



Modeling and Meshing for Tokamak Edge Plasma Simulations

Usman Riaz^a, E. Seegyoung Seol^a, Robert Hager^b, Mark S. Shephard^{a,1}

^a*Scientific Computation Research Center, Rensselaer Polytechnic Institute*

^b*Princeton Plasma Physics Laboratory*

Abstract

The accurate representation of a problem domain in terms of a geometric model and its effective discretization into a mesh plays an important role in achieving higher-quality simulation results and better computational efficiency. In this work, an automated modeling and meshing infrastructure is developed to produce high-quality tokamak meshes for large-scale fusion simulations based on the X-point Gyro-kinetic Code (XGC). The XGC edge plasma particle-in-cell (PIC) code requires field-aligned meshes with nodes properly placed on the magnetic flux curves and a one-element deep element mesh generated between the flux curves where acceptable and unstructured mesh in areas where the one-element deep mesh is not satisfactory. The procedures that determine areas where one-element deep meshes are not satisfactory are presented, as well as the procedures used to generate quality meshes in those areas. Also, the procedures to provide the mesh information needed for the efficient execution of the XGC computational operations are presented.

© 2011 Published by Elsevier Ltd.

Keywords: Edge plasma PIC simulation, tokamak mesh generation, field-aligned unstructured mesh

1. Introduction

The development of tokamak fusion systems capable of delivering power to the electric grid requires an ability to effectively model the system's plasma physics. A large percentage of the current state-of-the-art plasma physics analysis codes employ a discretization of the domain in terms of a mesh. In many cases, the meshes used are field-following structured meshes that are mapped based on specific physics considerations to effectively account for the anisotropy of the physics being modeled. Although field-following structured meshes support highly efficient calculations that accurately solve the specific physics equations, they require the development of specific mapping methods to distort a regular grid to the field-following coordinates, often contain mapping singularities (e.g., at the O-point) and cannot deal with the geometric complexity of real tokamak systems. On the other hand, unstructured mesh generation techniques are capable of generating graded anisotropic meshes for fully general 3D domains. Specific tokamak physics codes directly use unstructured meshes to effectively address tokamak physics cases. The M3D-C¹ code uses unstructured higher-order finite elements in the poloidal plane for 2D simulations and extrudes those meshes in the toroidal direction to create higher-order wedge elements for 3D simulations [1]. The PetraM/MFEM code uses graded unstructured tetrahedral meshes to perform tokamak radio frequency (RF) simulations, including detailed antenna geometries [2]. The GITRm code uses graded, anisotropic unstructured tetrahedral meshes to perform impurity

*Corresponding Author

Email address: shephm@rpi.edu (Mark S. Shephard)

transport simulations [3]. In the case of the XGC edge plasma particle-in-cell (PIC) code [4, 5], unstructured meshes capable of resolving fully detailed poloidal plane geometries while maintaining a nearly field-following representation in the toroidal direction, are being used. This paper is a follow-up to reference [6], which presented an unstructured mesh generation tool that met an initial set of meshing requirements for M3D-C¹ and XGC. The current paper is focused on recent developments carried out to extend the Tokamak Modeling and Meshing Software (TOMMS) to meet additional meshing requirements of the current XGC gyrokinetic PIC code.

To model plasma physics in the entire tokamak domain XGC employs a 2D triangular mesh defined on a poloidal plane which is then repeated on a set of poloidal planes [4, 5]. To gain the advantage of a field following mesh in the critical portions of the domain, a specific mesh generation process is applied in which the mesh vertices are placed at critical points and on a selected set of flux curves. The placement of the mesh vertices is such that as one traverse from one poloidal plane to the next, the field following the nature of the mesh is maintained. Further, the mesh defined between pairs of closed flux curves is such that no additional mesh vertices are introduced between flux curves until the region between the last closed curve and the separatrix, where additional mesh vertices are introduced to control the mesh quality near the X-point. Similarly, there are additional mesh vertices introduced between pairs of open flux curves, but only in areas close to where those curves intersect the tokamak wall and in regions bounded by a single flux curve and the tokamak wall. General unstructured meshes are generated in such areas where mesh vertices are desired between pairs of flux curves or areas bounded by a single flux curve and the tokamak wall.

To support the ability to automatically generate the desired poloidal plane meshes, the TOMMS mesh generator requires a geometric model that includes a description of not only the tokamak wall but the physics features of the O-point, X-point(s), and selected flux curves. Section 2 presents the procedures used to construct the base poloidal plane geometric model that consists of a set of model vertices, edges, loops, and faces that properly represent the tokamak wall and the desired physics features. The procedures presented in this paper are generalized to support a full range of O-point and X-point combinations, while the previous procedures [6] were limited to specific predefined X-point configurations.

