A 3D Unstructured Mesh based Particle Tracking Code for Impurity Transport Simulation in Fusion Tokamaks

Dhyanjyoti D. Nath[a], Vignesh V. Srinivasaragavan[a], Timothy R. Younkin[b], Gerrett Diamond[a], Cameron W. Smith[a], Alyssa Hayes[c], Mark S. Shephard[a], Onkar Sahni[a]

[a]*Scientific Computation Research Center, Rensselaer Polytechnic Institute, 110 8th St., Troy, 12180, NY, USA*
[b]*Oak Ridge National Laboratory, 1 Bethel Valley Rd., Oak Ridge, 37830, TN, USA*
[c]*Department of Nuclear Engineering, University of Tennessee, Knoxville, 37996, TN, USA*

Abstract

A fully 3D unstructured mesh based global impurity transport code, GITRm is presented in this paper. It is a high-performance Monte Carlo particle (neutral atom and ion) tracking code, based on the trace approximation, to simulate the erosion, ionization, migration, and redistribution of material from plasma-facing components in magnetically confined fusion devices. It is designed to target complex geometries including non-axisymmetric local features such as bumpers, probes, tile gaps etc., and uses strongly graded and anisotropic elements to accurately represent the plasma fields. GITRm is built on the PUMIPic infrastructure [1], executes using distributed meshes and is performant on GPU accelerated computer systems. Three example cases, including a weak scaling study with about 1.5 billion particles on up to 144 GPUs, are used to demonstrate the utility of GITRm.

Keywords: Particle-In-Cell, Unstructured Mesh, GPU

1. Introduction

To address the wide range of physics that must be understood to produce a burning fusion plasma from which energy can be harvested, a wide range of simulation codes, of various levels of modeling fidelity, have been, and continue to be, developed and applied. To meet the aggressive plan to

bring a prototype fusion power plant online by 2040 [2], a number of coupled simulation codes that can provide the needed levels of modeling fidelity for the full set of relevant physical behaviors are needed. A number of the codes with the potential of providing the required levels of physics modeling fidelity are based on particle-in-cell (PIC) methods. Ongoing advances in the development of exascale computing systems, physics models and numerical methods hold the promise of supporting PIC simulation workflows capable of providing the high-fidelity predictions needed.

Considering the complexity of the up-coming exascale computing systems, computational methods that execute on uniform structures such as mapped structured grids that follow physics are the most highly effective for addressing specific simulation needs. However, many of the design and operation questions for fusion reactors must carefully account for complex geometric components (e.g., divertor cassette assemblies, probe ports, etc.) and/or localize behaviors (e.g., pellet injection and ablation). Simulation tools that employ structured grid methods are not well suited to simulate geometrically complex local features within overall device simulations.

On the other hand, unstructured mesh generators have evolved to the point that strongly graded, anisotropic meshes can be easily generated that can take full account of geometric detail and/or localized behaviors of interest. An increasing number of PIC codes for the modeling of plasma physics in fusion reactors, [3, 4, 5, 6, 7], linear accelerators [8] and other systems are currently, or planning to, take advantage of unstructured mesh methods. The PUMIPic infrastructure [1] was developed to effectively support unstructured mesh based PIC codes.

Particle impurity simulation tools either consider the transport of impurities using a 2D fluid model or employ a kinetic impurity Monte Carlo approach. Numerical codes developed using a 2D fluid approximation include SOLPS [9]. SOLPS has been extended to give a full fluid treatment for tungsten impurities. However, this places a large demand on runtime and memory, even on a 2D domain. Further, the kinetic aspects of impurity transport are generally absent in these fluid models. Kinetic impurity tracking codes, like DIVIMP [10] or DORIS [11], were originally developed to study low-Z impurities [12], and thus, use a guiding center approximation. This enables simulation of larger domains with larger time steps, but does not fully resolve the gyro-orbits that is important for accurately capturing the particle trajectories in the sheath and magnetic pre-sheath regions. In contrast, codes such as IMPGYRO [12], ERO [13], ERO2.0 [14] and GITR

[15] are 3D Monte Carlo codes, and fully resolve the gyro-orbits. However, they typically employ uniform structured grid in the volume and are limited to cases that are 3D axisymmetric or localized. ERO has mostly focused on localized wall components whereas IMPGYRO currently does not include the magnetic pre-sheath. ERO2.0 [14] allows increasing the simulation volume in order to cover the entire plasma edge of a fusion device. On the other hand, GITR targets the entire tokamak domains and includes magnetic pre-sheath, but is currently limited to uniform structured grid in the volume. In summary, a fully unstructured 3D mesh based approach is lacking.

This paper presents a fully unstructured 3D mesh based particle impurity tracking code, GITRm. It builds on the PUMIPic [1] and Omega_h [16, 17] software libraries. It currently uses a mesh-centric data structure (in contrast to a particle-centric one), e.g., uses a partitioned mesh, and therefore, is different from other approaches/codes discussed above including GITR. It is designed to operate on high-performance massively parallel computer systems (i.e., accelerator/GPU-enabled and distributed-memory systems). It is capable of fully resolving the 3D impurity particle gyro-motion in realistic tokamak geometries. It is designed to target complex geometries including non-axisymmetric local features such as bumpers, probes, tile gaps etc. Additionally, it uses strongly graded and anisotropic unstructured elements to efficiently and accurately represent the plasma fields, especially in a 3D/volumetric fashion (this aspect is also an important difference from approaches/codes discussed above, e.g., in contrast to using a uniform structured grid to represent the plasma fields as currently done in GITR). It currently uses a fully 3D unstructured tetrahedral mesh. It also provides flexibility in the use of fields from various sources (including various sheath models) and in-memory coupling with other codes to address the disparity in spatial and temporal scales of various physical processes that is encountered in plasma material interaction studies. Although only static plasma conditions with one-way coupling are applied in this work, the procedures developed in this work for impurity transport are capable of handling transient plasma conditions and two-way coupling will be studied in the future.

Section 2 provides an overview of the impurity transport physics modeling capabilities included in the new code GITRm. Section 3 provides an overview of PUMIPic that provides the core PIC infrastructure for GITRm. Section 4 describes the PUMIPic based parallel algorithms used to implement the impurity transport operations in GITRm. Section 5 includes examples demonstrating the capabilities of GITRm and provides performance results.

Section 6 provides the closing remarks.

2. Fusion Plasma Impurity Transport Modeling

Fundamental to the operation of a fusion tokamak device is the containment of a high density core plasma that, in the case of ITER, is targeted to operate at 150 million degrees celsius. Outside the core plasma are plasma-facing components (PFC) that must provide the necessary heat exhaust capacity and also provide adequate protection for the tokamak components outside the core. To maintain the required level of core plasma purity, it is critical to limit the impurities that are sputtered into the core plasma from plasma facing components. Simulation of the processes associated with impurity transport requires the application of a set of coupled physics models and associated simulation tools. Currently, the research community is focused on increasing the fidelity of such tools and to couple them to support the evaluation of PMI options and the design of PFCs [18].

This work focuses on modeling and simulation of impurity transport in fusion plasma devices. Specifically, to capture the motion of impurity atoms that erode from PFCs into the plasma. This motion is dictated by the electric field, magnetic field, and impurity-plasma interactions. The steady-state Boltzmann transport equation is used in the trace approximation in that the eroded material concentrations remain low and therefore do not perturb the plasma state or undergo self (impurity-impurity) collisions. The governing partial differential equation for each impurity charge species is given as follows:

$$\mathbf{v}_z \cdot \nabla f_z + \frac{q}{m_z}\Big(\mathbf{E} + \mathbf{v}_z \times \mathbf{B}\Big) \cdot \frac{\partial f_z}{\partial \mathbf{v}_z} = C_{zb}(f_z, f_b) + S_1 + S_2 + S_3 \quad (1)$$

where the distribution function of the impurity species z, $f_z = f_z(\mathbf{r}, \mathbf{v}_z, q, t)$, is defined with respect to the position vector $\mathbf{r}$, velocity vector of the impurity particle $\mathbf{v}_z$, the charge state q, and time t. The number of impurity atoms of a given charge state is $N(q,t) = \iint f_z(\mathbf{r}, \mathbf{v}_z, q, t)\, d\mathbf{r}\, d\mathbf{v}_z$, the mass of the impurity is m_z and the charge of the impurity is q. $\mathbf{E}$ is the electric field and $\mathbf{B}$ is the magnetic field. The current solution procedure captures only the steady-state solution. For this reason, only temporally constant plasma profiles of electron temperature, ion temperature, electric field and magnetic field are used. Further, the collision operator, C_{zb}, which represents the

impurity species colliding with the background plasma species and defined in [19], assumes f_b to be Maxwellian and the effects from impurity species C_{zz} to be negligible. Currently, S_1 and S_2 are the source terms for surface interactions and S_3 is the source term for atomic physics. Specifically, S_1 represents the steady-state plasma surface erosion due to the the incoming background plasma flux and S_2 represents the resulting surface self erosion that arises when impurity material moving through the plasma strikes the material surface. S_3 represents the atomic physics based source that accounts for the change of impurity charge states.

To account for the transport of the impurities through the plasma, including interactions with the material surfaces, impurity migration patterns must be computed at the global or device scale (i.e., full tokamak) without the loss of short spatial scale effects. Currently, a kinetic simulation using the trace impurity approximation uses forces acting on particles to update the impurity particle position at discrete time steps [15]. In such an approach, ordinary differential equations in combination with Monte Carlo physics operators for stochastic processes are used to model the motion of particles as an approximation of the Boltzmann transport equation, i.e., Equation 1. It is composed of two physics components. The first accounts for surface sources/interactions, while the second addresses impurity transport in the plasma region. These are discussed below.

2.1. Surface Interactions

Material surface interactions play a critical role for impurities in magnetically confined fusion devices. In particular, the impurity material erodes from the plasma-facing surfaces/walls due to the incoming flux of the plasma species, ejects into the thin region of the scrape-off layer, and a fraction of it escapes into the core plasma region. The transport of impurity particles within the plasma region is discussed in the next subsection. The impurity material moving around in the plasma region also strikes or impacts the wall resulting in a surface event in the form of a deposition, reflection, or sputtering. The surface event is dictated by the impurity state at impact, i.e., energy and angle.

The erosion of impurities from the wall caused by plasma flux must be taken into account over the course of the simulation. However, in the current work, the erosion source due to the background plasma species is constant in time based on the trace approximation and allows setting the total amount of impurity erosion by the incoming plasma over the entire simulation at the

start. Thus, for incoming plasma-based erosion, impurity particles are initialized at the start of the simulation on the plasma-facing surfaces. The current numerical procedure for initializing the impurity particles to represent the eroded material is discussed in Section 4.3.

The surface response model is concerned with determining the surface interaction physics when an (incoming) impurity particle from the plasma hits the material surface. A material surface may only allow deposition or may also allow reflection or sputtering. If the material surface can sputter or reflect, then a detailed model for the surface response is required. Four physical outcomes are possible for such a surface response model: pure deposition, deposition and sputtering, reflection or reflection and sputtering. For any incoming particle, sputtering yield and reflection coefficient are used to determine the specific physical outcome (out of the four combinations) in a probabilistic sense. The sputtering yield and reflection coefficient depend on the energy and angle of the incoming particle (where the angle is defined between the surface normal and velocity vector of the particle at the point of impact). They can be described in terms of an analytic expression, a look-up/data table, a binary-collision model or a molecular dynamics simulation. Similarly, for the outgoing particles distributions for energy and angle (with the surface normal) can be described, while the in-plane angle (on the surface) follows a uniform distribution from 0 to 2π. At the surface level, the deposition or erosion is tracked based on the physical outcome for particles. For example, if a pure deposition happens then the surface mass is increased by the appropriate amount. Similarly, if a deposition and sputtering, or reflection and sputtering, event occurs then the net mass change is determined and updated for the surface.

In addition, the computational model used for surface response uses a particle weighting approach. This is done in order to maintain a constant number of simulated impurity particles such that different particles have a relatively different contribution or weight in the simulation. In this particle weighting approach, the surface response model modifies the computational weight of the outgoing particles. In such a computational model the particle weight is updated by a factor that depends on the sputtering yield and reflection coefficient. The computational model of the surface accounts for gross deposition and erosion (i.e., fluxes) based on the change in the computational weight of the particles due to the surface response/model. This weighting approach is described in more detail in [15].

In this work, the surface response model uses output data from a binary

collision approximation (BCA) model, e.g., from a BCA code F-TRIDYN [20]. This data involves reflection coefficients and sputtering yields, for example, when a tungsten atom/ion hits a tungsten surface. The BCA model provides the probability of reflection or sputtering as a function of the energy and angle of the incoming particles. Our ongoing work also includes on-the-fly computation of the surface response, e.g., based on a BCA code RustBCA [21], which will allow for transient plasma conditions, e.g., in case of an edge-localized mode (ELM), and in the future, two-way coupling when impurities ejected into the plasma can perturb the state of the plasma. The current numerical methodology used for the surface response of impacting impurity particles is discussed in Section 4.6.

2.2. Impurity Transport in Plasma

The motion of any impurity particle is based on the following equations that account for different physical processes including the Lorentz force, drag force, parallel and perpendicular diffusion and energy loss. Each of these physical processes is discussed below. The Lorentz force results from the left-hand side of Equation 1 and is defined as:

$$\mathbf{F}|_{Lorentz} = q(\mathbf{E} + \mathbf{v}_z \times \mathbf{B}) \tag{2}$$

In the current approach, a dirac delta function for velocity, $\delta(\mathbf{v}_z - \mathbf{v})$, represents a discrete particle velocity and can be used to take moments of the Boltzmann equation (i.e., Equation 1). The zeroth moment of the right-hand side collision operator is equal to zero, i.e., $\int C_{zb} d\mathbf{v}_z = 0$, because no mass is created or destroyed in this collision operator. The first moment, $\int C_{zb}\mathbf{v}_z d\mathbf{v}_z$, results in a drag force:

$$\mathbf{F}|_{Drag} = -\nu_s m_z \mathbf{v}_z \tag{3}$$

where ν_s is the slowing-down frequency or dynamical coefficient of drag.

The second moment, $\int C_{zb}\mathbf{v}_z\mathbf{v}_z d\mathbf{v}_z$, results in velocity diffusion in the parallel and perpendicular directions as well as in energy loss. For a short time scale or a short time step, when $\nu\Delta t << 1$ (where ν is a representative frequency), these can be defined as follows:

$$\frac{d}{dt}|\mathbf{v}_z - \bar{\mathbf{v}}_z|^2_{\parallel}\Big|_{\parallel\ diffusion} = \nu_{\parallel} v_z^2 \tag{4}$$

$$\frac{d}{dt}|\mathbf{v}_z - \bar{\mathbf{v}}_z|_\perp^2\Big|_{\perp\ diffusion} = \nu_\perp v_z^2 \tag{5}$$

$$\frac{d}{dt}\varepsilon\Big|_{Energy\ loss} = -\nu_\varepsilon \varepsilon \tag{6}$$

where $\nu_\perp$ is the perpendicular deflection frequency, $\nu_\parallel$ is the frequency for parallel velocity diffusion, and ν_ε is the energy loss frequency. These frequencies are related as: $\nu_\varepsilon = 2\nu_s - \nu_\perp - \nu_\parallel$. These frequencies are explicitly derived and listed in [19]. $\bar{\mathbf{v}}_z$ is the average particle velocity over a time step of size Δt, approximated by: $\mathbf{v}_{z0}(1-\nu_\varepsilon \Delta t/2)$, where $\mathbf{v}_{z0}$ is the particle velocity and $v_{z0} = |\mathbf{v}_{z0}|$ at this time step. The particle kinetic energy is: $\varepsilon = mv_z^2/2$ based on the particle speed v_z.

In the numerical procedure, $\mathbf{v}_{z0}$ is taken to be the updated velocity due to the Lorentz force. Taking contributions from Equations 3, 5, 4 and 6 in a relative frame results in a numerical model for the evolution of velocity as follows:

$$\begin{aligned}\mathbf{v}_z = v_{z0}(1-\nu_\varepsilon \Delta t/2)\big[\hat{\mathbf{e}}_1(1+\eta_1\sqrt{\nu_\parallel \Delta t})+ \\ |\eta_2|\sqrt{\nu_\perp \Delta t/2}(\hat{\mathbf{e}}_2 cos(2\pi\xi) + \hat{\mathbf{e}}_3 sin(2\pi\xi))\big] - \nu_s \Delta t \mathbf{v}_{z0}\end{aligned} \tag{7}$$

where $\hat{\mathbf{e}}_1$ is the parallel direction of particle travel (i.e., along $\mathbf{v}_{z0}$) and $\hat{\mathbf{e}}_2$ and $\hat{\mathbf{e}}_3$ are the perpendicular directions (i.e., in the local plane perpendicular to the velocity vector). The random variables η are sampled from a normal distribution with a mean of 0 and a standard deviation of 1. The random variable ξ follows a uniform distribution between 0 and 1 such that the resulting vector lies in the local plane perpendicular to the velocity vector.

