precise application of a hybrid fluid-kinetic model in a non-axisymmetric framework. Decoupled projection schemes used in simpler tokamak simulations are physically incorrect for stellarators because the toroidal Fourier harmonics are coupled by the intrinsic three-dimensional nature of the magnetic field.

The main contribution of this work is the design, implementation, and validation of a novel, globally coupled particle-in-cell projection scheme within the JOREK finite element code. This technique addresses the core challenge by building and solving a single, cohesive linear system over the entire 3D, non-axisymmetric mesh. By self-consistently accounting for the geometric coupling between all toroidal harmonics, the system ensures a high-fidelity transfer of information from the kinetic particle population to the fluid grid. An effective Fast Fourier Transform (FFT) based approach was used to make this computationally demanding global matrix construction possible. Qualitative projections onto realistic Wendelstein 7-X geometries demonstrated the scheme's ability to capture complex shaping. More significantly, quantitative convergence tests conclusively demonstrated the method's superiority: the coupled scheme achieved the theoretically expected spectral convergence for smooth analytical functions, whereas the uncoupled approach failed to converge, highlighting the necessity of this global treatment.

As a second significant contribution, an effective particle localization algorithm was implemented to address the prohibitive computational cost of a global scheme. We developed a scalable solution for associating millions of particles with their host finite elements by creating a three-dimensional R-Tree spatial index specific to the JOREK mesh structure. Performance analysis verified that this algorithm tackles the 3D search's complexity with a reasonable overhead, keeping the hybrid model tractable for large-scale, high-performance computing.

In addition to providing a solid and verified framework, this work opens up a number of fascinating directions for further investigation. In order to study how kinetic effects drive or dampen MHD instabilities, the most immediate next step is to enable the feedback of the kinetic particles on the fluid. A distributed memory model with MPI could be used to parallelize the assembly of the global matrix and right-hand-side vector, further improving scalability.

Although they come with some trade-offs, other possible improvements have been taken into consideration. One of them might be to find the true extrema of each element's geometry; for example, a Newton iteration could be used to determine the bounding boxes of the R-Tree with greater precision; however, this is probably too computationally costly for a marginal gain. Likewise, iterative solvers [38] are suboptimal to solve the projection system, even though they are frequently used for large linear systems. The use of direct solver libraries [1, 22, 36] based on a one-time matrix factorization is much more efficient because the projection system is solved repeatedly with the same matrix but a changing right-hand side.

Together, these contributions eliminate a significant obstacle to running high-fidelity, non-linear hybrid simulations of stellarator plasmas. This validated framework makes a previously unattainable range of predictive studies possible, enabling the community to better understand, optimize, and design the next generation of fusion energy devices and advancing our progress toward clean, sustainable energy.

References

- [1] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.
- [2] R. Aymar. Iter overview. *Fusion Engineering and Design*, 36(1):9–21, 1997.
- [3] H. Bergström, S.-J. Liu, V. Bandaru, M. Hoelzl, JOREK Team, and JET Contributors. Introduction of a 3d global non-linear full-f particle-in-cell model for runaway electrons in jorek. *Plasma Physics and Controlled Fusion*, 67(3):035004, feb 2025.
- [4] C. K. Birdsall and A. B. Langdon. *Plasma physics via computer simulation*. Taylor and Francis, New York, spl. indian ed. edition, 2005.
- [5] T. Bogaarts, M. Hoelzl, G. T. A. Huijsmans, and X. Wang. Development and application of a hybrid mhd-kinetic model in jorek. *Physics of Plasmas*, 29:122501, 12 2022.
- [6] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, Mineola, NY, second edition, 2001.
- [7] C. Sovinec and B. Cornille. Initial Results of NIMSTELL, the Stellarator Variant of NIMROD. In *APS Division of Plasma Physics Meeting Abstracts*, volume 2021 of *APS Meeting Abstracts*, page PP11.092, October 2021.
- [8] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Scientific Computation. Springer, Berlin, Heidelberg, 2006.
- [9] J. Cooley and J. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
- [10] O. Czarny and G. T. A. Huijsmans. Bézier surfaces and finite elements for mhd simulations. *Journal of Computational Physics*, 227(16):7423–7445, 2008.
- [11] D. C. van Vugt. *Nonlinear coupled MHD-kinetic particle simulations of heavy impurities in tokamak plasmas*. Phd thesis 1 (research tu/e / graduation tu/e), Applied Physics and Science Education, July 2019. Proefschrift.
- [12] G. L. Delzanno and E. Camporeale. On particle movers in cylindrical geometry for particle-in-cell simulations. *Journal of Computational Physics*, 253:259–277, 2013.
- [13] N. M. Ferraro and S. C. Jardin. Calculations of two-fluid magnetohydrodynamic axisymmetric steady-states. *Journal of Computational Physics*, 228(20):7742–7770, 2009.
- [14] M. Frigo and S. G. Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