Section 3 briefly summarizes the mesh generation procedures initially developed to construct XGC meshes, the details of which are provided in reference [6]. The resulting procedure is capable of producing good-quality meshes as long as the aspect ratios of elements between pairs of flux curves are not too high. As the XGC code continued to be applied to perform more accurate physics simulations, it became necessary to support mesh elements with high aspect ratios between flux curves. As indicated in section 4, the generation of satisfactory meshes for these cases required further developments with respect to defining the topology of the geometric model being meshed and the application of specific mesh generation procedures on the resulting model faces.

The ability to effectively execute the various XGC computational operations requires providing specifically structured sets of mesh information. Section 5 discusses providing this data and having it ordered for efficient execution.

2. Base Poloidal Plane Geometric Model Construction

The most general input to an automatic mesh generator is a boundary representation of the domain to be meshed in terms of a topological description of the domain in which the shape information is associated with the topological entities. In the case of a poloidal plane cross-section of a tokamak, the topological model is a set of faces where each face is bounded by loops of edges that are bounded by vertices.

To most easily support the TOMMS user community the tokamak geometric model is constructed from two commonly used community inputs. The first input is the discrete description of the tokamak wall. The second is the poloidal flux function (ψ) field provided as the ψ values at the vertices of a background grid that encloses the tokamak. The ψ field, which is provided in the GEQDSK equilibrium file, is used to construct the physics geometric model entities.

A sample computational grid boundary and physical wall curve of the DIII-D tokamak [7] are shown in figure 1a. Figure 1b illustrates the (R, ϕ, Z) tokamak coordinate system used in this work.

2.1. Modeling Tokamak Wall Boundary

The input received for the wall geometry is a set of line segments defined between consecutive points which could be directly used to define the loop associated with the wall either as a single piecewise linear edge with a single model

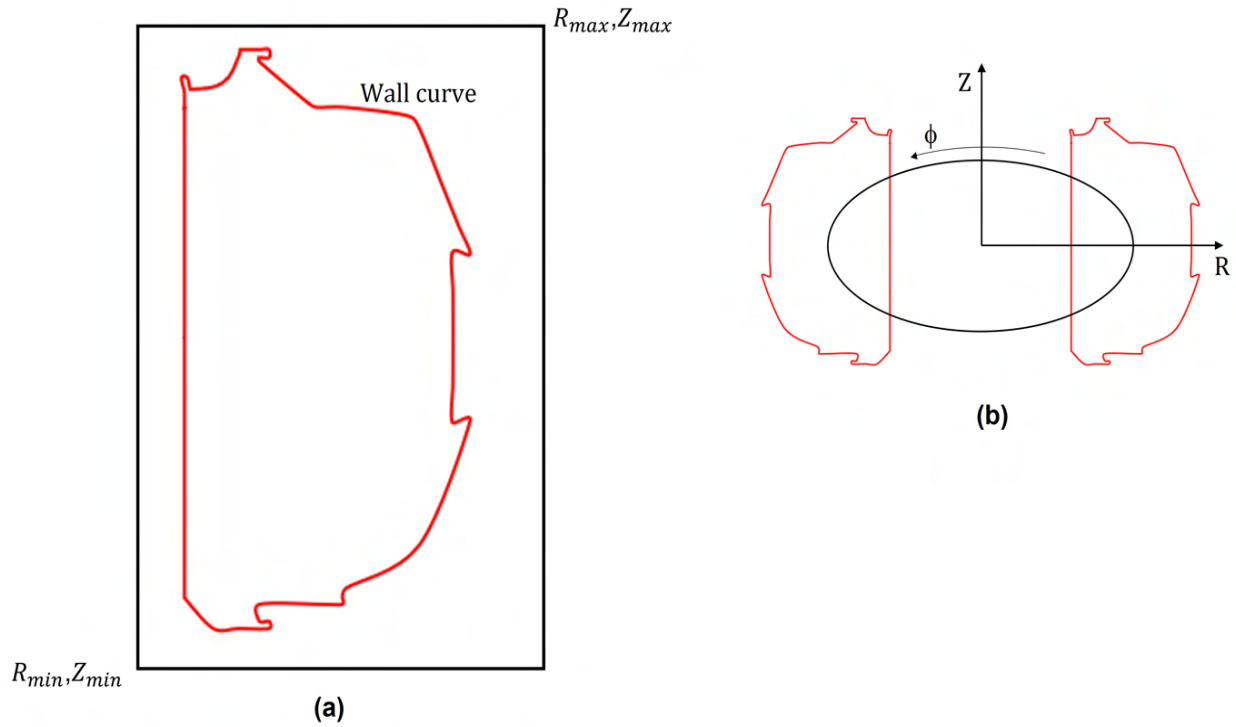


Figure 1. (a) a sample computational grid boundary (black) and actual physical wall curve (red) of a DIII-D tokamak, (b) the (R, ϕ, Z) coordinate system where R is the major radius of the torus, Z is the vertical axis, and ϕ is the toroidal angle

vertex at the starting point or as a set of model edges where each input segment is a model edge bounded by two model vertices. Neither of these options is ideal in that in the first case geometric features like model corners would be lost and in the second case, the mesh generator would be constrained to place mesh vertices at the locations of all of the large set of model vertices. The desired tokamak wall loop would have model vertices only at the appropriate locations that represent “corners” with edges of the appropriate shape defined between those model vertices. Since such information is not provided, the input set of line segments is processed to extract the desired set of model edges as follows:

The procedure begins by placing a model vertex at the start of the first line segment. The next line segment is then examined. If the angle between the two line segments is 180° to within a tight tolerance, the line segment is concatenated to the current edge that is identified as a linear edge. The procedure examines line segments until one is found where the angle is no longer close enough to 180° . A model vertex is placed at the end of the previous segment, and the current segment is considered to be the start of the next edge. When the angle between two line segments is not 180° , a decision needs to be made as to if there will be a model vertex placed between the two line segments or if the segment should be concatenated to the previous segment and considered as part of a curved edge. The procedure for making this decision is based on the algorithm presented in reference [8]. Additional segments are considered part of the curved edge until either consecutive line segments again meet the linear edge criteria at which time they form the beginning of a straight edge, or the angle change is so large that a model vertex is required and then we begin processing the next two line segments to determine if the next model edge is a straight line or a curved edge.

Algorithm 1 provides a pseudo code for the method described above. The results are presented in figure 2. The set of line segments defined in terms of their endpoints can be seen in figure 2b. The set of model edges with their endpoints (red) after processing the points using the algorithm 1 is presented in figure 2c.

Algorithm 1 tokamak Wall Modeling Algorithm

```

1: procedure BUILDWALLCURVE( $P(1:N)$ )                                ▶  $N$  number of given points
2:    $P(1) \leftarrow$  End Point( $E$ )                                  ▶ set first point as end point
3:   for  $i = 2$  to  $N$  do
4:     Evaluate if  $P(i)$  is an End Point( $E$ ), Curve Point( $C$ ) or point on straight line( $S$ )
5:      $M \leftarrow 1$                                              ▶ Initiate  $M$  with 1
6:     if  $P(i) = E$  and  $P(i - 1) = E$  then
7:       Connect  $P(i)$  and  $P(i - 1)$  with a straight line
8:     end if
9:     if  $P(i) = C$  then
10:      if  $P(i - 1) = E$  then
11:        Save  $P(i - 1)$  and  $P(i)$  in Curve
12:      else
13:        Save  $P(i)$  in Curve
14:      end if
15:       $M \leftarrow M + 1$ 
16:    end if
17:    if  $P(i) = E$  and  $P(i - 1) = C$  then
18:      Save  $P(i)$  in Curve
19:       $M \leftarrow M + 1$ 
20:      Fit a spline for  $M$  number of points in Curve
21:      Clear the Curve for the next curve
22:       $M \leftarrow 1$ 
23:    end if
24:    if  $P(i) = S$  then
25:      if  $P(i - 1) = E$  then
26:         $S1 \leftarrow P(i - 1)$ 
27:      else
28:        Ignore the point
29:      end if
30:    end if
31:    if  $P(i) = E$  and  $P(i - 1) = S$  then
32:       $S2 \leftarrow P(i)$ 
33:      Connect  $S1$  and  $S2$  with a straight line
34:      Clear the points  $S1$  and  $S2$  for next straight line points
35:    end if
36:  end for
37: end procedure