In addition, an ad-hoc operator for anomalous cross-field diffusion is used:

$$\mathbf{r} = \mathbf{r}_0 + \sqrt{4D\tau_1}\hat{\mathbf{B}}_\perp \tag{8}$$

where $\mathbf{r_0}$ is the position vector of the particle after applying the Lorentz force, $\mathbf{r}$ is the position vector of the particle after applying the cross-field diffusion, D is the diffusion coefficient, τ_1 is a time scale and at any given step it is set to be time step size, Δt, and $\hat{\mathbf{B}}_\perp$ is a uniformly selected random unit vector that lies in the local plane perpendicular to the magnetic field. Thus, Equation 8 accounts for the anomalous cross-field diffusion in the perpendicular direction to the magnetic field by adding a stochastic displacement vector to the particle position.

The above equations are used to update the position and velocity of a particle at each time step.

The electric field, $\mathbf{E}$, in Equation 2 comprises the sheath and the pre-sheath electric field near the material surface, as well as any other bulk electric fields provided. The sheath electric field is a relatively strong electric field that forms near a plasma-facing material surface and is directed towards the surface. The sheath region can be modeled using the Chodura sheath based on various analytic models such as the Brooks model [22] and the Langmuir model [23]. For example, in the Brooks model it is composed of two components and the magnitude of the sheath electric field, $E = |\mathbf{E}_{sheath}|$, at a distance, d, from the wall is given as:

$$E = \phi_0 \frac{f_d}{2\lambda_D} e^{-d/2\lambda_D} + \phi_0 \frac{1 - f_d}{r_L} e^{-d/r_L} \tag{9}$$

where ϕ_0 is a potential and a function of the electron temperature T_e, f_d is a function of the angle between the surface normal and the magnetic field, λ_D is the Debye length and r_L is the Larmor radius. From Equation 9 it is seen that the sheath electric field is prominent near the wall, has a sharp gradient, and decreases exponentially as one moves away from the wall. For complex geometries, such as castellated walls or tiled wall with gaps, an analytic model is not suitable. In such cases, a sheath simulator can be employed, e.g., a sheath code hPIC [4]. Our ongoing work includes the use of a sheath simulator that will also account for transient plasma conditions. Note that the width of the sheath electric field in tokamaks like ITER can be of the order of a few thousands of Debye lengths and is only covered by tens of mesh elements/cells in the wall-normal direction within the edge plasma and impurity transport codes, i.e., element size is hundreds of Debye lengths near the wall. On the other hand, a sheath simulator uses a much finer mesh with an element size of Debye length near the wall. These aspects are important to account for in the numerical procedure. The numerical procedure used currently to account for the sheath electric field is discussed in Section 4.4.

As the particles move in the plasma region, their charge, q, evolves through ionization and recombination processes at the atomic level. To account for these atomic physics, a set of Monte Carlo operators is used. The coefficients for these operations are obtained from the Atomic Data and Analysis Structure (ADAS) database [24] which gives the Maxwellian-averaged reaction rates γ. In particular, the ionization and recombination coefficients are obtained from the ADF11 dataset, where the coefficients de-

pend on the electron temperature, T_e, and the electron density, n_e, and are calculated by collisional-radiative models. The mean time for a particular atomic interaction is defined as:

$$\tau_{atm} = \frac{1}{\gamma} \tag{10}$$

The probability of occurrence of an atomic interaction at any given time step is given as:

$$P = 1 - e^{-\frac{\Delta t}{\tau_{atm}}} \tag{11}$$

such that an atomic interaction is assumed to occur when this probability exceeds a random draw from a uniform distribution between 0 and 1.

Note that in the current work, the fields T_e and n_e for calculating the atomic physics are obtained from the edge plasma model. However the variation of these quantities in the sheath and pre-sheath regions can be considered for higher accuracy.

3. Parallel Unstructured Mesh Infrastructure for PIC Calculations

GITRm builds upon a parallel unstructured mesh infrastructure for PIC calculations, PUMIPic [1]. PIC methods are implemented as a time advancing procedure in which "particles" are tracked as they move through a domain, driven by a field that can be a function of the position of the particles. In the coupled case there are four steps carried out in each time advance [4, 5, 6, 25, 26, 27, 28]. Those steps are:

Field to Particle: The values of the current mesh-based fields that drive the particles are associated with each particle through an appropriate interpolation procedure.

Particle Push: The particles are moved to a new location as a function of the fields and time step.

Charge Deposition: The "charge" information associated with the particles is related to the domain definition such that the forcing function driving the field evolution is updated.

Field Solve: The equations, typically partial differential equations, governing the field are then solved using this updated forcing function.

Note that the last two steps are not needed under the current trace impurity approximation. The PUMIPic infrastructure does support these steps and thus, can be activated in extended versions of the code in the future.

The typical approach to the development of an unstructured mesh PIC code is for the particle data structure to be the core data structure in which the particles store a pointer to the current element in which it is contained. The mesh is stored in an independent data structure that is typically copied into the memory of each process. Although the mesh data is substantially smaller than the particle data, maintaining a copy of the mesh in each memory space does limit the scalability with respect to mesh size. PIC codes using particle centric data structures may also include an easily indexed spatial association of unstructured mesh elements to speedup the determination of the element containing each particle at any given time step.

PUMIPic [1] takes an alternative approach in which the core data structure is a distributed mesh with the particles tied to the mesh elements in which they currently reside. This approach provides an effective opportunity to distribute the mesh over the nodes of the compute system, thus supporting scalability with respect to the mesh, while maintaining ready access to the particles those element's fields will move in the next push step. The mesh data structure used in PUMIPic [16, 17] stores a complete mesh topology and has been designed to support the effective execution of unstructured mesh operations on distributed meshes on massively parallel computers employing GPU accelerators.

To store particles based on mesh elements, an additional structure is maintained that groups particles in memory based on the mesh element they are within. PUMIPic supports alternative effective data structures to support the storage of this information [1]. To maintain the particle structure as particles move through the domain, three operations are required. First, after every particle push, each particle must determine if it has moved to a new mesh element and if so which element that is. To achieve this an adjacency search is executed on each particle using ray tracing and barycentric coordinates based methods. After the particles determine their new element, the particle structure must be updated to reflect these changes. Since the unstructured mesh is distributed the particles must check if they need to be migrated to a new process based on the new element. The efficient implementation of this step is supported by the inclusion of buffer parts or elements and dynamic load balancing procedures in the PUMIPic infrastructure [1]. Once all the particles are migrated to the correct process, the particle struc-

ture is rebuilt in order to add or remove particles, and reorder particles based on their new mesh element and process.

A key component of PUMIPic is the manner in which the mesh is distributed, as a set of so-called PICparts, to the MPI processes employed on today's parallel computers. Since the Field-to-Particle and Charge Deposition steps in a PIC calculation will require communications in each time advance, PUMIPic deems it is satisfactory to move particles between PICparts during those steps, while the PICpart definition is such that during the Particle Push step all required information is local to the PICpart and no communications are required. This is accomplished by employing a buffer of mesh elements surrounding the elements on a PICpart that will contain particles to be pushed in the current push operation such that the particles do not move outside that buffer. Since after a Particle Push particles can enter the buffer and be close enough to the boundary of the PICpart that they may move off the PICpart in the next Particle Push, it becomes necessary to do the communication needed to move the particle onto a PICpart for which that element is sufficiently far from that PICpart's boundary. The communications required to move those particles can be coordinated and carried out in the Charge Deposition step which always requires communications.

Specific care is required in the definition of this buffer to ensure it does not produce large increases in memory used or effort required to maintain the mesh distribution information. As explained in reference [1], the definition of the PICparts begins by partitioning the mesh into a non-overlapping set of parts. Each of those parts defines the core of a PICpart. The left image of Figure 1 shows a 2D tokamak cross section with a very coarse mesh partitioned into 15 non-overlapping parts while the right image shows the PICpart defined for the core part labeled A. The remainder of the PICpart is the set of other parts surrounding the PICpart core such that there is sufficient buffer that any particle that is in an element in the core part at the start of a push operation will end-up in an element on that PICpart. When defined in this manner a substantial percentage of the particles that move to elements in surrounding parts are far enough from the PICpart boundary that they would not exit the PICpart on the next push. In these cases, particles are only migrated for the purpose of improving load balance for the next push operation. As discussed in Section 4, the fact that impurity particles are far from uniformly distributed over the domain, and they move through the domain during the simulation, the regular application of dynamic load balancing, as described in [1], is a core operation carried out in a GITRm

simulation.

Figure 1: Left: Two-dimensional unstructured mesh partitioned using multi-level graph partitioner. Right: PICpart generated for the core part A. Note that although part B does not share any boundary with part A, it is included in part A's PICpart because it is only few elements away from part A.

4. GITRm Implementation and Algorithms

The current simulation workflow for tokamak impurity transport, based on GITRm, requires the execution of the following steps:

1. Obtain the geometry definition and generate the desired graded anisotropic mesh with specific consideration of the locally varying physics.
2. Initialize the plasma-related electric, magnetic, ion and electron temperature and density fields over the unstructured mesh and use procedures to update these fields for coupled simulations. Currently, only one-way coupling with temporally constant fields is supported. Efforts are underway to support time dependent fields that will allow for transient plasma conditions, e.g., in case of an edge-localized mode (ELM).
3. Prescribe impurity sources due to incoming plasma flux.
4. Define sheath electric field model and determine mesh elements within sheath region, including distance to boundary, for effective particle operations.

5. Execute particle push operations including particle surface interactions.

4.1. Geometric Model

The definition of the geometric domain to be meshed in an impurity transport simulation can contain a number of physical components which, in the case of a tokamak, will include the walls of the containment vessel and other plasma facing components such as divertor cassettes, limiters, diagnostic probes, etc. To support the effective specification of the analysis control parameters and mesh size/resolution control, it is desirable that the geometric model provided to the mesh generation code also includes physics geometry such as selected flux curves [29]. Since the resulting geometric domain can be quite complex, it is desirable to be able to apply fully automatic mesh generation methods. Automatic mesh generators such as Gmsh [30] and MeshSim [31] require a properly defined geometric model. In the current work, tools such as Open Cascade [32] and SimModeler [31] support the definition of such geometric models.

This subsection details the procedure for creating the geometric model used in GITRm for the ITER and DIII-D tokamaks. Although the procedure is described for these specific cases, the procedure is general and can be applied to other cases. Two curves are of specific importance for creating the geometric model of the computational domain to be used in the impurity particle simulations: the wall curve representing the main chamber wall of the tokamak device and the separatrix curve. The wall curve is obtained from the GEQDSK data [33]. It is useful in cases when other geometric/wall features (e.g., bumpers or probes) must be included in impurity particle simulations. GEQDSK-based wall curve does not necessarily conform to the SOLPS mesh [9]. SOLPS is the edge-plasma model that is used to obtain the plasma fields for the current ITER case. This non-conformity is shown in the left image of Figure 2 for a r-z poloidal plane. To address this issue, the edges of the SOLPS mesh in the plasma facing region are merged with the wall curve information obtained from the GEQDSK data. Thus, the geometry of the inner and outer divertors, i.e., target surfaces, conforms exactly to the edges of the SOLPS mesh in that region as shown in the right image of Figure 2. The separatrix curve including the X point is also extracted from the SOLPS mesh. The presence of the separatrix curve in the geometry is necessary for setting mesh parameters to create highly graded and anisotropic mesh locally in that region to capture the sharp directional gradients in the plasma fields.

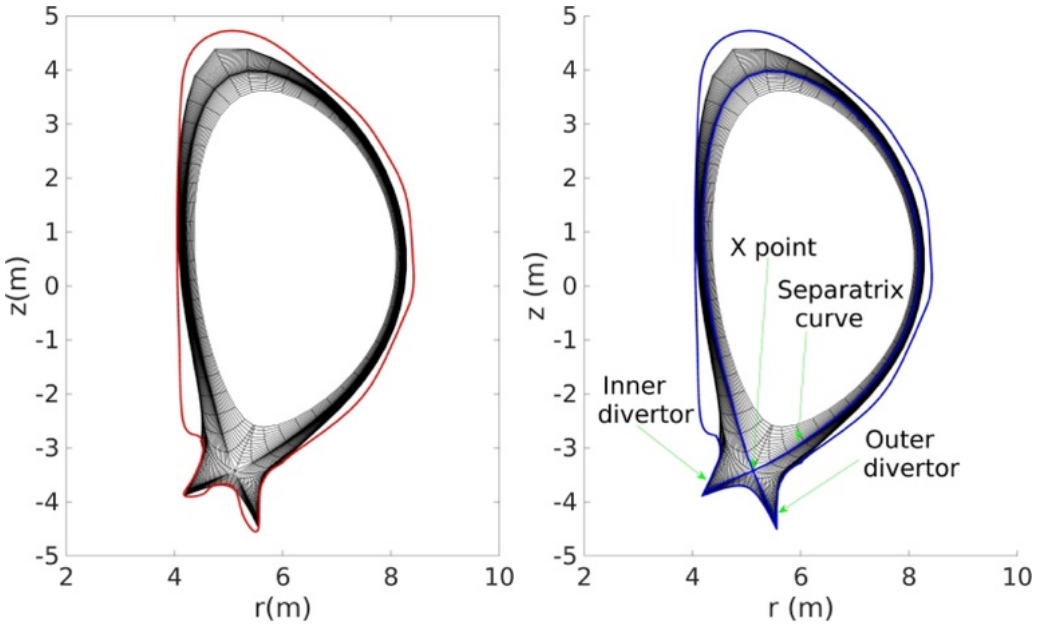


Figure 2: Left: Non-conformity between the boundary of the SOLPS mesh (in black) and the GEQDSK-based wall curve (in red). Right: Geometric model curves (in blue) consisting of separatrix and combined outer boundary (based on the wall curve from the GEQDSK data and the boundary of the SOLPS mesh at the targets).

The merged boundary and the separatrix curve are given as inputs to the TOMMS software [29, 34] which creates the CAD geometry with the desired flux curves including the separatrix. The poloidal plane geometry is revolved in the toroidal direction to obtain the 3D geometry. Figure 3 shows a cross sectional view of the 3D ITER geometry thus created. The separatrix curve and the target surfaces which conform exactly to the edges of the SOLPS mesh are surfaces of specific interest and highlighted in yellow.

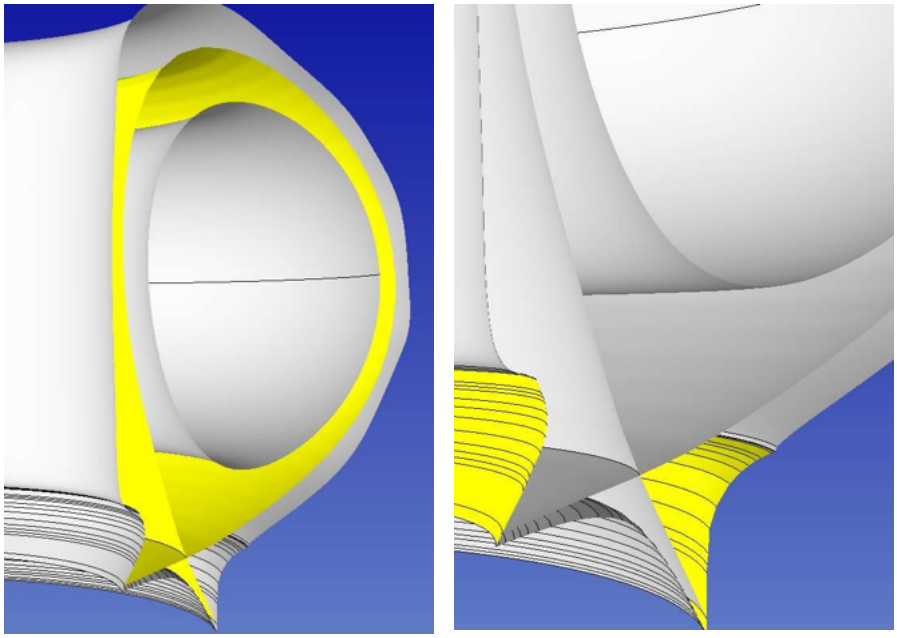


Figure 3: A cross sectional view of the ITER tokamak geometry highlighting the separatrix surface (on the left) and the target surfaces (on the right).