- [15] A. G. Goodman, P. Xanthopoulos, G. G. Plunk, H. Smith, C. Nührenberg, C. D. Beidler, S. A. Henneberg, G. Roberg-Clark, M. Drevlak, and P. Helander. Quasi-isodynamic stellarators with low turbulence as fusion reactor candidates. *PRX Energy*, 3:023010, Jun 2024.
- [16] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, page 47–57, New York, NY, USA, 1984. Association for Computing Machinery.
- [17] P. Helander, C. D. Beidler, T. M. Bird, M. Drevlak, Y. Feng, R. Hatzky, F. Jenko, R. Kleiber, J. H. E. Proll, Yu. Turkin, and P. Xanthopoulos. Stellarator and tokamak plasmas: a comparison. *Plasma Physics and Controlled Fusion*, 54(12):124009, nov 2012.
- [18] P. Helander and D. J. Sigmar. *Collisional Transport in Magnetized Plasmas*. Cambridge University Press, 2002.
- [19] M. Hoelzl, G. T. A. Huijsmans, S. J. P. Pamela, M. Bécoulet, E. Nardon, F. J. Artola, B. Nkonga, C. V. Atanasiu, V. Bandaru, A. Bhole, D. Bonfiglio, A. Cathey, O. Czarny, A. Dvornova, T. Fehér, A. Fil, E. Franck, S. Futatani, M. Gruca, H. Guillard, J. W. Haverkort, I. Holod, D. Hu, S. K. Kim, S. Q. Korving, L. Kos, I. Krebs, L. Kripner, G. Latu, F. Liu, P. Merkel, D. Meshcheriakov, V. Mitterauer, S. Mochalskyy, J. A. Morales, R. Nies, N. Nikulsin, F. Orain, J. Pratt, R. Ramasamy, P. Ramet, C. Reux, K. Särkimäki, N. Schwarz, P. S. Verma, S. F. Smith, C. Sommariva, E. Strumberger, D. C. van Vugt, M. Verbeek, E. Westerhof, F. Wieschollek, and J. Zielinski. The jorek non-linear extended mhd code and applications to large-scale instabilities and their control in magnetically confined fusion plasmas. *Nuclear Fusion*, 61(6):065001, May 2021.
- [20] I. Holod, M. Hoelzl, P. S. Verma, G. T. A. Huijsmans, R. Nies, and JOREK Team. Enhanced preconditioner for jorek mhd solver. *Plasma Physics and Controlled Fusion*, 63(11):114002, sep 2021.
- [21] G. T. A. Huijsmans. Implementation of an iterative solver in the non-linear mhd code jorek. Technical Report ANR-CIS-2006-001-02, Association Euratom-CEA, CEA/DSM/DRFC, Centre de Cadarache, 13108 St. Paul lez Durance, France, 2008.
- [22] P. Hénon, P. Ramet, and J. Roman. Pastix: A parallel direct solver for sparse spd matrices based on efficient static scheduling and memory managment. 01 2001.
- [23] L.-M. Imbert-Gerard, E. J. Paul, and A. M. Wright. An introduction to stellarators: From magnetic fields to symmetries and optimization, 2020.
- [24] S. C. Jardin. Review of implicit methods for the magnetohydrodynamic description of magnetically confined plasmas. *Journal of Computational Physics*, 231(3):822–838, 2012. Special Issue: Computational Plasma Physics.
- [25] J. Killeen, G. D. Kerbel, M. G. McCoy, and A. A. Mirin. *Introduction*, pages 1–7. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.
- [26] T. Klinger, A. Alonso, S. Bozhenkov, R. Burhenn, A. Dinklage, G. Fuchert, J. Geiger, O. Grulke, A. Langenberg, M. Hirsch, G. Kocsis, J. Knauer, A. Krämer-Flecken, H. Laqua, S. Lazerson, M. Landreman, H. Maaßberg, S. Marsen, M. Otte, N. Pablant, E. Pasch, K. Rahbarnia, T. Stange, T. Szepesi, H. Thomassen, P. Traverso, J. L. Velasco, T. Wauters, G. Weir, T. Windisch, and The Wendelstein 7-X Team. Performance and properties of the first plasmas of wendelstein 7-x. *Plasma Physics and Controlled Fusion*, 59(1):014018, oct 2016.
- [27] S. Q. Korving. *MHD Simulations of Neutral and Impurity Transport in ELM-controlled Plasmas: with application to ITER*. Phd thesis 1 (research tu/e / graduation tu/e), Applied Physics and Science Education, June 2024. Proefschrift.
- [28] C. Liu, S. C. Jardin, H. Qin, J. Xiao, N. M. Ferraro, and J. Breslau. Hybrid simulation of energetic particles interacting with magnetohydrodynamics using a slow manifold algorithm and gpu acceleration. *Computer Physics Communications*, 275:108313, June 2022.
- [29] D. Liu. GC¹ continuity conditions between two adjacent rational Bézier surface patches. *Computer Aided Geometric Design*, 7(1-4):151–163, 1990.
- [30] Max Planck Institute for Plasma Physics. Wendelstein 7-a. <https://www.ipp.mpg.de/3951949/wendelstein7a>, 2025. Accessed: 2025-07-14.