```

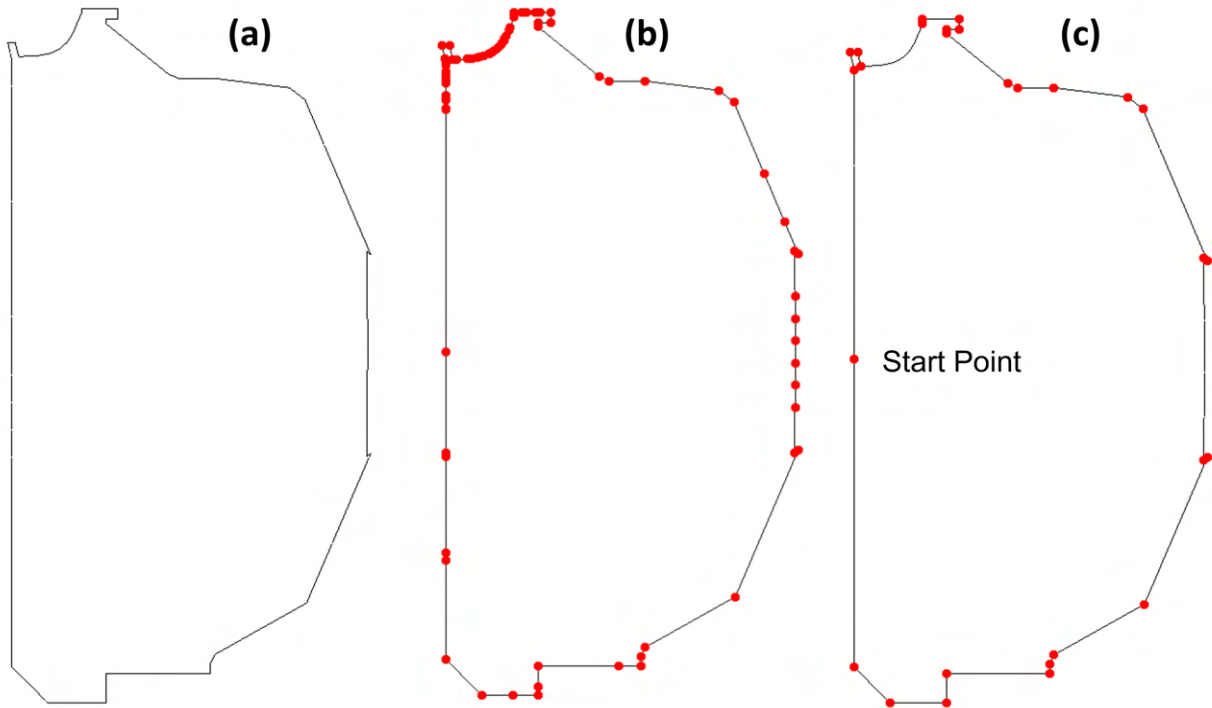


Figure 2. (a) actual wall geometry, (b) set of points before processing, (c) final points used in the model

2.2. Processing Magnetic Flux Function Field Information to Define the Physics Geometry

The poloidal flux function (ψ) field drives the construction of the physics components of the tokamak geometric model that consist of O-points, X-points, and selected flux curves. Figure 3a depicts a set of representative physics model entities generated on the rectangular grid that the flux field is defined over. Figure 3b represents a completed tokamak geometric model that can be given to mesh generation procedures.

The minimum point of the ψ field makes up the magnetic axis (O-point) of the domain shown as $V1$ in figure 3a. Topologically, the O-point is the model vertex with no adjacent model edges. The X-points shown as $V2$ and $V3$ are the saddle points of the ψ field where the magnetic flux curve diverges. The most common tokamak geometric configurations contain one or two X-points. There are three types of physics curves in the domain. The closed magnetic flux curves ($C1$) are the non-interesting curves constructed along the ψ field in the core region ($R1$) of the tokamak. The separatrix curves ($C2$ and $C4$) intersecting at X-points split the domain into different physics regions. There is a separatrix curve associated with each X-point in the domain. Topologically a separatrix curve consists of a set of model edges in such a way that every X-point has three or four model edges adjacent to it. The open flux curves ($C3$) intersect with the wall curves at their starting and ending points along the ψ field, and they do not self-intersect at any point in the domain. The plasma core region ($R1$), which is enclosed by the inner separatrix contains all the closed curves and magnetic axis. The scrape-off layer between the inner and outer separatrix curves marked as $R2$ contains a set of open curves. The magnetic field at the inboard plane (left of the magnetic axis) is stronger than the magnetic field at the outboard plane (right of the magnetic axis) and plays an important role in the definition of physics regions. The physics regions at the outboard are low-field regions, and regions at the inboard planes are high-field regions. The low field side edge region ($R3$) and high field side edge region ($R5$) are the regions containing the open curves outside of the outer separatrix. These regions end with the last open flux curves. The regions between the last open curves and wall curves are defined as near-vacuum regions ($R4$) and ($R6$). The regions bounded by the two legs of separatrix and a wall curve are defined as private regions. Depending upon the Z-coordinates of X-points, one region is defined as a lower private region ($R7$) and the other one as an upper private region ($R8$).

There are three steps involved in constructing the physics geometric model entities from the flux field input of the