GITRm has the capability to deal with non-axisymmetric tokamak geometries and the challenges associated with studying impurity transport in critical regions far from the plasma facing surfaces. For this study we select the DIII-D tokamak and introduce non-axisymmetry through the addition of bumper limiters and collector probes. Bumper limiters are used to absorb the energy of the plasma before they can reach the walls [35] and can be an important part of tokamak design. The probes inserted into the plasma are used to measure the impurity particle content in the far scrape off layer which can be used to infer the impurity transport mechanisms [36].

The outer boundary required for creating the geometric model of the computational domain is obtained from a combination of the wall curve (main chamber wall of the tokamak device) based on the GEQDSK data and the targets based on the OEDGE mesh. In the present study, the background fields for the DIII-D case are obtained from the edge-plasma code OEDGE [37]. The separatrix curve is also obtained from the OEDGE mesh as shown

in the left image of Figure 4. The tungsten plate which faces plasma spans from $r = 1.4014$ m to 1.4568 m and is shown in magenta in the right image of Figure 4. The 3D geometric model is created by rotating the poloidal geometry in the toroidal direction. The 3 bumper limiters are located at angles 0°, 135° and 225° with respect to the x-axis and are indicated in red lines in the left image of Figure 5. The bumper limiters are formed by portions of cylinders intersecting with the 3D geometry. These cylinders have a diameter of 1.06 m and are situated radially at a distance of 2.826 m from the center of the tokamak. The cylinders have a height of 0.9 m. The collector probes are located at an angle of 215° with respect to the x-axis and indicated by a green line in the same image. The probes are 230 mm long and have diameters of 5 mm, 10 mm and 30 mm. The final geometry is created by following the above procedure and zoomed portions of this geometry near the one of the bumper limiters and the probes are shown in the middle and right image of Figure 5, respectively.

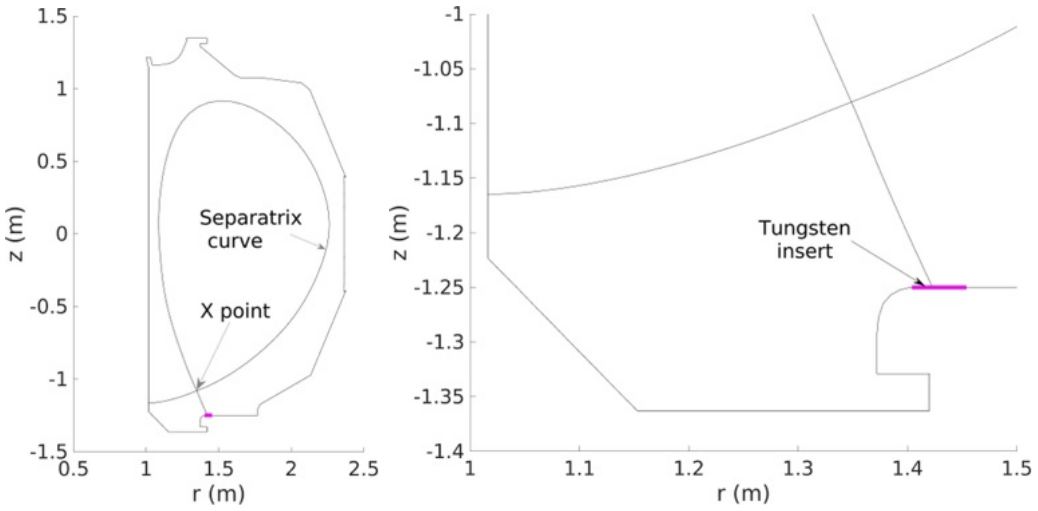


Figure 4: Left: Separatrix curve. Right: The magenta portion represents the tungsten insert.

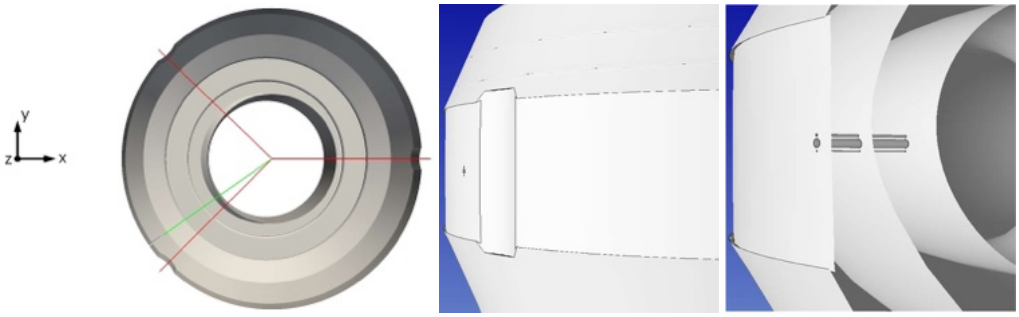


Figure 5: Left: The position of the bumpers and the probes in the tokamak. Center: A zoomed portion of the DIII-D geometry near one of the bumpers. Right: Collector probes which are of interest.

4.2. Mesh and Field Transfer

The edge-plasma codes SOLPS or OEDGE (see Figure 6 as an example) from which GITRm obtains the background fields are based on 2D block structured meshes mapped to a rectilinear computational domain. However, GITRm employs a fully unstructured 3D mesh. The accurate transfer of fields from a 2D SOLPS or OEDGE mesh to a fully 3D unstructured mesh used in GITRm requires careful control of the resolution in the 3D unstructured mesh.

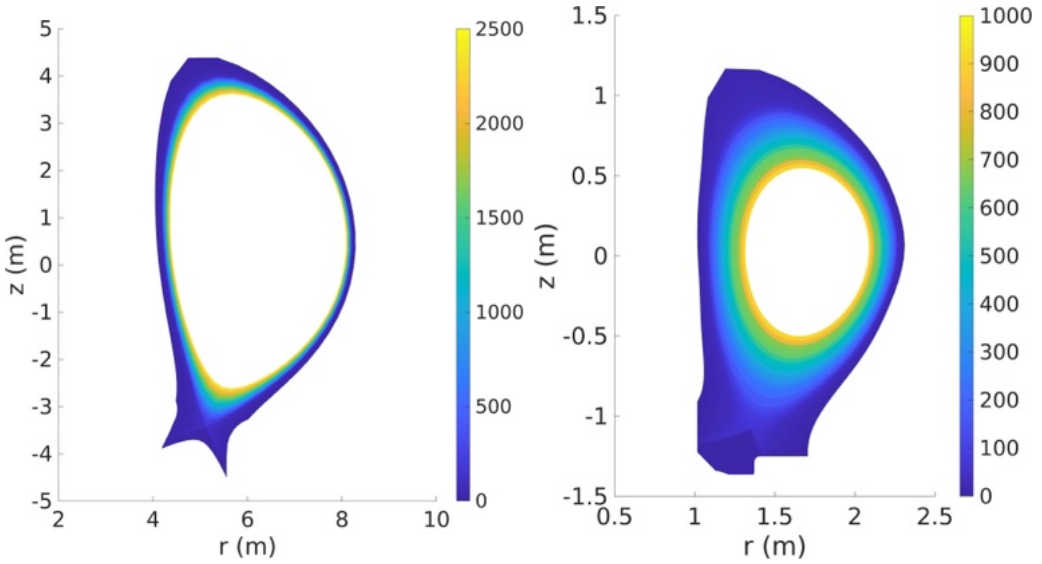


Figure 6: Electron temperature in the 2D SOLPS mesh (on the left) and the OEDGE mesh (on the right) from which GITRm obtains the plasma fields.

Thus, the 3D unstructured mesh used in GITRm is constructed by determining and specifying the desired mesh resolution (including anisotropy) at different locations and components of the geometry. The mesh control parameters are applied to various components and flux surfaces of the tokamak. A robust workflow for mesh control is developed to identify the optimal local mesh parameters needed at different locations of the geometry. The first step in the workflow involves identifying locations in the poloidal plane where the plasma fields vary drastically. Identified locations for the tokamak reactors include the regions close to separatrix surface and divertor target surfaces. At these locations the edge-plasma grid uses highly graded and anisotropic elements. To reproduce such fields on the 3D unstructured mesh, graded and layered elements with high anisotropy are used. The parameters defining the layered elements are the thickness of the first or smallest layer, the gradation or growth rate and the total thickness of layered elements. Of these, the first layer thickness is estimated as the smallest element size in the edge-plasma grid at all locations of interest. However, the other two mesh control parameters require additional processing to ensure that the 3D unstructured mesh does not end up with an unnecessarily high number of elements. This is done by analyzing the variation of the fields normal to specific surfaces of interest (e.g., separatrix) and identifying the extent of gradation in elements needed

to represent a given field. An estimate is made by calculating the second derivative of the fields and locating when its value drops beyond a certain relative tolerance (e.g., 95% of the peak value). An example of this is shown in the left image of Figure 7 where the variation of the second derivative of the electron temperature in the SOLPS mesh is plotted as a function of the element number normal to the separatrix towards the scrape off layer. This is done for all the fields and the final control parameters for the boundary-layer mesh are taken to be the minimum or most restrictive mesh parameters estimated from every plasma field. A crinkled cross-sectional view of the fully 3D unstructured mesh for the ITER geometry, created by employing the above stated procedure is shown in the right image of Figure 7. Note that a crinkled view is more useful for a cross section of 3D graded and/or anisotropic meshes and thus, is used subsequently for any cross-sectional view of 3D meshes.

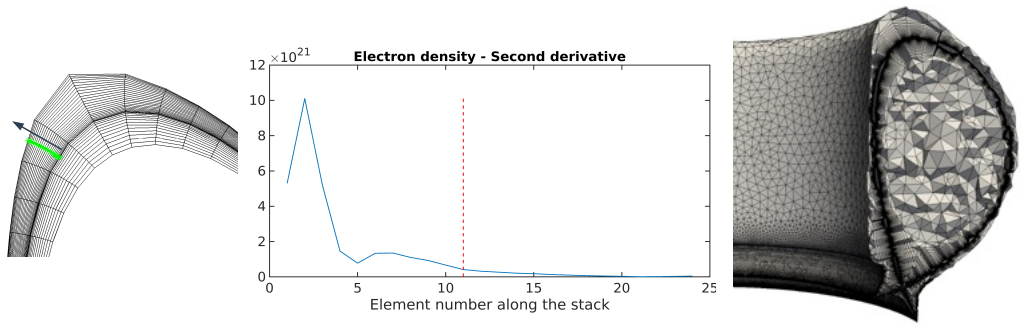
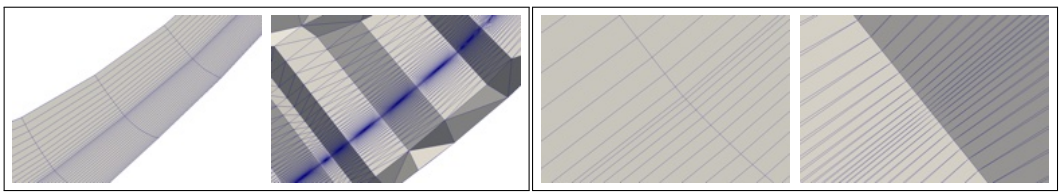


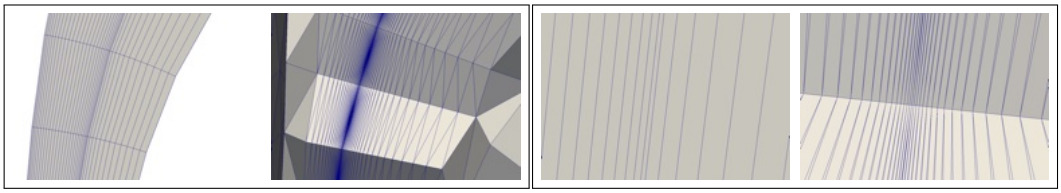
Figure 7: Left and center: Control parameter estimation on separatrix surface towards wall. Right: The 3D unstructured mesh for the ITER geometry.



(a) SOLPS (left) and GITRm (right) meshes

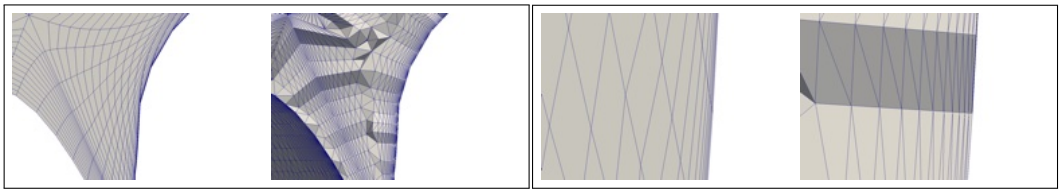
(b) SOLPS (left) and GITRm (right) meshes

Figure 8: A comparison of the SOLPS and GITRm meshes near the lower side of the separatrix for the ITER geometry (the right pair of figures shows a zoomed-in view).



(a) SOLPS (left) and GITRm (right) meshes (b) SOLPS (left) and GITRm (right) meshes

Figure 9: A comparison of the SOLPS and GITRm meshes near the upper side of the separatrix for the ITER geometry (the right pair of figures shows a zoomed-in view).



(a) SOLPS (left) and GITRm (right) meshes (b) SOLPS (left) and GITRm (right) meshes

Figure 10: A comparison of the SOLPS and GITRm meshes near the outer target for the ITER geometry (the right pair of figures shows a zoomed-in view).

A comparison of the SOLPS and GITRm meshes are provided at three different locations in Figures 8, 9 and 10. Figures 8 and 9 show these meshes near the separatrix on the lower and upper sides, respectively. Zoomed-in views are also provided for clarity. Figure 8, for the lower side of the separatrix, clearly shows that the GITRm mesh has a similar resolution as the SOLPS mesh along both the normal and lateral/tangential directions to the separatrix. Note that in the case of GITRm, the separatrix is represented as a 3D surface within the volume and is used to generate highly graded and anisotropic elements as layered stacks in a local neighborhood. This is done to capture the sharp directional gradients in the plasma fields. Away from the layered stacks, unstructured (isotropic) tetrahedral elements are used. The layered stack of anisotropic elements, and unstructured tetrahedral elements surrounding them, can be seen in the zoomed-out view (i.e., in the GITRm mesh in the left pair of figures). Figure 9, for the upper side of the separatrix, shows that the GITRm mesh has a similar resolution in the tangential direction and a finer resolution in the normal direction. This is because in GITRm mesh the normal resolution is kept to be a constant along the entire separatrix and is equal to the minimum value of the normal

resolution in the SOLPS mesh (i.e., SOLPS mesh has a normal resolution that varies along the separatrix). Similarly, Figure 10 shows the SOLPS and GITRm meshes near the outer target surface, where both meshes have a similar resolution along the normal and tangential directions to the target surface.

The fields in edge-plasma models are piecewise constant. Thus, the field transfer from the the edge-plasma mesh to the 3D unstructured mesh is accomplished by locating the projection of the centroid of the unstructured tetrahedral element in an element of the edge-plasma mesh and assigning the field value from the edge-plasma mesh element to the unstructured tetrahedral element. The fields transferred onto the 3D unstructured mesh are also piecewise constant. In addition, any unstructured tetrahedral element residing out of the edge-plasma domain is assigned a default value of 0 (e.g., in the core region). This is acceptable since in case of target sources, only a very small fraction of impurity particles (if any) reach such regions and thus, are statistically insignificant. We note that the gradients of the appropriate fields are also transferred in a similar fashion from the edge plasma mesh to the 3D unstructured mesh, i.e., the gradients are not directly computed on the 3D unstructured mesh. We also note that the magnetic field is obtained from the equilibrium file and is interpolated from the uniform rectilinear grid in the equilibrium file to the particle position directly using bi-linear interpolation.

Further, a coarser resolution is used in the unstructured mesh in such regions (e.g., in the core region). The current procedure for the field transfer resulted in a relative $L2$ error in any transferred field to be below 0.1%. For any field Q, the relative $L2$ error is defined as: $\sqrt{\frac{\int_\Omega (Q_{SOLPS}-Q_{GITRm})^2 d\Omega}{\int_\Omega Q_{SOLPS}^2 d\Omega}}$ (where, Ω is the domain over which the source field Q_{SOLPS} is defined). A comparison of the background electron density on the 2D mapped structured meshes and the 3D unstructured meshes used in GITRm is shown in Figure 11.