- [31] N. Nikulsin, M. Hoelzl, A. Zocco, K. Lackner, and S. Günter. A three-dimensional reduced mhd model consistent with full mhd. *Physics of Plasmas*, 26(10), October 2019.
- [32] N. Nikulsin, M. Hoelzl, A. Zocco, K. Lackner, and S. Günter. Testing of the new jorek stellarator-capable model in the tokamak limit. *Journal of Plasma Physics*, 87(3), May 2021.
- [33] N. Nikulsin, R. Ramasamy, M. Hoelzl, F. Hindenlang, E. Strumberger, K. Lackner, S. Günter, and JOREK Team. Jorek3d: An extension of the jorek nonlinear mhd code to stellarators. *Physics of Plasmas*, 29(6):063901, 06 2022.
- [34] S. J. P. Pamela, G. T. A. Huijsmans, and M. Hoelzl. A generalised formulation of g-continuous bezier elements applied to non-linear mhd simulations. *Journal of Computational Physics*, 464:111101, 2022.
- [35] W. Park, S. Parker, H. Biglari, M. Chance, L. Chen, C. Z. Cheng, T. S. Hahm, W. W. Lee, R. Kulsrud, D. Monticello, L. Sugiyama, and R. White. Three-dimensional hybrid gyrokinetic-magnetohydrodynamics simulation. *Physics of Fluids B: Plasma Physics*, 4(7):2033–2037, 07 1992.
- [36] F.-H. Rouet, X. S. Li, P. Ghysels, and A. Napov. A distributed-memory package for dense hierarchically semi-separable matrix computations using randomization, 2015.
- [37] P. Rác. Gpu-accelerated stiffness matrix construction for finite element mhd code jorek. Master's thesis, Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland, 2024. Supervisors: M. Multerer; Supporting Supervisors: M. Hoelzl, I. Holod.
- [38] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003.
- [39] C. R. Sovinec, A. H. Glasser, T. A. Gianakon, D. C. Barnes, R. A. Nebel, S. E. Kruger, D. D. Schnack, S. J. Plimpton, A. Tarditi, and M. S. Chu. Nonlinear magnetohydrodynamics simulation using high-order finite elements. *Journal of Computational Physics*, 195(1):355–386, 2004.
- [40] W. M. Tang. Scientific and computational challenges of the fusion simulation project (fsp). *Journal of Physics: Conference Series*, 125(1):012047, jul 2008.
- [41] Y. Todo and T. Sato. Linear and nonlinear particle-magnetohydrodynamic simulations of the toroidal alfvén eigenmode. *Physics of Plasmas*, 5(5):1321–1327, 05 1998.
- [42] T. Ueshiba and G. Roth. Generating smooth surfaces with bicubic splines over triangular meshes: toward automatic model building from unorganized 3d points. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)*, pages 302–311, 1999.
- [43] X. Wang, S. Briguglio, L. Chen, C. Di Troia, G. Fogaccia, G. Vlad, and F. Zonca. An extended hybrid magnetohydrodynamics gyrokinetic model for numerical simulation of shear alfvén waves in burning plasmas. *Physics of Plasmas*, 18(5), May 2011.
- [44] J. Wesson. *Tokamaks*. Oxford University Press, 4 edition, 2011.
- [45] R&D World. An overview of the late 2024 supercomputing landscape in 6 charts. <https://www.rdworldonline.com/an-overview-of-the-late-2024-supercomputing-landscape-in-6-charts/>, 2025. Accessed: 2025-08-28.
- [46] Y. Xu. A general comparison between tokamak and stellarator plasmas. *Matter and Radiation at Extremes*, 1(4):192–200, 2016.
- [47] Y. Zhou, N. M. Ferraro, S. C. Jardin, and H. R. Strauss. Approach to nonlinear magnetohydrodynamic simulations in stellarator geometry. *Nuclear Fusion*, 61(8):086015, July 2021.
- [48] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*, page iii. Butterworth-Heinemann, Oxford, January 2013.