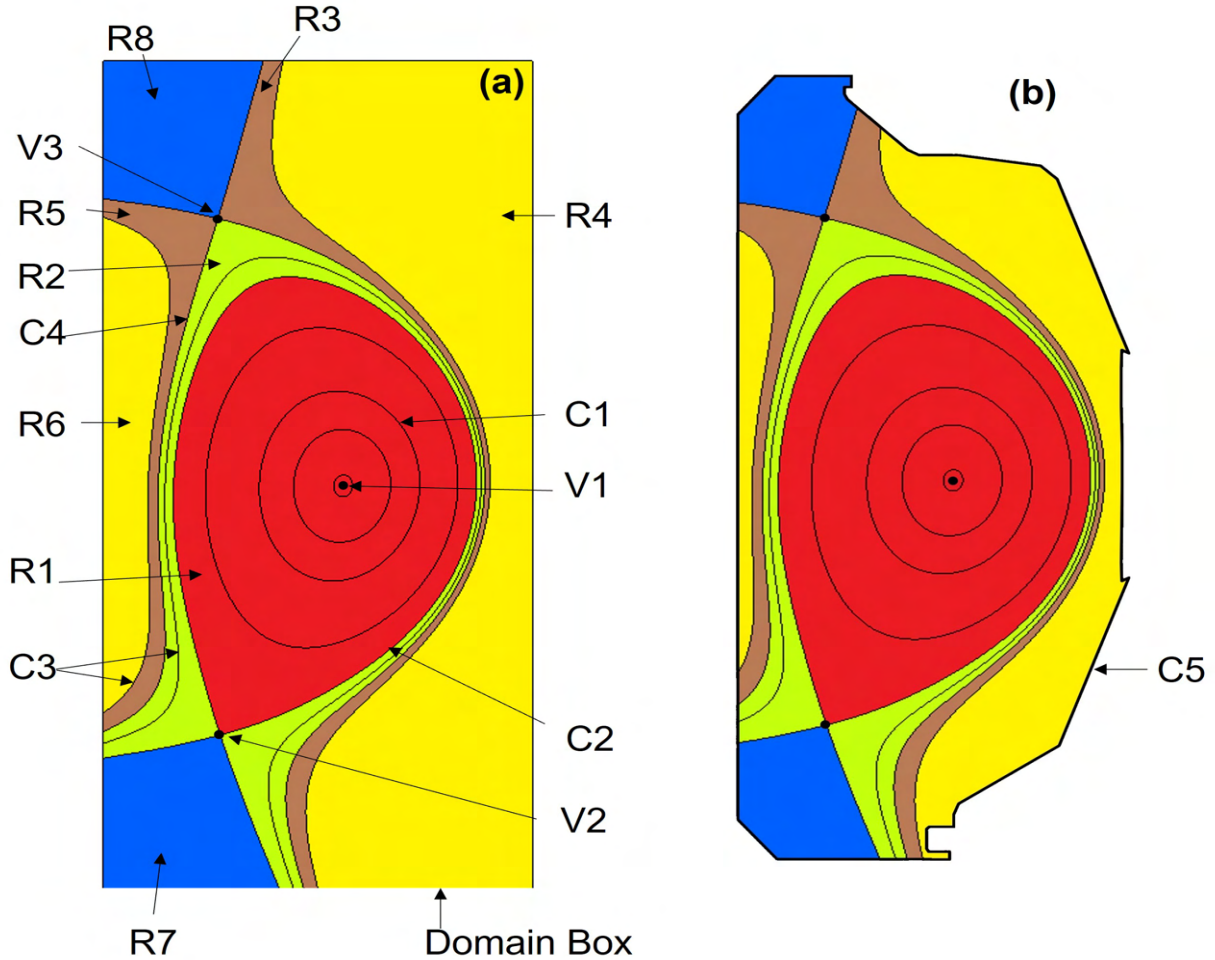


Figure 3. (a) Physics components of TOMMS model. V represents model vertices, C represents curves, and R represents physics regions. V1 = magnetic axis, V2, V3 = X-points, C1 = closed magnetic flux curves, C2, C4 = separatrix curves, C3 = open magnetic flux curves, R1 = plasma core region, R2 = scrape-off layer, R3 = low field side edge region, R4 = near-vacuum region, R5 = high field side edge region, R6 = near-vacuum region, R7 = lower private region, R8 = upper private region, (b) Physics model entities bounded by a physical boundary (wall curve C5).

ψ values at the vertices of the background. The first step constructs a ψ field over the grid that is C^2 continuous to support the operations of the second step, which is to determine the locations of the critical points which are used in the third step of defining the desired set of flux curves.

2.2.1. Discrete Field to Continuous Field

The ψ field is defined in terms of discrete values at the vertices of a background grid. A C^2 continuous bi-cubic spline ψ field is constructed using the PSPLINE routine [9]. Having a C^2 representation effectively supports the procedures used to determine the locations of the O-points and X-points since they employ first and second-derivative evaluations. The bi-cubic spline of ψ is the fundamental shape information defining the geometric shape of the flux curves used as physics geometry in the TOMMS tokamak geometric model.

2.2.2. Critical Point Search Method

The critical points are needed to identify the physics regions in the domain and in the construction of the desired set of flux curves. The O-point, also known as the magnetic axis point, is defined as the minimum of the poloidal flux function ψ . The O-point represents the “center” of the core region that will be surrounded by a selected number of

closed flux curves in the construction of the TOMMS geometric model. The X-points, also known as the magnetic null point, are saddle points of the ψ field. Since there can be multiple minima (only one of which is at the O-point of the core) and multiple saddle points (X-points), an accurate search method capable of identifying all critical points over the domain of the background grid is required. The search method employed is the Downhill Simplex method [10]. The method uses simplex cells, in 2D triangles. References [10, 11] provide a full description of the method. To employ the Downhill Simplex method for extreme point search, the background grid is subdivided into a fine search grid. The local extreme points are found using Newton-Raphson root-finding [12] with the points evaluated by the Downhill Simplex method as initial guesses for the Newton method. To find out if the point is a minimum point (O-point) or saddle point (X-point), the Second Partial Derivative (Hessian (H)) test [13] of the ψ field is used. The determinant of H is defined using:

$$\det H = \left(\frac{\partial^2 \psi}{\partial R^2} \times \frac{\partial^2 \psi}{\partial Z^2} \right) - \left(\frac{\partial^2 \psi}{\partial R \partial Z} \right)^2 \quad (1)$$