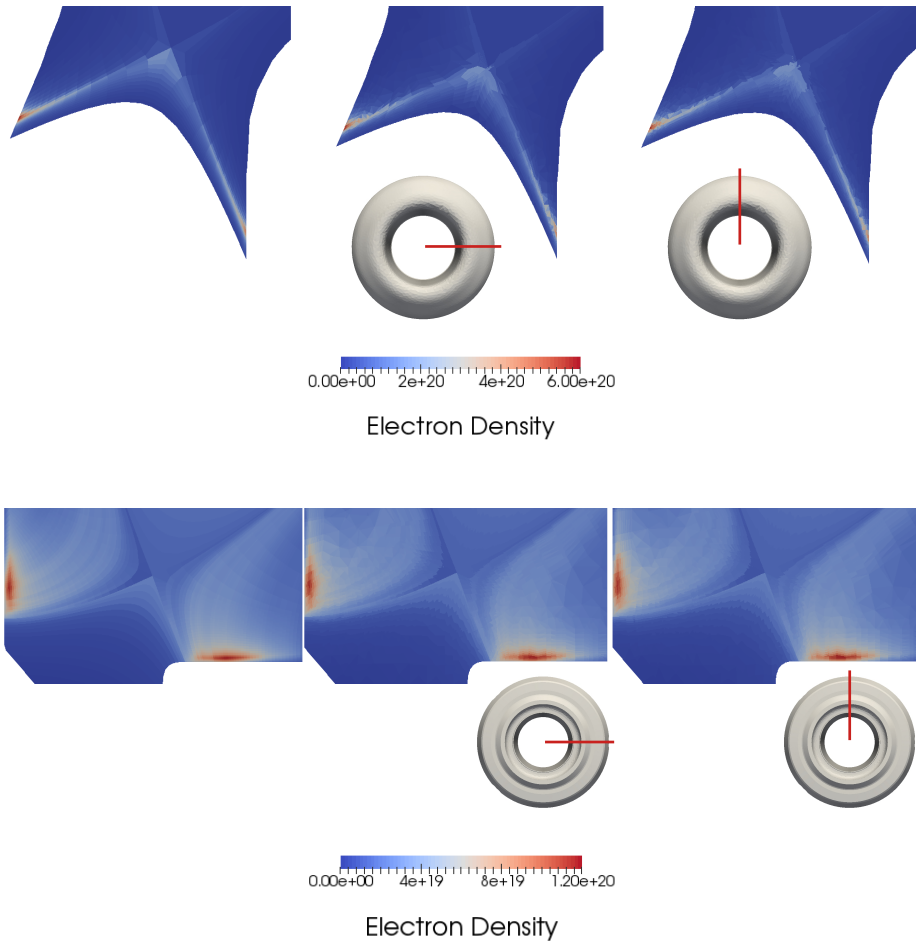


Figure 11: Top: Transfer of background electron density for the ITER case from the SOLPS mesh (the left image) to different poloidal planes on the GITRm mesh (the middle and right images). Bottom: Transfer of background electron density for the DIII-D case from the OEDGE mesh (the left image) to different poloidal planes on the GITRm mesh (the middle and right images).

4.3. Particle Initialization

This work is concerned with the tracking of impurities that enter the plasma from specific plasma facing components. To support these simulations, GITRm includes procedures for initiating impurity particles from selected boundary surfaces that represent the plasma facing components. The initialized particles represent the impurities eroded by the incoming plasma.

So, the inputs required for initializing the particles are the ion fluxes to the plasma facing components (obtained from the edge-plasma code) and the yield rates of the ion species on these surfaces. Yield rates are obtained by providing energy and angle distribution of the ions to the BCA code, where the energy and angle distribution must include sheath/pre-sheath effects and can be obtained from an analytic form, an independent precursor (GITRm) simulation or a high-fidelity sheath simulator (PIC code). GITRm initializes particles with a distribution based on the particle flux which is the product of the ion flux and the yield rate for the ion species.

The input particle flux helps in creating the spatial distribution of the initialized particles. To do this, a novel procedure is used in which particles are independently initialized and associated to each mesh face residing on the relevant boundaries, i.e., target surfaces. On a distributed/partitioned mesh, this can be done in an embarrassingly parallel way. Another possibility is to initialize particles globally independent of the mesh and then use a search algorithm to associate them to unstructured mesh elements. This is not a scalable procedure in parallel for a partitioned mesh. Currently particles are initialized randomly within each relevant mesh face as presented in Algorithm 1, which is described later. The number of particles within each relevant mesh face depends on its area and the local flux. Note that currently in GITRm a tetrahedral volume mesh is used and thus, a triangular surface mesh is used.

The top image of Figure 12 shows the particles initialized on one of the triangular faces of a tetrahedron. In this algorithm, a wall-normal distance can be specified (if needed) to initialize particles slightly off the walls for numerical stability, as shown in the bottom image of Figure 12.

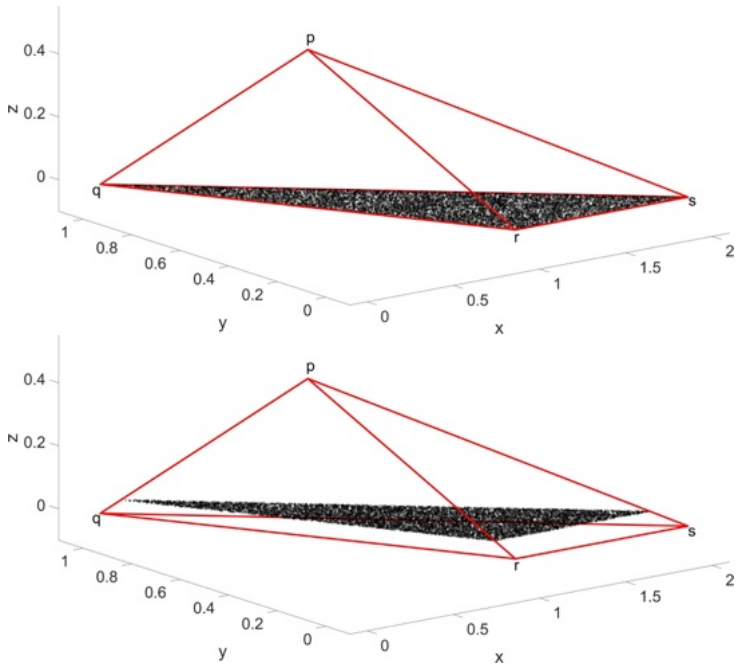


Figure 12: Top: Particles initialized on one of the faces of a tetrahedral element. Bottom: Particles initialized inside a tetrahedral element at a fixed distance from one of the faces.

Algorithm 1 Particle Initialization

a is any tetrahedral element in 3D unstructured mesh
$\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the barycentric co-ordinates of any tetrahedron
$G_j^2 =$ the j^{th} entity of dimension 2 in geometric model G
$M_i^2 =$ the i^{th} entity of dimension 2 in mesh model M
$K = M_i^2 \subset G_j^2$ ▷ Set of mesh faces on the relevant boundary
p, q, r, s are the co-ordinates of the vertices of a tetrahedron
h is the height of the tetrahedron from the vertex opposite to the mesh face residing on the boundary
l is the user-specified distance from the boundary where to initialize particle
x is the initialized particle position
for any k in K **do**
    $a = k\{M^3\}$ ▷ Upward adjacency to find the element
    $\lambda_4 = l/h$
    $y_1 \sim \mathcal{U}[0, 1]$ ▷ Uniformly distributed random number
    $y_2 \sim \mathcal{U}[0, 1]$ ▷ Uniformly distributed random number
    if $y_1 > y_2$ **then**
        $\lambda_1 = y_2,\ \lambda_2 = y_1 - y_2,\ \lambda_3 = 1 - \lambda_4 - y_1$
    else
        $\lambda_1 = y_1,\ \lambda_2 = y_2 - y_1,\ \lambda_3 = 1 - \lambda_4 - y_2$
    end if
    $x = p\lambda_1 + q\lambda_2 + r\lambda_3 + s\lambda_4$
end for

GITRm supports initializing the particles with a given energy distribution. In addition, particles can be initialized with an initial velocity vector whose direction is perpendicular to the surface or with a cosine distribution as described in [38]. All particles are initialized with unit weight and no charge. During the course of the GITRm simulation, the motion of the particles and the atomic operations like ionization or recombination are computed as described in Section 2.2. Particles may encounter a physical wall during the course of their motion, which is handled during a push step. If the physical wall is a material boundary that can sputter or reflect, the surface response model as described in Section 2.1 decides the outcome.

In GITRm, particles are initialized in the core region of a PICpart and since each PICpart is present in a different MPI process, particles are ini-

tialized on different processes in an embarrassingly parallel fashion. Figure 13 illustrates particle initialization for a PUMIPic-based mesh partition consisting of 4 PICparts across 4 MPI processes, where particles are initialized on each MPI process on the outer divertor surface of the tokamak. Note that PUMIPic has been demonstrated on problems with mesh partitions consisting of thousands of PICparts, see [1].

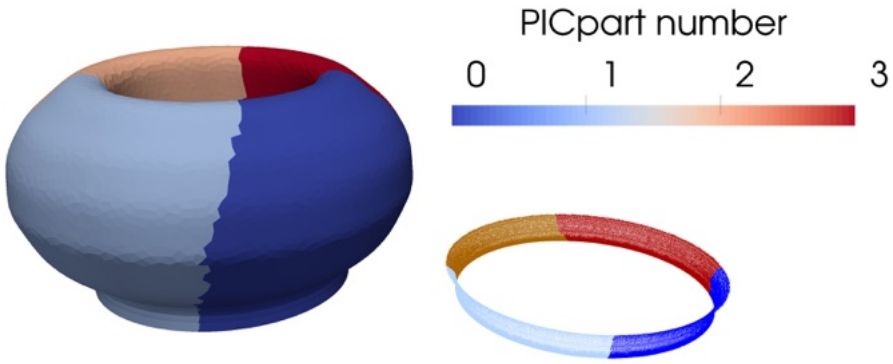


Figure 13: Illustration of particle initialization: Core region of the PICparts of a mesh partitioned into 4 parts (on the left) and particles initialized on the outer divertor surface in each of these PICparts (on the right).

4.4. Sheath Electric Field Calculation

Calculating the sheath electric field in GITRm through analytical models requires calculating the distance between the particle and the nearest plasma facing material surface as discussed in Section 2.2. The variation of the sheath electric field with distance at a point in the outer divertor for the ITER tokamak under He burning plasma is shown in Figure 14 along with a nominal electric field value obtained from SOLPS at a distance of 1 cm normal to the wall. In this case, it is seen that the sheath electric field becomes relatively negligible within a normal distance of 1 cm from the wall. Recall that a typical tokamak size is on the order of metres and thus the sheath electric field calculation can be limited in a small portion of the domain and mesh. Note that the evaluation of the sheath electric field is a computationally expensive operation since it requires the determination of the closest distance to a boundary for all particles after every push operation.

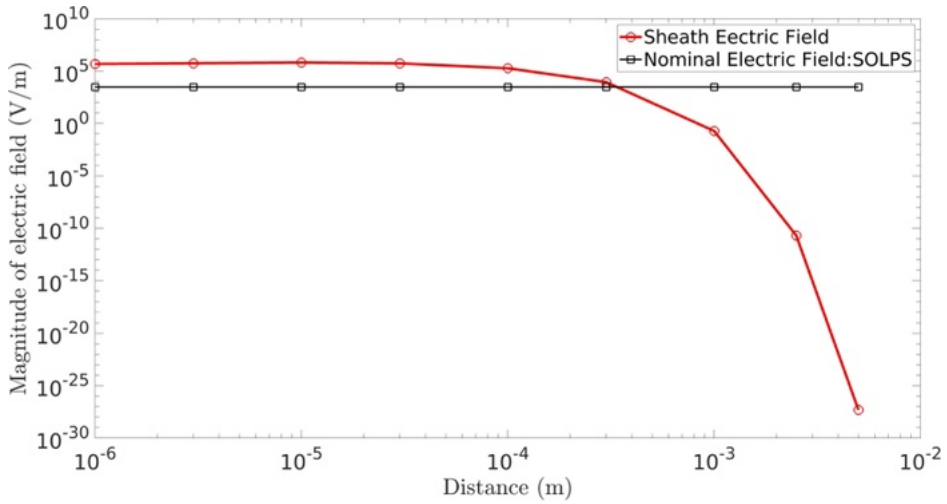


Figure 14: Sharp decay of the sheath electric field along the surface normal upto a distance of 1 cm (a nominal value of the electric field obtained from SOLPS is included for reference).

GITRm avoids unnecessary sheath electric field calculations beyond a certain pre-defined distance by tagging elements near the relevant boundaries. The relevant boundaries can be specified by the user/application. The user may specify all the wall boundaries/surfaces to be relevant for sheath calculation or only a specific set of material surfaces where sputtering or reflection can occur, for example, at the tungsten insert in the DIII-D tokamak. This is a useful choice because the influence of sheath/pre-sheath electric field on impurity particles can be dominant around certain specific boundaries. Figure 15 shows the elements whose centroids lie within a distance of 2 cm from the tungsten insert and the sheath electric field calculation can be limited to particles located only in this region. By tagging elements, the calculations of near-zero contributions far from the relevant boundaries are avoided. This requires an a-priori estimate of the region where the sheath electric field is prominent as shown in Figure 14.

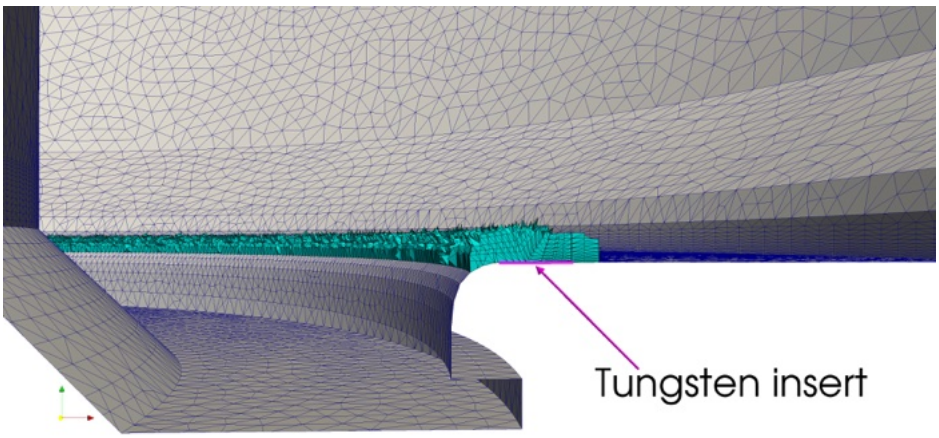


Figure 15: Left: Tetrahedral elements whose centroids lie within a distance of 2 cm from the tungsten insert in the DIII-D tokamak (a part of the surface mesh is shown for perspective).

Calculating the distance between a point and the target surface involves finding the minimum distance between the point and the triangular mesh faces on the boundary. Even if the distance to boundary calculations are performed on that part of the domain where elements are tagged and the sheath electric field is prominent, calculating the distance between a point and all the triangular mesh faces classified on the boundary is expensive. GITRm addresses this problem by performing a pre-processing step that stores a list of mesh faces residing on the boundary that are closest to a tetrahedral element and in the course of the simulation, the distance to boundary calculation between a point and the nearest surface can be limited to a relatively small subset of boundary mesh faces. Note that the subset of boundary mesh faces for a given element can reside on multiple target surfaces.

The distance between a point and a triangle can be estimated by calculating the distance between the point and the vertices of the triangle, the point and the edges of the triangle, and the point and the projection of the point on the plane of the triangle. However, the more efficient method as described in [39] is used by checking in which Voronoi region of the triangle, the orthogonal projection of the point falls on. It is important to note that this process uses the surface mesh/triangulation. A situation can arise for a smooth curved geometry where a given point inside the domain has the closest point on the surface mesh (forming the shortest distance) that is shared

between two or more triangles, see point labeled P2 in Figure 16. In this situation, the electric field direction is taken from the given point to the closest point on the surface mesh, i.e., pointing towards the surface along the line joining the two points. Such a situation also arises for non-smooth geometries with sharp corners or edges. We also want to note that for the current cases, the sheath electric field is only considered around selected geometric surfaces that are smooth and curved.