Algoritmi e ottimizzazioni per la simulazione globale e non-lineare di stellarator tramite modelli ibridi fluido-cinetici con il metodo degli elementi finiti

Il progresso della ricerca sull'energia da fusione nucleare richiede lo sviluppo di modelli predittivi per i plasmi nei reattori stellarator. Tuttavia, la modellizzazione di questi sistemi presenta alcune sfide computazionali. Tra queste, una delle difficoltà principali consiste nel simulare con precisione la dinamica delle particelle all'interno del plasma, un compito che richiede modelli ibridi, a metà tra fluidodinamica e cinetica. A differenza dei tokamak, la geometria non assisimmetrica degli stellarator induce un accoppiamento intrinseco tra i modi toroidali, che complicano ulteriormente la simulazione.

Partendo da tali premesse, il presente lavoro propone, dunque, un nuovo schema di proiezione, integrato nel codice a elementi finiti JOREK. Tale approccio consente di trasferire l'informazione cinetica sulla mesh in modo consistente e fisicamente fondato, gestendo la complessità della mesh 3D. Ciò avviene costruendo e risolvendo un unico sistema lineare che include simultaneamente tutte le armoniche toroidali.

Al fine di gestire in modo efficiente e gestibile la localizzazione di milioni di particelle, è stato sviluppato un algoritmo di indirizzamento spaziale attraverso un R-Tree tridimensionale, il quale garantisce la fattibilità computazionale del metodo su larga scala. L'affidabilità del codice sviluppato è stata rigorosamente verificata su geometrie realistiche del reattore stellarator Wendelstein 7-X.

I test di convergenza hanno confermato il raggiungimento della convergenza spettrale prevista dalla teoria, evidenziando, inoltre, il contrasto con le scarse prestazioni degli approcci non accoppiati. Di conseguenza, si è evidenziato come lo sviluppo di questo strumento computazionale fornisca una risorsa essenziale per l'analisi predittiva e l'ottimizzazione di futuri progetti cinetici su stellarator.

Parole chiave: Simulazioni di stellarator, Modelli ibridi fluido-cinetici, Metodo degli Elementi Finiti (FEM), Metodo Particle-in-Cell (PiC), JOREK, Calcolo ad Alte Prestazioni (HPC)