- If $\det H < 0$, its a saddle point (X-point)
- If $\det H > 0$ and $\frac{\partial^2 \psi}{\partial R^2} > 0$, its a minimum point
- If $\det H > 0$ and $\frac{\partial^2 \psi}{\partial R^2} < 0$, its a maximum point

The search method returns the number and type of each critical point along with the location of critical points and ψ values at these points.

2.2.3. Flux Curves

The next step is to form the base definition of the flux curves in the domain. Given the fact that we have a continuous definition of the ψ field over the background grid, the base definition of the flux curves will consist of the determination of the ψ value of each flux and separatrix, the topology of the model edge(s) that will represent the flux curves, and the location of model vertices that must be found as part of determining the topology of the edge(s) that represent the flux curves.

The input required for this process is the number of desired flux curves N and a “normalized” $\tilde{\psi}_i$ value for each flux curve where $\tilde{\psi}_i > 0$. The normalization is performed such that $\tilde{\psi}_i = 1$ will corresponds to the first separatrix. Given that, $0 < \tilde{\psi}_i < 1$ will correspond to closed flux curves inside the first separatrix and $\tilde{\psi}_i > 1$ correspond to open flux curves outside the first separatrix. Since a flux curve will be generated for each separatrix the total number of physics curves to be defined is $N_{total} \leq N + N_X$ where N_X is the number of separatrix points found. In the common case of when there is a single separatrix with $\tilde{\psi}_i = 1$ the number of flux curves generated is N . The flux value associated with each of the requested N flux curves is defined as

$$\psi_i = \tilde{\psi}_i(\psi_{sep} - \psi_0) + \psi_0, \quad i = 0, 1, 2, \dots, N - 1 \quad (2)$$

Where ψ_0 and ψ_{sep} are the ψ values at the O-point, and first separatrix curve (X-point), respectively.

After ψ values for the closed flux curves are determined from the input $\tilde{\psi}_i$ values, the next step is to find the starting points of each closed flux curve along the outboard midplane as shown in figure 4. The starting point of each closed flux curve is found by a bisection procedure that iterates until a point on the outboard plane with the required ψ value is found. The model edge defining the closed flux curve is bounded by a single model vertex that is placed on the starting point on the outboard plane as shown in figure 4.

Separatrix curves are the flux curves that are self-intersecting at the X-point and also intersect at the wall curve. There can be multiple X-points in the domain, and each X-point has a ψ value already evaluated from the procedures in section 2.2.2. Note that each separatrix curve is associated with one or more X points. Since the locations of the X-points are already known, the topology of the separatrix curves can be defined by evaluating the intersection of these curves with the wall shown as points i, ii, iii, iv in figure 5. To find these intersections, all the possible points on the wall boundary with the ψ value of the given separatrix curves are evaluated by a traversal of the wall geometry determining locations where the ψ value is that of the flux curve of interest. The model edges defining the separatrix curves are of three types, (a) model edge bounded by a single X-point model vertex as shown in figure 5a, (b) model

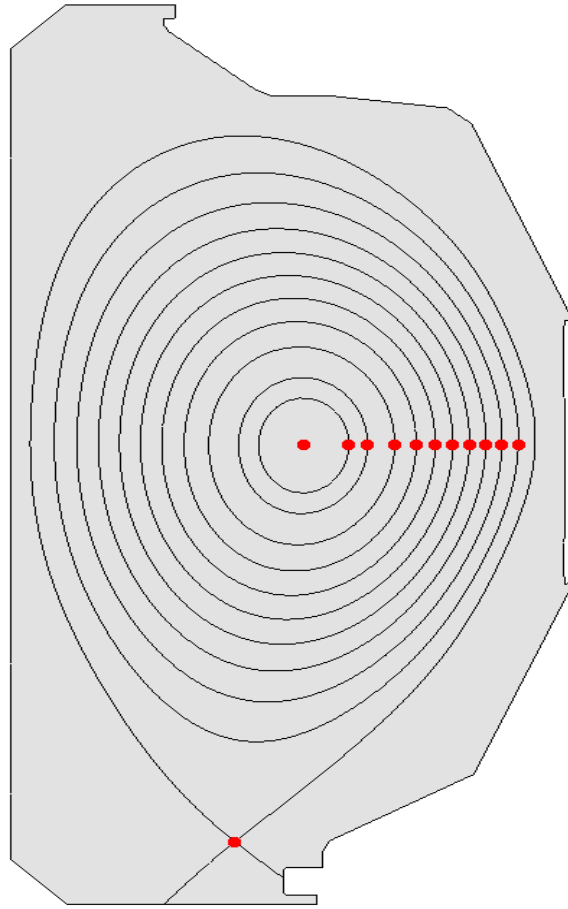


Figure 4. Demonstration of starting points for the definition of closed curves for DIII-D case.

edge bounded by two model vertices, one at X-point and the other one at the intersection of separatrix and wall, and (c) model edge bounded by two X-points model vertices as shown in figure 5c.