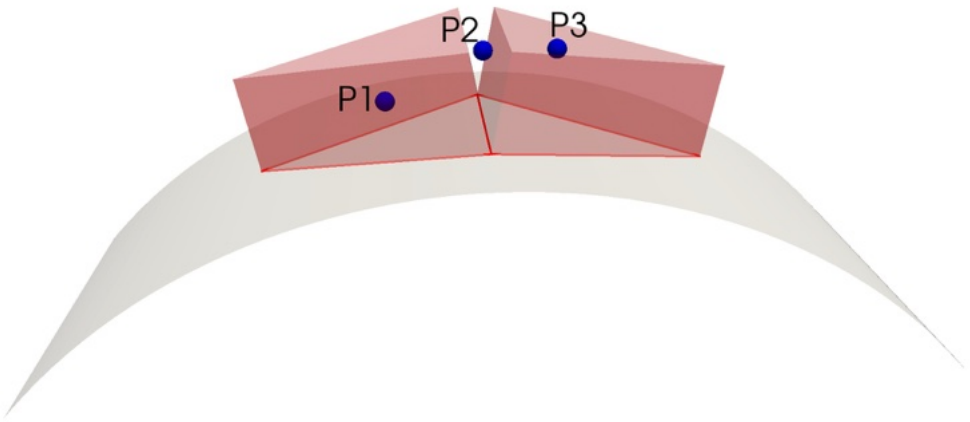


Figure 16: 3 arbitrary points P1, P2 and P3 are shown within the domain along with two triangles on a smooth curved surface: points P1 and P3 each have the closest point on the surface within a triangle whereas the point P2 has the closest point on the surface that is shared by two triangles.

To summarize, the distance to boundary calculations in GITRm involves:

- Marking of the elements that have any portion of the element within the selected cut-off distance and the sheath electric field calculations are limited to only the particles within those elements.
- In the pre-processing step, for each tetrahedral element in the mesh, a list of boundary mesh faces that are closest to it are stored and during the simulation, the search for the distance between any particle in the element and the boundary can be limited to the list of pre-processed boundary mesh faces.

- An efficient method of calculation of distance between a point and a triangle is used.

Following the procedures, the distance to boundary calculations are performed and Figure 17 shows the nearest boundary mesh faces for a tetrahedral element in the domain. The mesh used has about 11.4 million elements and about 200,000 mesh faces on the divertor surfaces.

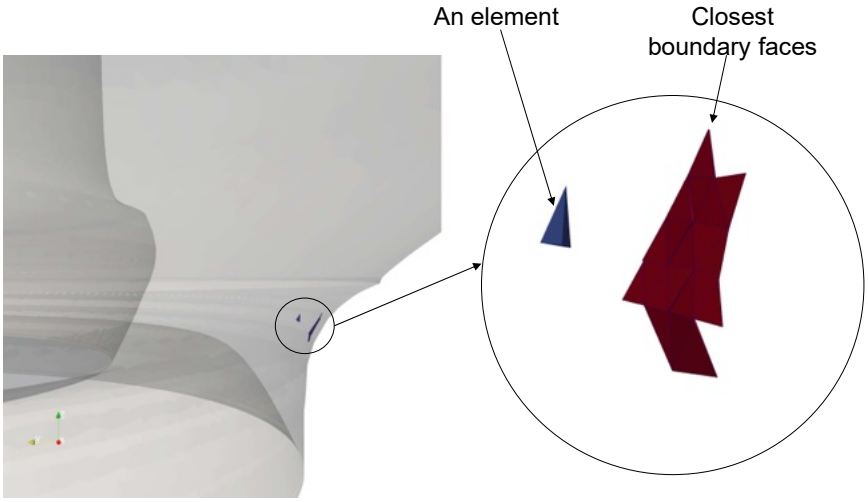


Figure 17: A tetrahedral element in the domain and the boundary mesh faces closest to it.

4.5. Particle Push

The equations governing the motion of the particles have been discussed in Section 2.2. The integration of Equation 2 is done in two parts. The first part involves ODE integration of the Lorentz force using the Boris algorithm/pusher [25, 40, 41]. The Boris algorithm is a second-order accurate time centered algorithm and has been widely adapted for particle-in-cell (PIC) codes [42]. The second part involves giving a velocity kick to the particle at each time step which is governed by Equation 7. Equation 8 adds a displacement in the perpendicular to field line direction representing anomalous cross-field diffusion.

At the core of the push operation calculation are the plasma fields that are a function of their position within the domain, the proximity of the particle to specific boundary surfaces (due to the presence of the sheath electric field

as discussed in the previous subsection) and the local time step. In the PUMIPic library, the mesh is the primary data structure and the particles are accessed through the mesh. So, the field value at any of the particle's location required for the push operations can be quickly accessed from the mesh element data. This requires that after each push operation the mesh particle association is updated as the particles can cross or leave the current element.

4.6. Particle Wall Interaction Calculations

The execution of the GITRm surface model requires accurate determination of where particles hit the walls and their velocity vector as they strike the wall. If a particle's path intersects the boundary during the particle search operation, then PUMIPic reports the intersection point and mesh face where the intersection occurs. A particle that hits the wall is then marked/flagged as such in GITRm and no further physics operations are carried out on that particle until the surface model decides the surface response, and the particle is either deposited or re-initialized with reflected/sputtered conditions at the next time step. As discussed in Section 2.1, a particle weighting approach is used in which the particle weight is updated by a factor that depends on the sputtering yield and reflection coefficient and similarly, gross deposition and erosion are tracked at the surface level during the simulation (for more details see [15]).

A high-level overview of the implementation of the surface model is given in Algorithm 2 for a computational particle. The functions $f_1(E_0, \theta_0)$ and $f_2(E_0, \theta_0)$ are look up-tables obtained from the BCA code F-TRIDYN which give the sputtering yields and reflection coefficients, respectively, for tungsten on tungsten interactions as described in Section 2.1. F-TRIDYN also gives distributions for the outgoing energy (which determines the magnitude of velocity) and angle (with the surface normal) of the reflected or sputtered particle. Recall that the in-plane angle of the reflected or sputtered particle follows a uniform distribution from 0 to 2π.

Algorithm 2 Surface Model (for a given computational particle)

E_0 ▷ Energy of incoming particle
θ_0 ▷ Angle of incoming particle
W_0 ▷ Computational weight of incoming particle
$Y_0 = f_1(E_0, \theta_0)$ ▷ Sputtering yield
$R_0 = f_2(E_0, \theta_0)$ ▷ Reflection coefficient
E_{out} ▷ Energy of outgoing particle
θ_{out} ▷ Angle (with surface normal) of outgoing particle
$\phi_{out} \sim \mathcal{U}[0, 2\pi]$ ▷ In-plane angle of outgoing particle
$\xi_{uni} \sim \mathcal{U}[0, 1]$ ▷ Uniformly distributed random number
$\xi_{surf} = \frac{Y_0}{Y_0 + R_0}$ ▷ Probability of sputtering
$W_{out} = W_0(Y_0 + R_0)$ ▷ Computational weight of outgoing particle
if $Y_0 + R_0 > 0$ **then** ▷ Particle will reflect/sputter
    if $\xi_{surf} < \xi_{uni}$ **then**
        Particle re-initialized with reflected conditions
    else
        Particle re-initialized with sputtered conditions
    end if
else
    Particle deposits
end if
if $E_{out} == 0$ **then**
    Particle deposits
    $W_{out} = 0$
end if

5. GITRm RESULTS and DISCUSSION

The results of the impurity transport simulation are carried out for three cases: the PISCES geometry, the ITER geometry and the DIII-D geometry. The PISCES case was used for verification and validation. In the ITER case, the ability of GITRm to deal with realistic tokamak geometries and effectively address impurity transport and re-deposition phenomena near the target surfaces is shown. Note that in the current ITER case the target surface is axisymmetric and does not include any castellations, which requires use of an appropriate sheath electric field and will be done in the future by coupling GITRm with a sheath simulator. In the DIII-D case we show

the capability of GITRm to effectively address non-axisymmetric features in tokamak geometries.

5.1. PISCES Case

The impurity transport model of GITRm is compared against the linear plasma machine experiment in PISCES-A [43, 44]. The specific experiment is referred to as the 'high flux experiment' with a background He plasma species, a peak electron temperature of 9 eV and a peak ion flux of 4 $\times 10^{22}$ $m^{-2}s^{-1}$. The geometry considered for this case consists of a tungsten base plate of radius 47.5 mm and a titania (TiO_2) bead tower sitting at a radius of 44.6 mm. Impurities are eroded from the tungsten base plate due to exposure to the He plasma. The first bead on the tower is 12.75 mm and is insulated. The remaining 13 beads are each 10 mm in height and used to collect impurity particles eroded from the tungsten base plate. Bead 1 is closest to the tungsten base plate whereas Bead 13 is the farthest. The tower has an outer radius of 5mm. The tungsten base plate and the titania tower beads of the PISCES geometry are indicated in the left image of Figure 18. The right image of Figure 18 shows the mesh on these surfaces. More details on the background fields and the particle source can be found in [45, 43]. For validation purposes, the mass loss of the tungsten base and the mass gained by the titania beads are compared with the experimental data. The number of particles striking the titania tower as well as the energy angle distributions of the particles striking back and reflected/sputtered off the base plate are shown for verification. The simulations are carried out with 1 million particles, for 100,000 time steps and with a constant time step of 5×10^{-9} s.

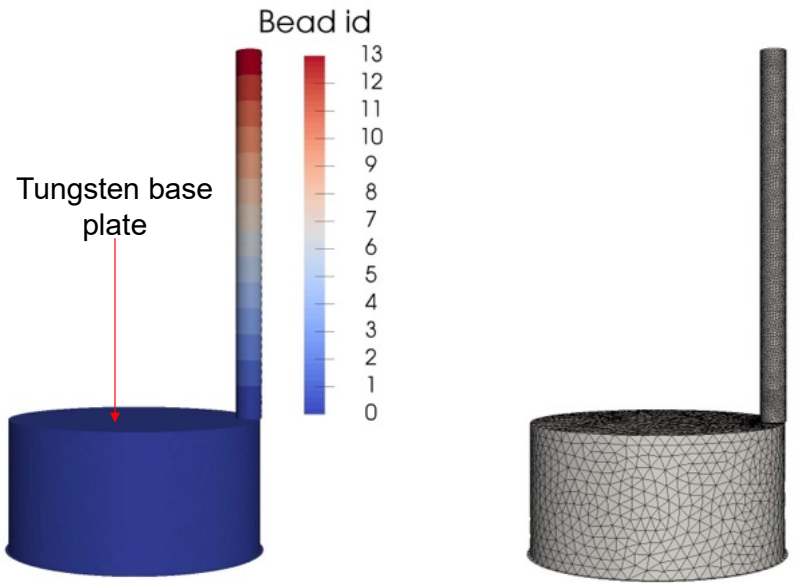


Figure 18: Left: The tungsten base plate and titania beads in the PISCES geometry. Right: The unstructured mesh on these surfaces.

Figure 19 serves two purposes, first to demonstrate the comparison between GITR and GITRm and secondly, the agreement in trends of the simulated quantities in GITRm with experiments. The left image shows the number of simulated computational particles deposited on the titania bead tower. This tower is an absorbing boundary, as sputtered tungsten atom energies are below the reflection and sputtering yield of titania and the bead tower is outside the plasma column where re-erosion can occur. The simulated particles are representative of a rate of particles traversing the plasma and hitting surfaces. The right image of Figure 19 demonstrates this conversion to compare with the experimentally measured values. GITRm captures the trend of decreasing mass gain on the titania tower with increasing axial distance from the target. As compared to the experimental data, GITRm over-predicts the deposited mass near the surface (low bead number) and under-predicts deposited mass further axially. The shape of this curve is largely attributed to the sputtered angle distribution of tunsten ions by both the He plasma and tungsten-tungsten interactions at the target surface. Further, the mass loss of the tungsten base plate is predicted as 76.43 mg by GITRm whereas the experimentally measured value is 79.53 mg. The small differences that are seen between GITR and GITRm can be attributed to

the different methods used in distance to boundary calculations.

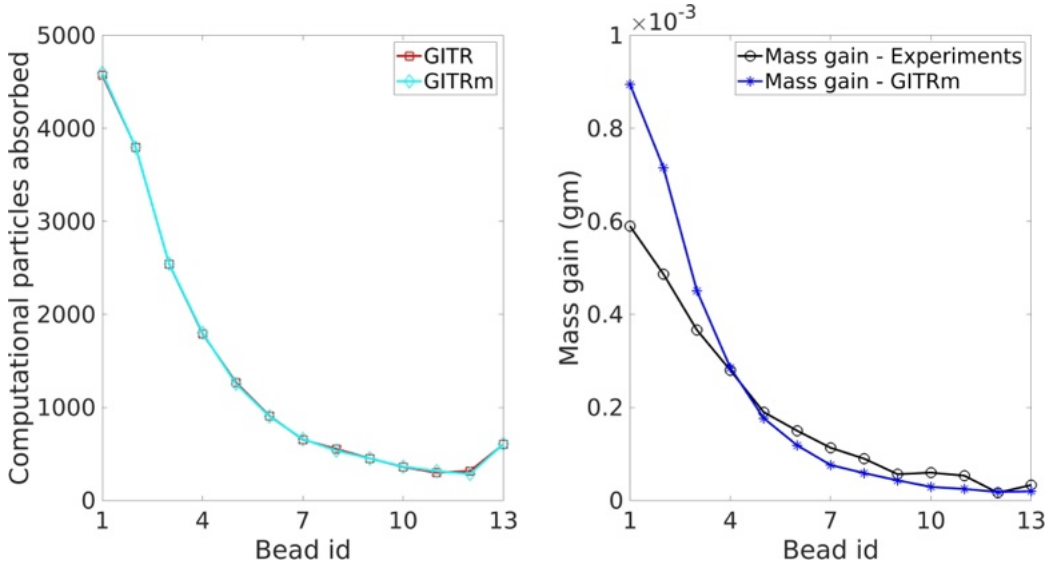


Figure 19: Left: Comparison of the number of computational particles collected in the titania beads between GITR and GITRm. Right: Comparison of the mass gain of the titania beads predicted by GITRm with experiments.

Figure 20 shows the tungsten ion energy-angle distribution at the tungsten base plate. The comparison of the right and left panes shows qualitative and quantitative agreement between GITR and GITRm. There are several features of these histograms which can be pointed out as physically significant. The dominant feature of these plots is a peak in impacting ion counts at 250 eV. This energy matches the bias voltage of the tungsten base plate, indicating a population of singly ionized tungsten atoms returning to the surface which has been accelerated through most or all of this surface potential/sheath voltage. These impacts are near normal incidence (0 degrees), which is as expected with the magnetic field configuration perpendicular to the tungsten base plate surface and low W ion temperature. This is because the perpendicular velocity of the W ion is small compared to that in the parallel direction from the surface electric potential. A tail of lower energy and higher angle impacts stem from this peak at 250 eV (going left and up on the plot) which represent ionized tungsten atoms which were accelerated through part of the sheath voltage. These represent tungsten atoms which were ionized in the sheath and returned to the surface, not accelerated through the

full surface bias voltage. Similar to the peak of impacts at 250 eV, there is a significant amount of ions striking the surface at approximately 500 eV. This peak corresponds to the doubly ionized tungsten atoms which were accelerated through the sheath. Outside of these dominant features, there is a scatter of many other impacting energy and angles. An array of various physics can be the cause of these smaller probability populations. Ionization, coulomb collisions, and reflection of tungsten ions can play a significant role in spreading impact energies and angles.

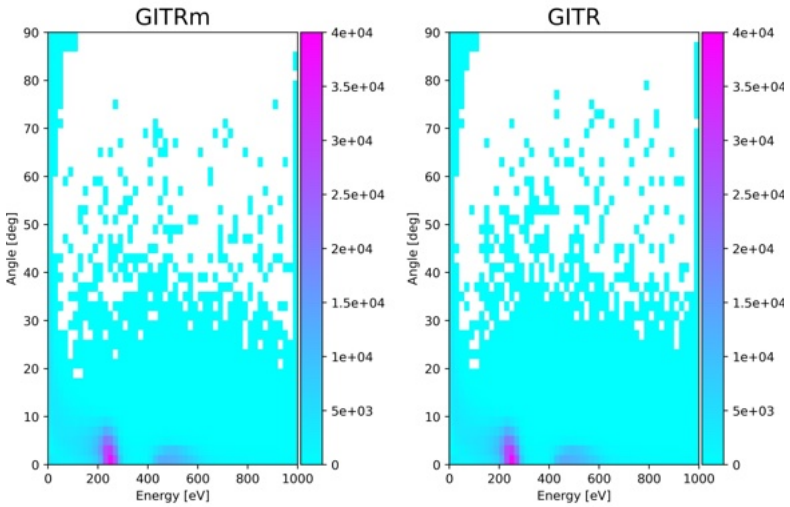


Figure 20: Energy-angle distribution of the weights of particles hitting the tungsten base plate.

Figures 21 and 22 demonstrate data collection within GITR and GITRm on the responses of the surface model to incoming ion impacts. These histograms indicate the characteristics of reflected and sputtered tungsten by incoming tungsten ions in the simulation. Based on the incoming tungsten ion energy-angle distribution (Figure 20), Figures 21 and 22 show the response of the surface. Figure 21 shows the energy-angle distribution of reflected particles. There is a significant reflected particle population of above 100 eV and with a peak in reflected angle away from normal incidence. The reflected particle angle distribution peaks between 40 and 70 degrees. This indicates, that despite many tungsten ions impinging near normal incidence (concentrated near 0 degrees, see Figure 20), the reflected angle (i.e., θ_{out})

distribution is based on the F-TRIDYN data used by the surface model, which is close to a cosine distribution. Figure 22 shows similar data for that of sputtered tungsten atoms. In this case the distribution function is integrated over energy to more clearly demonstrate comparison between GITR and GITRm. Once again, despite the near normal incidence impinging ion population, sputtered angles are peaked near 50 degrees. These results come about largely due to the input surface model which is compiled data from F-TRIDYN. GITRm collects this data as an investigative tool to give insight into the resulting physics of the simulation.

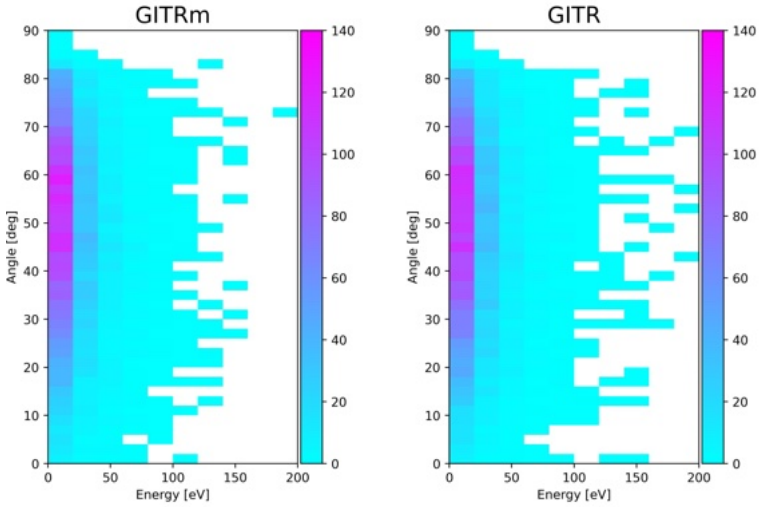


Figure 21: Energy-angle distribution of the weights of the outgoing reflected particles.

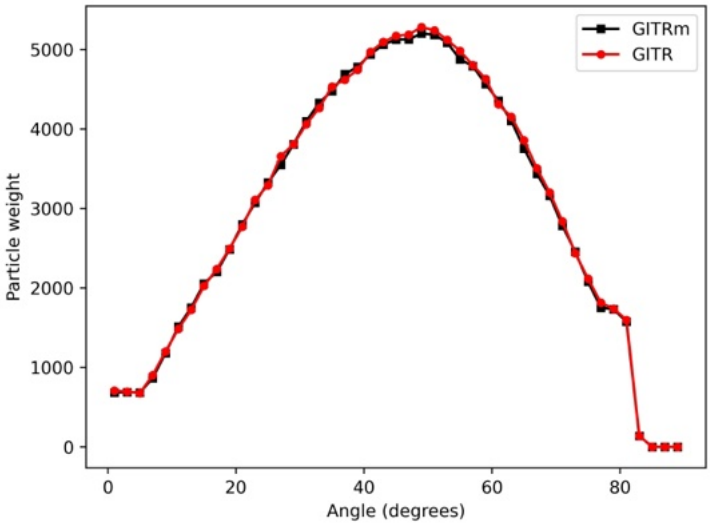


Figure 22: Variation of the weights of sputtered particles with angle, summed over all energies.

5.2. ITER Case

The procedure for obtaining the geometry, transfer of fields from the edge-plasma mesh and initialization of particles based on a given flux distribution have been discussed in Sections 4.1, 4.2 and 4.3. Simulations are carried out for a 40 MW He plasma discharge in both GITR and GITRm. Simulations are carried out with 2 million particles, for 100,000 time steps and with a constant time step of 1×10^{-8} s.

The relative difference in the gross erosion and deposition predicted by GITR and GITRm is less than 1%. These differences are expected due to the underlying differences in the calculations carried out by each code. Further GITR and GITRm comparison is performed as follows. Figure 23 shows a comparison of the variation of the (scaled) net deposition, gross deposition and gross erosion along the outer divertor target. The scaling is done by a reference/nominal value of 10^{19} $m^{-2}s^{-1}$. Figures 24 and 25 show a comparison of the energy-angle distribution of the weights of particles striking the outer tungsten divertor and the particles reflected off it, respectively. Figure 26 shows a comparison of the variation of the weight of the sputtered particles with angle, summed over all energies.

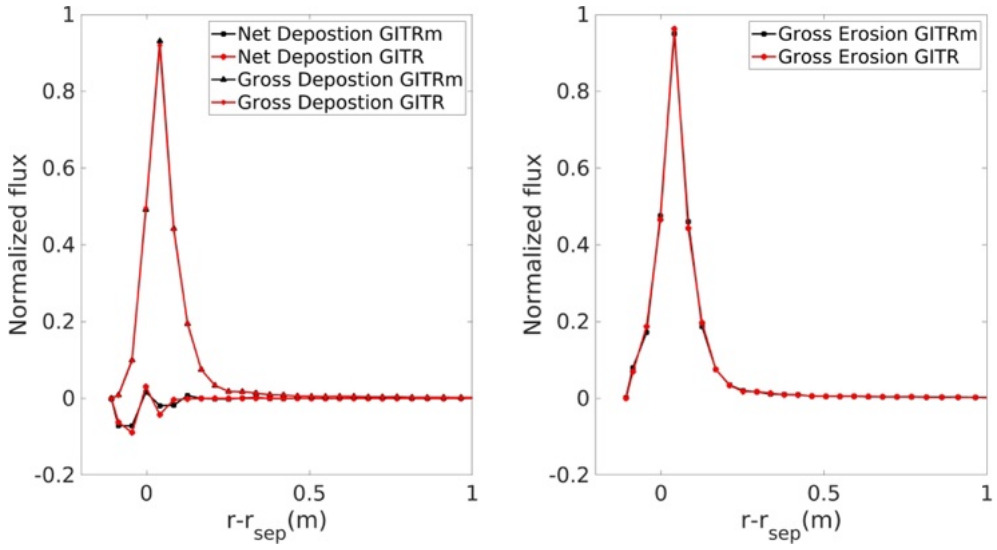


Figure 23: Variation of the (scaled) net deposition, gross deposition and gross erosion along the outer divertor target.

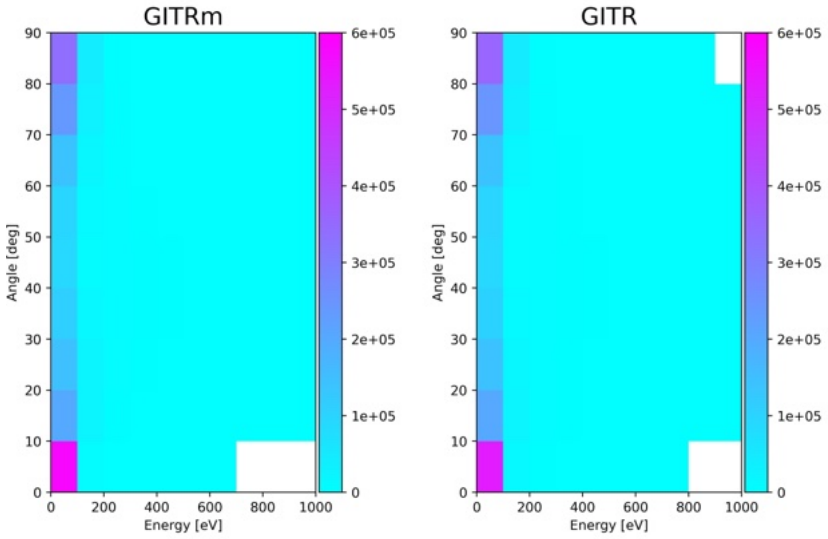


Figure 24: Energy-angle distribution of the weights of particles hitting the outer divertor target.

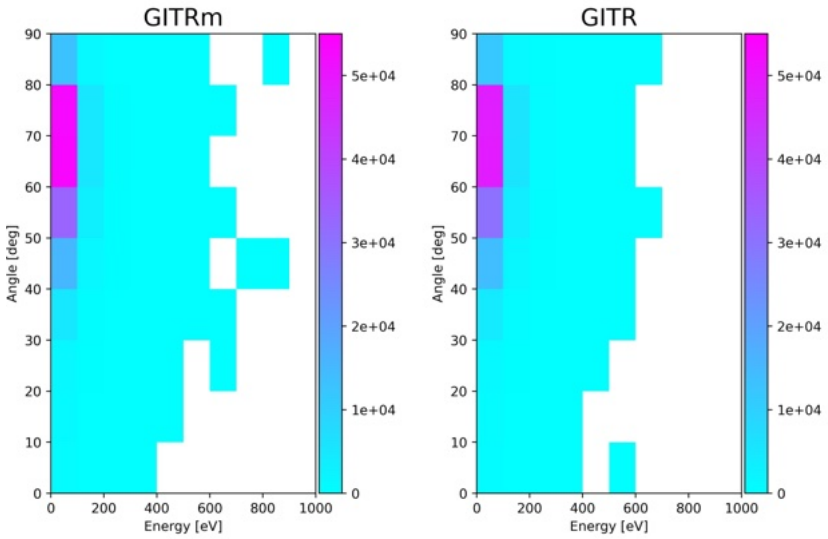


Figure 25: Energy-angle distribution of the weights of particles reflected off the outer divertor target.

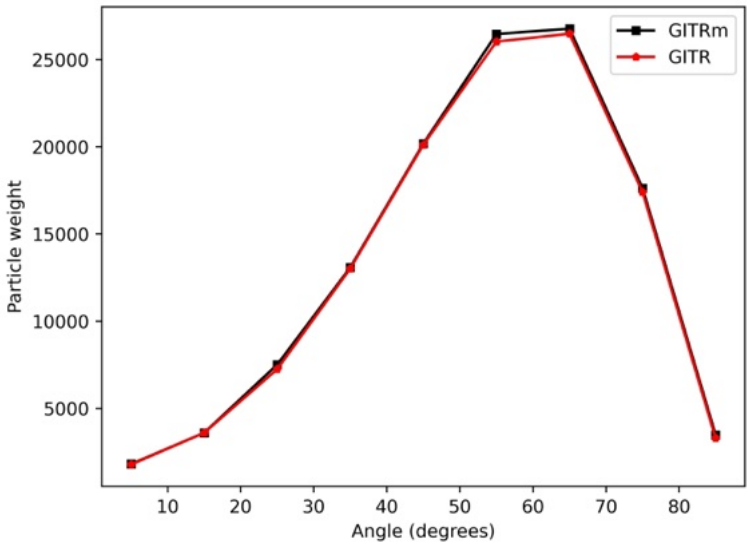


Figure 26: Variation of the weights of sputtered particles with angle, summed over all energies.

For demonstrating the long range impurity transport simulation capability of GITRm, the number of time steps in the simulations are increased to 500,000 while keeping the number of particles the same. Figure 27 shows a comparison of the long range impurity transport patterns between GITR and GITRm. Figure 28 shows zoomed views of the impurity particle densities near the top and bottom part of the poloidal cross section. In summary, the surface response and impurity transport results presented demonstrate the use of GITRm as a computational tool to effectively study transport and re-deposition phenomena of impurity particles on large tokamak geometries.

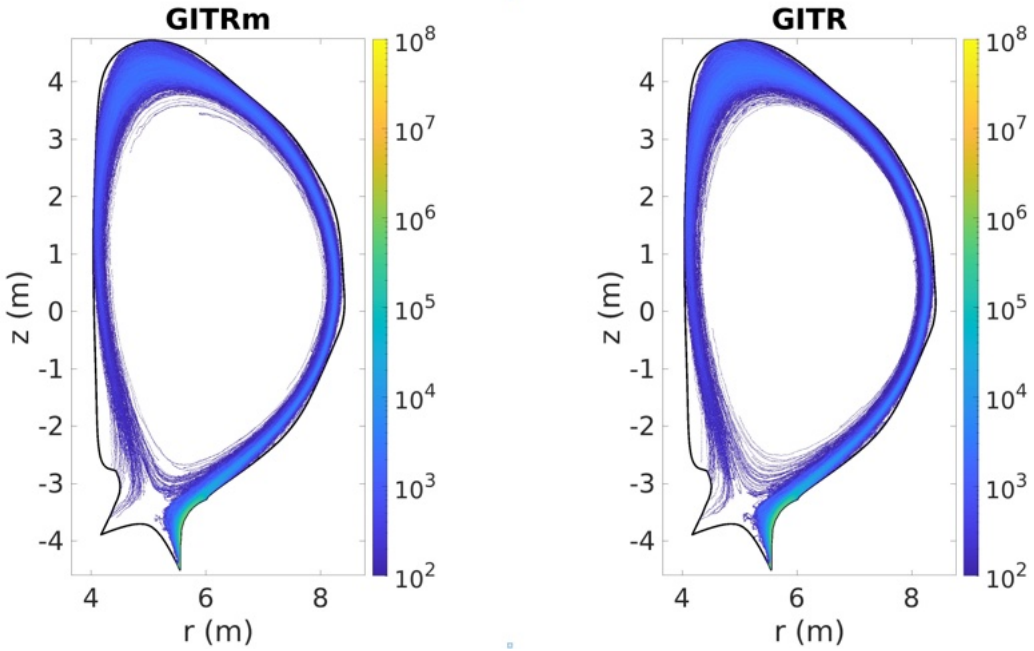


Figure 27: Impurity density of the particle computational weights on a log 10 scale over all charged states.