Open flux curves are the flux curves that exist outside of the core region ($\tilde{\psi} > 1$) and intersect with the wall boundary. A single flux curve can have multiple intersections with the wall boundary; therefore, an open flux curve can require the use of multiple model edges to represent, as can be seen in figure 6, where one flux curve intersects the wall at six locations, thus defining three model edges bounded by model vertices at the locations where the flux curve intersects the wall. To define the topology and model vertex locations, a method similar to that used for the separatrix-wall intersections is applied.

2.3. Example Models

The physics and physical model entities created using the procedure explained in the preceding subsections produce the base geometric model.

Figure 7 illustrates the models for different tokamak geometries created using procedures explained in this section. The figures 7a and 7b present base models with one X-point and two X-points, respectively, where X-point is shown in the solid black circle. Each color in the figure shows a different physics region with specific meshing requirements. Figure 7c demonstrates a special case of two X-point model, where both the X-points are on the same flux curve defined by the same ψ value.

The procedure for model generation explained in section 2, and the models presented in figure 7 are 2D. The 3D tokamak model, as required in some fusion simulation codes, can be created by rotating the base 2D model for the

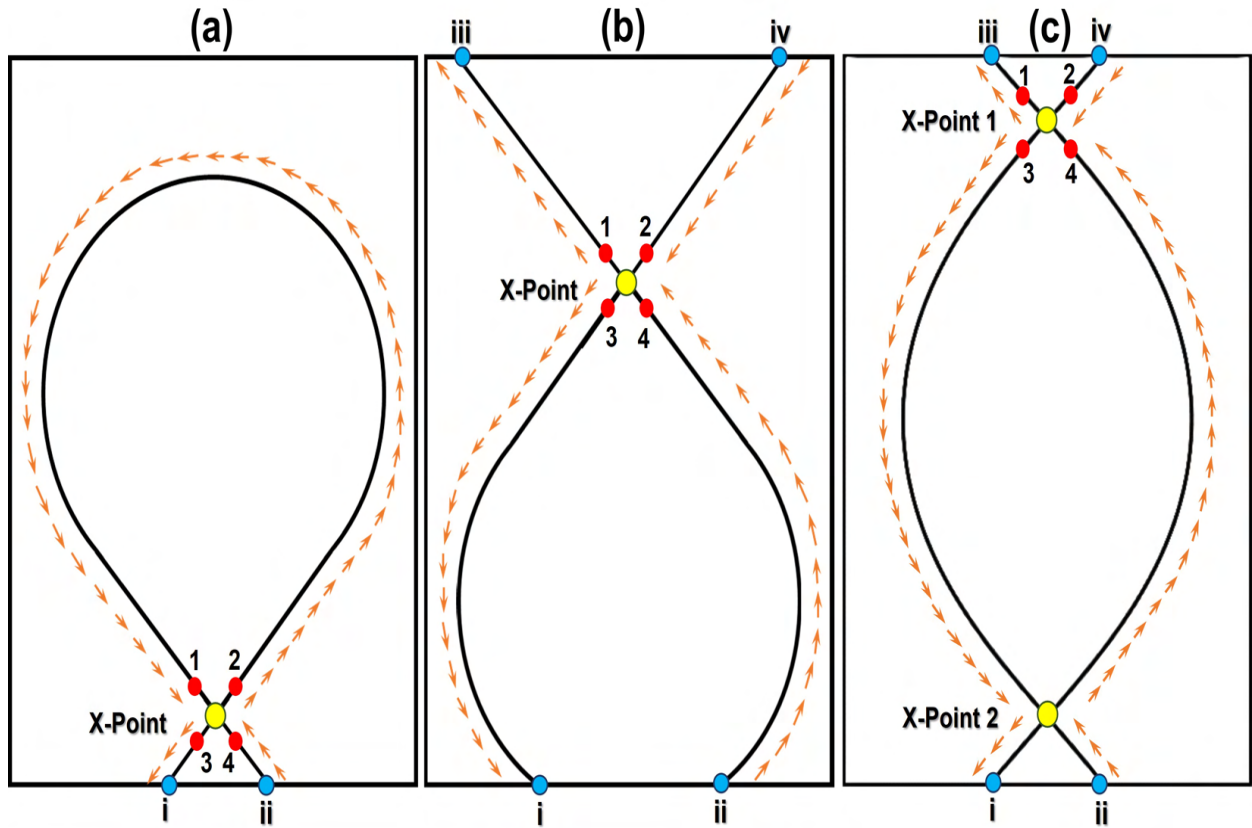


Figure 5. Construction of separatrix curves. (a) inner separatrix curve, (b) outer separatrix curve, (c) separatrix curve with two X-points. Note that in this example the wall geometry is a simple rectangle.

required toroidal angle. Figure 8 presents the cross-sectional view of a 3D model created using 180° rotation of a 2D poloidal base model with one X-point.