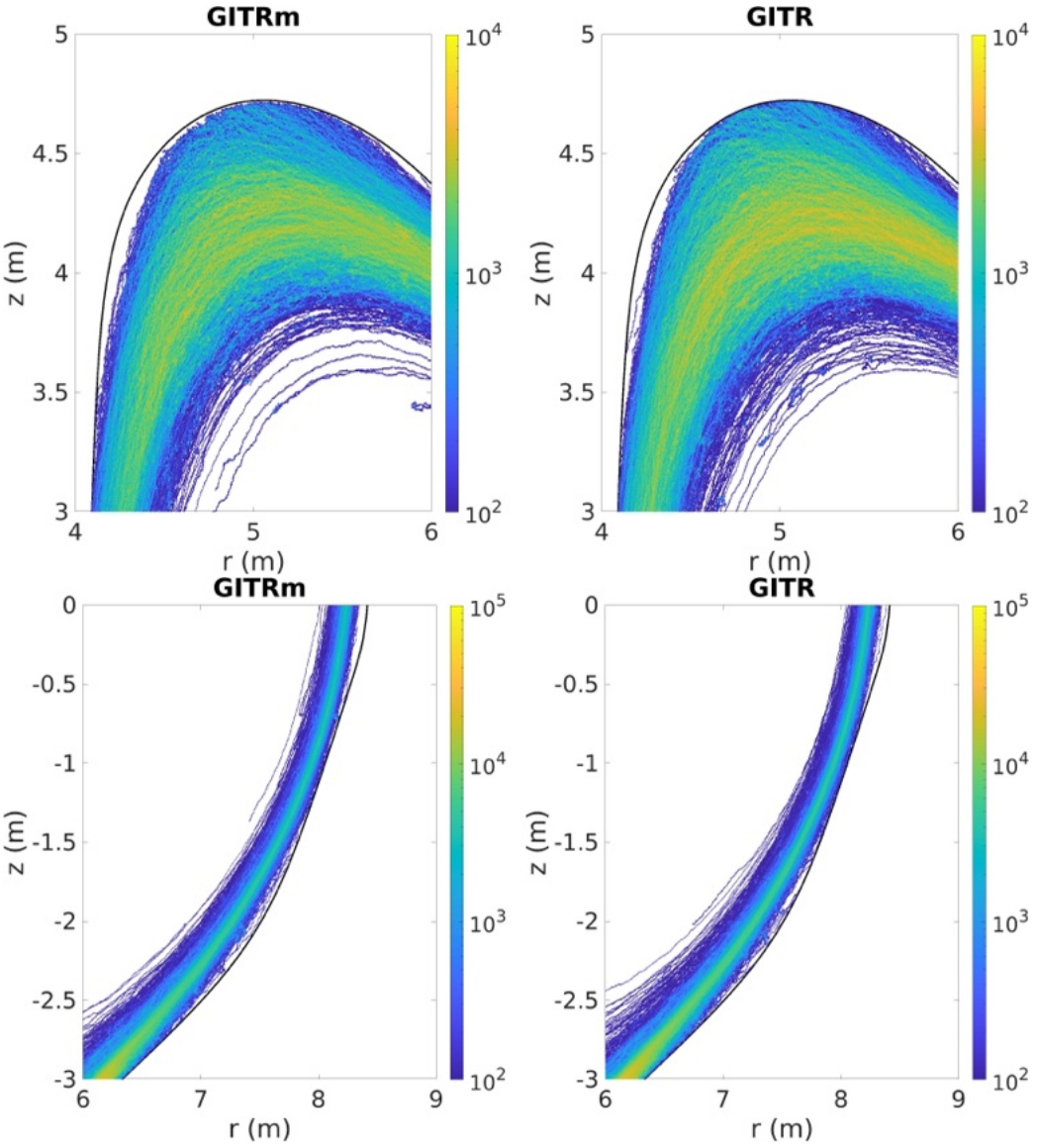


Figure 28: Zoomed views of impurity density near the top (top image) and bottom right (bottom image) of the poloidal cross section.

5.3. DIII-D Case

A cross section of the mesh created for the DIII-D geometry (see Figure 5 in Section 4.1) is shown in the left image of Figure 29. The right image of Figure 29 shows the graded mesh in the collector probe region which

is a region of interest. A comparison of the OEDGE and GITRm meshes are provided at three different locations in Figures 30, 31 and 32. Figures 30 and 31 show these meshes near the separatrix on the lower and upper sides, respectively. Zoomed-in views are also provided for clarity. Figure 30, for the lower side of the separatrix, clearly shows that the GITRm mesh has a similar resolution as the OEDGE mesh along both the normal and lateral/tangential directions to the separatrix. Similar to the ITER case, in GITRm the separatrix is represented as a 3D surface within the volume and is used to generate highly graded and anisotropic elements as layered stacks in a local neighborhood. This is done to capture the sharp directional gradients in the plasma fields. Away from the layered stacks, unstructured (isotropic) tetrahedral elements are used. The layered stack of anisotropic elements, and unstructured tetrahedral elements surrounding them, can be seen in the zoomed-out view (i.e., in the GITRm mesh in the left pair of figures). Figure 31, for the upper side of the separatrix, shows that the GITRm mesh has a similar resolution in the tangential direction and a finer resolution in the normal direction. This is because in GITRm mesh the normal resolution is kept to be a constant along the entire separatrix and is equal to the minimum value of the normal resolution in the OEDGE mesh (i.e., OEDGE mesh has a normal resolution that varies along the separatrix). Similarly, Figure 32 shows the OEDGE and GITRm meshes near the outer target surface, where both meshes have a similar resolution along the normal and tangential directions to the target surface.

Particles are initialized in the part of the tungsten plate that lie to the right of the point where the separatrix intersects the tungsten plate as shown in the right image of Figure 4. Incoming particle flux is used as described in Section 4.3. The simulations are carried out with a fixed time step size of 1×10^{-8} s.

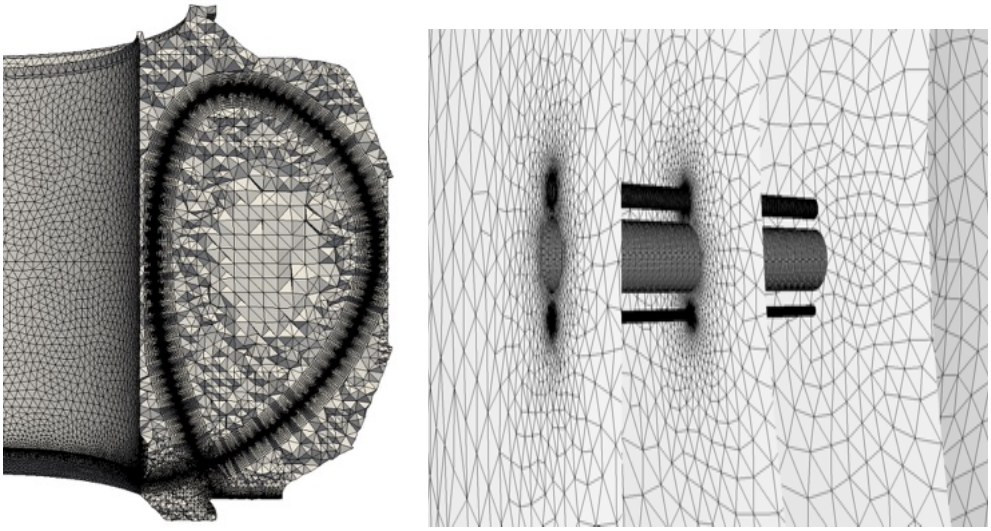
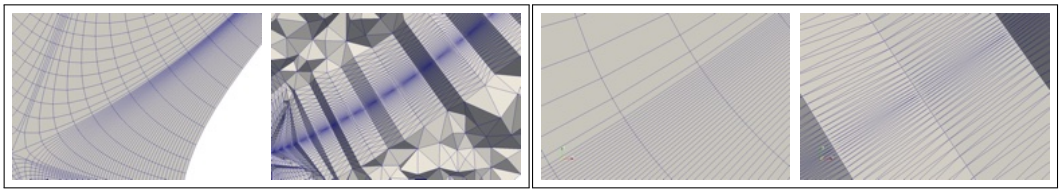
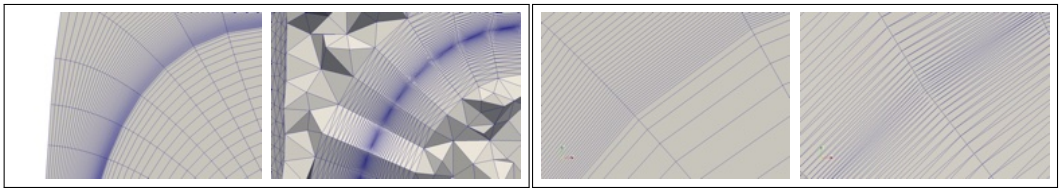


Figure 29: Left: A cross section of the DIII-D mesh. Right: A zoomed-in view of the mesh near collector probes of the DIII-D tokamak.



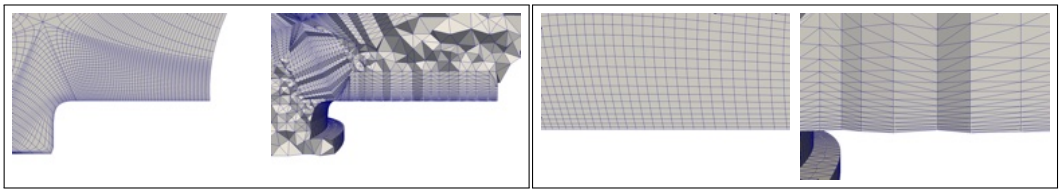
(a) OEDGE (left) and GITRm (right) meshes (b) OEDGE (left) and GITRm (right) meshes

Figure 30: A comparison of the OEDGE and GITRm meshes near the lower side of the separatrix for the DIII-D geometry (the right pair of figures shows a zoomed-in view).



(a) OEDGE (left) and GITRm (right) meshes (b) OEDGE (left) and GITRm (right) meshes

Figure 31: A comparison of the OEDGE and GITRm meshes near the upper side of the separatrix for the DIII-D geometry (the right pair of figures shows a zoomed-in view).



(a) OEDGE (left) and GITRm (right) meshes (b) OEDGE (left) and GITRm (right) meshes

Figure 32: A comparison of the OEDGE and GITRm meshes near the target for the DIII-D geometry (the right pair of figures shows a zoomed-in view).

The three challenges associated in computing particle trajectories far away from the target surfaces include: i) estimation of near-probe plasma conditions, ii) low incidence angle of magnetic field such that it grazes the target surfaces [46], and iii) statistical under representation or sampling of particles in far away regions of interest [47].

The procedures developed in GITRm support fully 3D plasma conditions based on a volumetric unstructured mesh, however, we use plasma conditions from an axisymmetric edge-plasma case since a non-axisymmetric one is not available. A grazing angle of the magnetic field implies that it will take a much larger time for an impurity particle to travel far from the target wall. For example, an impurity particle with an energy of 4 eV and aligned with the magnetic field lines at about 1.5 degrees with the tungsten insert will travel a distance of 0.0536 m vertically up in 100,000 time steps with a constant time step of 1×10^{-8} s, whereas the vertical distance between the collector probe of interest and the tungsten insert in the present geometry is 1.25 m. This is with the assumption that the slope of the magnetic field lines is constant. So the number of time steps required will be of the order of millions to study the effect of impurity particles in regions far from the plasma facing surface.

In addition, only a small fraction of impurity particles strike the collector probes. In the current simulation with 2 million particles, under 10 particles hit the collector probes over 100,000 steps. This leads to statistical under representation or sampling of particles in far away regions of interest [47]. Therefore, in this study, as a computational exercise we also carried out a simulation with the collisional force turned off. With 2 million particles and 100,000 time steps we observe that about 50 particles strike the collector probes. The number of particles impacting the probes can be increased by increasing the total number of particles and/or time steps. For example, a case with 4 million particles and 100,000 time steps, or a case with 2 million

particles and 200,000 time steps, the number of particles striking the collector probes increases to about 120. To achieve approximately 12,800 particles striking the collector probes, another simulation is performed with 400 million particles for 100,000 time steps. The spatial distribution of particles striking the probes is shown in the right image of Figure 33. 48 GPUs are used for this simulation, which is feasible due to relatively large number of particles considered. This is done to demonstrate the computational capability of GITRm to account for fully 3D geometry including small-scale features and efficiently operate on GPU-based parallel computers. It is noteworthy that adaptive particle sampling (e.g., see [48]) is useful to incorporate in such multi-scale problems and must be studied in the future.

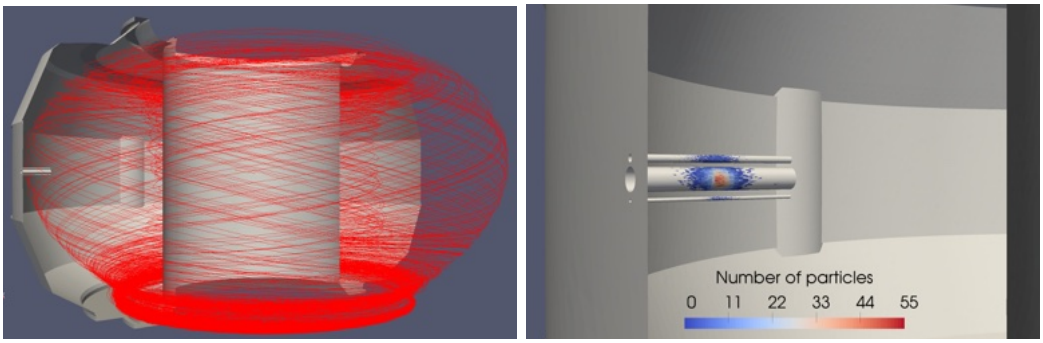


Figure 33: Left: Impurity particle paths. Right: A view of the spatial distribution of the number of particles hitting the probes.

5.4. DIII-D Case - Weak Scaling

To measure the performance of GITRm, a weak scaling study was performed on RPI's AiMOS supercomputer [49] for the DIII-D case from 1 to 24 nodes. For each node, all 6 GPUs are used with one core and one MPI rank per GPU. The simulation is executed for 250 time steps using a range of number of particles from averaging 1.25 million to 10 million per GPU. Figure 34 shows the weak scaling efficiency for each particle count. Weak scaling is defined as how the solution time varies with the number of processes for a fixed problem size per process. Specifically, it is defined as: $\frac{t_1}{t_n}$, where t_n is the time required for n units of work over n process and t_1 is the time required for 1 unit of work over 1 process. Note that a particle load balancing is needed during the simulation as particles evolve, and thus, it is applied as discussed below. Without such a particle load balancing, an

efficiency loss of about 10-50% is observed in the weak scaling. A dashed line is included at 1.00 signifying perfect weak scaling.

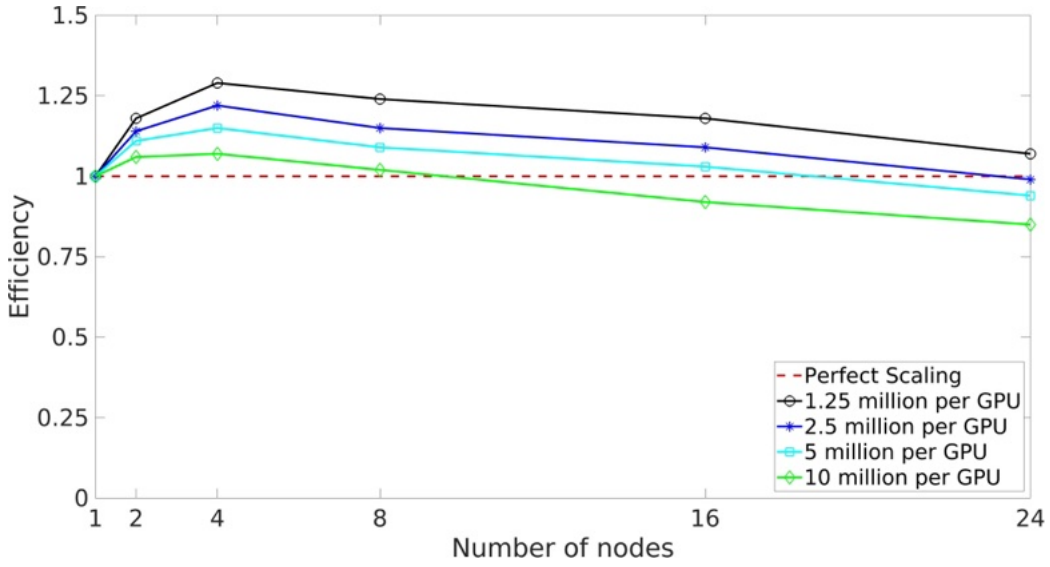


Figure 34: The weak scaling performance of GITRm (higher is better).

For two and four nodes we observe superscaling over 1.00 as a result of the strong scaling of the mesh partition. As the number of nodes increases, the benefits of partitioning the mesh diminish and heavy imbalance of particles caused by the initialization of particles reduces the efficiency. As shown in Section 4.3 particles are initialized on the bottom portion of the mesh, so for the beginning of the simulation only processes with that portion of the mesh will have particles until the particles disperse throughout the domain. This problem reoccurs as particles move higher in the domain leaving the processes with the bottom of the domain without particles.

The problem of particle distribution can be partially alleviated by using particle load balancing. The PUMIPic library takes advantage of the overlap of the mesh partitions in the PICparts to perform particle load balancing by migrating additional particles between processes that share mesh elements [1]. For the DIII-D case, the particle load balancing speeds up the rate the particles migrate to the processes with the higher portions of the domain thus improving the performance throughout the simulation. Figure 35 shows the imbalance of particles for a simulation involving 10,000 time steps, 48 GPUs (eight nodes) and 400 million particles. With particle load balancing turned

on, the particle imbalance is reduced from over 200% down to 5% in the first few hundred time steps and is maintained below 10% for the remaining time steps.

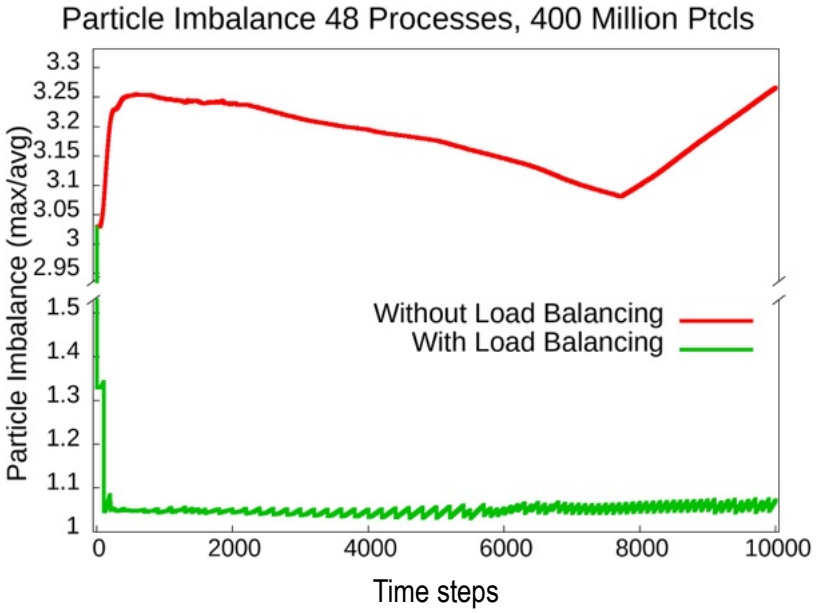


Figure 35: Particle imbalance in GITRm for both with and without particle load balancing (lower is better).

The current load balancing approach is found to be sufficient, however, as more nodes are used and the number of mesh parts increases, the particle distribution problem cannot be solved only using particle load balancing as many processes will be idle in the beginning and later stages of the simulation. A planned development for the PUMIPic library to improve this is to allow groups of processes to copy the same PICpart allowing the particles in that PICpart to be redistributed to any of the processes in the group. This helps the particle distribution problem both by increasing the number of processes working on the heavy concentration of particles and reducing the number of PICparts. Dynamically changing the number of processes assigned to each group during the simulation will allow this approach to follow the changing particle distribution through the entire simulation.

6. Closing Remarks

In this study we presented a fully 3D global impurity transport code, GITRm. It is built on PUMIPic infrastructure to perform the impurity transport on 3D geometries and unstructured meshes in parallel on accelerators, currently GPUs. We showed that GITRm can target complex geometries including non-axisymmetric features such as bumper limiters and collector probes in the DIII-D case. We also showed that GITRm scales with the number of particles simulated as well as the domain and mesh size and geometric fidelity. To demonstrate the effectiveness of the GITRm code, three example cases were used including the PISCES, ITER and DIII-D cases. For the DIII-D case, weak scaling study was also performed with about 1.5 billion particles on up to 144 GPUs and was shown to achieve excellent weak scaling.

7. Acknowledgments

This research was supported by the U.S. Department of Energy, Office of Science, under awards DE-SC0021285 (FASTMath SciDAC Institute) and DE-SC0018275 (Unstructured Mesh Technologies for Fusion Simulation Codes). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Energy.

This research used computational resources of CCI (Center for Computational Innovations) at RPI. This research also used computational resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory and National Energy Research Scientific Computing Center, both of which are supported by the Office of Science of the U.S. Department of Energy.

8. References

References

[1] G. Diamond, C. W. Smith, C. Zhang, E. Yoon, M. S. Shephard, PUMIPic: A mesh-based approach to unstructured mesh Particle-In-Cell on GPUs, Journal of Parallel and Distributed Computing 157 (November 2021) 1–12.

[2] Committee on the Key Goals and Innovation Needed for a U.S. Fusion Pilot Plant, Bringing Fusion to the U.S. Grid (2021), Tech. rep., The National Academies of Sciences, Engineering and Medicine (2021). doi:https://doi.org/10.17226/25991.

[3] E. DÁzevedo, S. Abbott, T. Koskela, P. Worley, S.-H. Ku, S. Ethier, E. Yoon, M. Shephard, R. Hager, J. Lang, J. Choi, N. Podhorszki, S. Klasky, M. Parashar, C.-S. Chang, The fusion code XGC: Enabling kinetic study of multi-scale edge turbulent transport in ITER, in: T. Straatsma, K. Antypas, T. Williams (Eds.), Exascale Scientific Applications: Scalability and Performance Portability, CRC Press, Taylor & Francis Group, 2017, pp. 529–551.

[4] R. Khaziev, D. Curreli, hPIC: A scalable electrostatic Particle-in-Cell for Plasma–Material Interactions, Computer Physics Communications 229 (2018) 87–98. doi:https://doi.org/10.1016/j.cpc.2018.03.028.

[5] K. Madduri, K. Ibrahim, S. Williams, E.-J. Im, S. Ethier, J. Shalf, L. Oliker, Gyrokinetic toroidal simulations on leading multi-and many-core HPC systems, in: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, 2011, p. 23.

[6] B. Wang, S. Ethier, W. Tang, K. Ibrahim, K. Madduri, S. Williams, L. Oliker, Modern gyrokinetic particle-in-cell simulation of fusion plasmas on top supercomputers, The International Journal of High Performance Computing Applications 33 (1) (2019) 169–188. doi:10.1177/1094342017712059.

[7] W. Tang, Z. Lin, Global Gyrokinetic Particle-in-Cell Simulation, in: T. Straatsma, K. Antypas, T. Williams (Eds.), Exascale Scientific Applications: Scalability and Performance Portability, CRC Press, Taylor & Francis Group, 2017, pp. 507–528.

[8] K. Ko, A. Candel, L. Ge, A. Kabel, R. Lee, Z. Li, C. Ng, V. Rawat, G. Schussman, L. Xiao, et al., Advances in Parallel Electromagnetic Codes for Accelerator Science and Development, Tech. rep., SLAC National Accelerator Laboratory (SLAC) (2011).

[9] S. Wiesen, D. Reiter, V. Kotov, M. Baelmans, W. Dekeyser, A. Kukushkin, S. Lisgo, R. Pitts, V. Rozhansky, G. Saibene, I. Veselova, S. Voskoboynikov, The new SOLPS-ITER code package, Journal of Nuclear Materials 463 (2015) 480–484. doi:https://doi.org/10.1016/j.jnucmat.2014.10.012.

[10] P. Stangeby, C. Farrell, S. Hoskins, L. Wood, Monte Carlo modelling of impurity ion transport for a limiter source/sink, Nuclear Fusion 28 (11) (1988) 1945–1962. doi:10.1088/0029-5515/28/11/003.
URL https://doi.org/10.1088/0029-5515/28/11/003

[11] D. Reiser, D. Reiter, M. Tokar, Improved kinetic test particle model for impurity transport in tokamaks, Nuclear Fusion 38 (2) (1998) 165–177. doi:10.1088/0029-5515/38/2/302.

[12] S. Yamoto, Y. Homma, K. Hoshino, M. Toma, A. Hatayama, IMPGYRO: The full-orbit impurity transport code for SOL/divertor and its successful application to tungsten impurities, Computer Physics Communications 248 (2020) 106979. doi:https://doi.org/10.1016/j.cpc.2019.106979.

[13] ERO code, http://www.ero-code.de/ero_old/Manual/index_manual.HTML (accessed 8 January 2023).

[14] J. Romazanov, D. Borodin, A. Kirschner, S. Brezinsek, S. Silburn, A. Huber, V. Huber, H. Bufferand, M. Firdaouss, D. Brömmel, B. Steinbusch, P. Gibbon, A. Lasa, I. Borodkina, A. Eksaeva, C. Linsmeier, J. Contributors, First ERO2.0 modeling of Be erosion and non-local transport in JET ITER-like wall, Physica Scripta 2017 (T170) (2017) 014018. doi:10.1088/1402-4896/aa89ca.
URL https://dx.doi.org/10.1088/1402-4896/aa89ca

[15] T. Younkin, D. Green, A. Simpson, B. Wirth, GITR: An accelerated global scale particle tracking code for wall material erosion and redistribution in fusion relevant plasma-material interactions, Computer Physics Communications (2021) 107885.

[16] D. Ibanez, Omega_h GitHub repository, https://github.com/sandialabs/omega_h (accessed 8 January 2023).

[17] D. A. Ibanez, Conformal mesh adaptation on heterogeneous supercomputers., Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY, Troy, NY (2019).
URL https://dspace.rpi.edu/handle/20.500.13015/1817

[18] B. Wirth, K. Nordlund, D. Whyte, D. Xu, Fusion materials modeling: Challenges and opportunities, MRS Bulletin 36 (3) (2011) 216–222. doi:10.1557/mrs.2011.37.

[19] P. Helander, D. Sigmar, Collisional Transport in Magnetized Plasmas (2005).

[20] J. Drobny, A. Hayes, D. Curreli, D. N. Ruzic, F-TRIDYN: A Binary Collision Approximation code for simulating ion interactions with rough surfaces, Journal of Nuclear Materials 494 (2017) 278–283. doi:https://doi.org/10.1016/j.jnucmat.2017.07.037.

[21] J. T. Drobny, D. Curreli, RustBCA: A High-Performance Binary-Collision-Approximation Code for Ion-Material Interactions, Journal of Open Source Software 6 (64) (2021) 3298. doi:10.21105/joss.03298.
URL https://doi.org/10.21105/joss.03298

[22] J. N. Brooks, Near-surface sputtered particle transport for an oblique incidence magnetic field plasma, Physics of Fluids B: Plasma Physics 2 (8) (1990) 1858–1863. doi:https://doi.org/10.1063/1.859457.

[23] M. Lieberman, A. Lichtenberg, Principles of Plasma Discharges and Materials Processing (2005).

[24] H. P. Summers, The ADAS User Manual, version 2.6, http://www.adas.ac.uk (accessed 8 January 2023).

[25] C. Birdsall, A. Langdon, Plasma physics via Computer Simulation, CRC press, 2004.

[26] J. Neudorfer, A. Stock, R. Schneider, S. Roller, C. Munz, Efficient Parallelization of a Three-Dimensional High-Order Particle-in-Cell Method for the Simulation of a 170 GHz Gyrotron Resonator, IEEE Transactions on Plasma Science 41 (1) (2013) 87–98.

[27] J. Qiang, X. Li, Particle-field decomposition and domain decomposition in parallel particle-in-cell beam dynamics simulation, Computer Physics Communications 181 (12) (2010) 2024–2034.

[28] H. Vincenti, M. Lobet, R. Lehe, L. Vay, J. Deslippe, PIC Codes on the Road to Exascale Architectures, in: T. Straatsma, K. Antypas, T. Williams (Eds.), Exascale Scientific Applications: Scalability and Performance Portability, CRC Press, Taylor & Francis Group, 2017, pp. 375–407.

[29] F. Zhang, R. Hager, S. H. Ku, C. S. Chang, S. C. Jardin, N. M. Ferraro, E. S. Seol, E. Yoon, M. S. Shephard, Mesh generation for confined fusion plasma simulation, Engineering with Computers 32 (2) (2016) 285–293.

[30] Gmsh Team, Gmsh web page, `https://gmsh.info/` (accessed 8 January 2023).

[31] Simmetrix, Simmetrix: Enabling simulation-based design, `http://www.simmetrix.com/` (accessed 8 January 2023).

[32] Open Cascade community Edition, Open cascade web page, `https://github.com/tpaviot/oce` (accessed 8 January 2023).

[33] G EQDSK format, `https://w3.pppl.gov/ntcc/TORAY/G_EQDSK.pdf` (accessed 8 January 2023).

[34] SCOREC, Fusion GitHub repository, `https://github.com/SCOREC/Fusion_Public/wiki` (accessed 8 January 2023).

[35] M. Ulrickson, Limiters and Divertor Plates, Springer US, Boston, MA, 1986, pp. 855–890. doi:10.1007/978-1-4757-0067-1_19.

[36] D. C. Donovan, E. A. Unterberg, P. C. Stangeby, S. Zamperini, J. D. Auxier, D. L. Rudakov, W. R. Wampler, M. Zach, T. Abrams, J. D. Duran, J. D. Elder, A. L. Neff, Utilization of outer-midplane collector probes with isotopically enriched tungsten tracer particles for impurity transport studies in the scrape-off layer of DIII-D (invited), Review of Scientific Instruments 89 (10) (2018) 10I115. arXiv:https://doi.org/10.1063/1.5039347, doi:10.1063/1.5039347.

[37] W. Fundamenski, P. Stangeby, J. Elder, A CFD onion-skin model for the interpretation of edge experiments, Journal of Nuclear Materials 266-269 (1999) 1045–1050. doi:https://doi.org/10.1016/S0022-3115(98)00852-6.
URL https://www.sciencedirect.com/science/article/pii/S0022311598008526

[38] J. Greenwood, The correct and incorrect generation of a cosine distribution of scattered particles for Monte-Carlo modelling of vacuum systems, Vacuum 67 (2) (2002) 217–222. doi:https://doi.org/10.1016/S0042-207X(02)00173-2.
URL https://www.sciencedirect.com/science/article/pii/S0042207X02001732

[39] C. Ericson, Chapter 5 - Basic Primitive Tests, in: C. Ericson (Ed.), Real-Time Collision Detection, The Morgan Kaufmann Series in Interactive 3D Technology, Morgan Kaufmann, San Francisco, 2005, pp. 125–233. doi:https://doi.org/10.1016/B978-1-55860-732-3.50010-3.
URL https://www.sciencedirect.com/science/article/pii/B9781558607323500103

[40] R. W. Hockney, J. W. Eastwood, Computer Simulation Using Particles, CRC Press, 1988. doi:https://doi.org/10.1201/9780367806934.

[41] J. P. Boris, Relativistic plasma simulation-optimization of a hybrid code, Proceeding of Fourth Conference on Numerical Simulations of Plasmas (November 1970).

[42] P. Abreu, L. Silva, J. Pereira, A Distributed Memory Grid Enabled GPU Implementation of the Boris Particle Pusher Algorithm, in: APS Division of Plasma Physics Meeting Abstracts, Vol. 48 of APS Meeting Abstracts, 2006, p. VP1.123.

[43] T. Younkin, D. Green, A. Simpson, B. Wirth, Quantification of the effect of uncertainty on impurity migration in PISCES-A simulated with GITR 264 (2021) 107885. doi:https://doi.org/10.1016/j.cpc.2021.107885.
URL https://www.sciencedirect.com/science/article/pii/S0010465521000369

[44] M. J. Baldwin, R. P. Doerner, D. Nishijima, M. Patino, M. J. Simmonds, G. Tynan, J. H. Yu, A. Založnik, Plasma-Material-Interaction Research Using PISCES Linear Plasma

Devices, Fusion Science and Technology 75 (7) (2019) 664–673. arXiv:https://doi.org/10.1080/15361055.2019.1646608, doi:10.1080/15361055.2019.1646608.

[45] T. R. Younkin, Fusion Machine Scale Wall Material Erosion and Redeposition Modeled with GITR as a Part of an Integrated Simulation to Evaluate Tungsten Plasma Facing Component Lifetimes and Feedback on Fusion Plasmas, Ph.D. thesis, University of Tennessee, Knoxville, Tennessee, Knoxville (2019).

[46] J. Guterl, T. Abrams, C. A. Johnson, A. E. Jaervinen, H. Wang, A. G. Mclean, D. L. Rudakov, W. R. Wampler, H. Guo, P. B. Snyder, ERO modeling and analysis of tungsten erosion and migration from a toroidally symmetric source in the DIII-D divertor, Nuclear Fusion 60 (1) (2020) 6018. doi:https://doi.org/10.1088/1741-4326/ab4c54.

[47] J. Guterl, I. Bykov, R. Ding, P. Snyder, On the prediction and monitoring of tungsten prompt redeposition in tokamak divertors, Nuclear Materials and Energy 27 (2021) 100948. doi:https://doi.org/10.1016/j.nme.2021.100948.
URL https://www.sciencedirect.com/science/article/pii/S2352179121000387

[48] D. Welch, T. Genoni, R. Clark, D. Rose, Adaptive particle management in a particle-in-cell code, Journal of Computational Physics 227 (1) (2007) 143–155. doi:https://doi.org/10.1016/j.jcp.2007.07.015.
URL https://www.sciencedirect.com/science/article/pii/S0021999107003245

[49] Center for Computationial Innovation, AiMOS (DCS) Supercomputer, https://docs.cci.rpi.edu/clusters/DCS_Supercomputer/ (accessed 8 January 2023).