3. Base Mesh Generation Procedure

The original mesh generation procedures employed the base geometric model in combination with a near field-following mesh vertex placement procedure and three specific meshing tools [6].

The first step in this process is to place the desired field following mesh vertices on the physics geometry of open and closed flux curves and separatrix curve(s). Each of the model edges representing either all or part of a physics curve has at least one starting model vertex. The model vertex will represent a mesh vertex on that physics curve and is assumed to be the starting point of a field-following curve on the poloidal plane mesh. From there, the field-following procedure given below will determine the location where a field-following point starting at that mesh vertex will intersect the next poloidal plane. This intersection will define the location of a new mesh vertex on the poloidal plane mesh. This process is continued traversing all the poloidal planes until reaching the original poloidal plane. Since the location of the field-following intersection on the original poloidal plane does not match the location of the original vertex, this process continues for a number of toroidal traversals until the intersection on the original toroidal plane is sufficiently close to a mesh vertex that has been defined on that poloidal plane mesh.

The field-following locations of mesh vertices are traced by placing the points on the poloidal cross-section such that a magnetic field line passes through these points at each poloidal plane. The magnetic field \mathbf{B} for a given ψ and poloidal current density $I(\psi)$ is defined as:

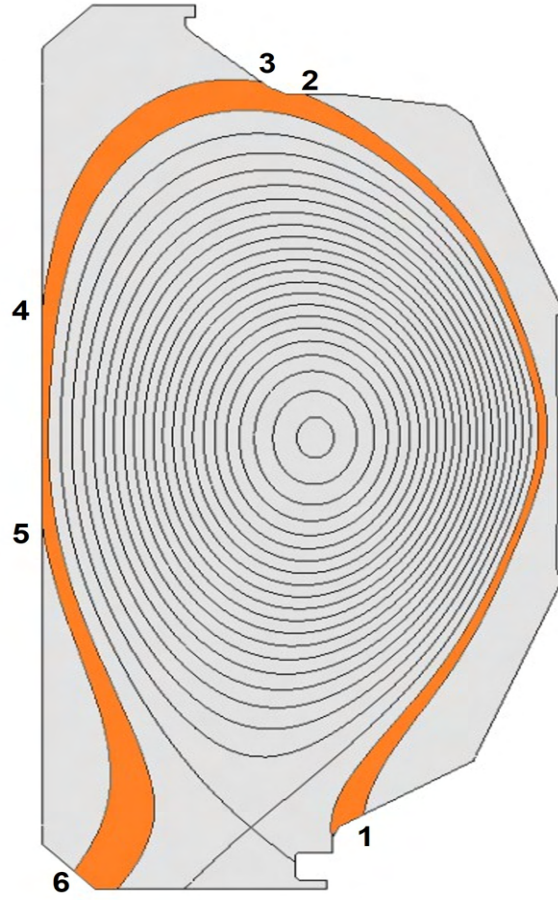


Figure 6. A face on open curves of tokamak cross-section of DIII-D. The inner loop consists of a flux curve with a single edge, and the outer loop consists of a flux curve with multiple edges.

$$\mathbf{B} = -\frac{1}{R} \frac{\partial \psi}{\partial Z} \hat{R} + \frac{1}{R} \frac{\partial \psi}{\partial R} \hat{Z} + \frac{I(\psi)}{R} \hat{\phi} \quad (3)$$

where $\hat{R}, \hat{Z}, \hat{\phi}$ represent the directions along major radius R , vertical axis Z , and toroidal axis ϕ respectively in the (R, ϕ, Z) coordinate system as shown in figure 1b. The position vector of any point on the 3D field line can be expressed in a parametric form $\mathbf{L}(t)$ with respect to parameter t as shown in equation 4

$$\mathbf{L}(t) = [L_R(t), L_Z(t), L_\phi(t)] \quad (4)$$

where its components are expressed in (R, ϕ, Z) coordinates.

If the trajectory of $\mathbf{L}(t)$ follows the field line, then $d\mathbf{L}(t)/dt$ is always parallel to the \mathbf{B} leading to the equation 5 that represents the change in the poloidal magnetic field with the unit change of ϕ .

$$\begin{aligned} \frac{dL_R}{dt} &= \frac{RB_R}{B_\phi} = \frac{\partial \psi}{\partial Z} \frac{R}{I(\psi)} \\ \frac{dL_Z}{dt} &= \frac{RB_Z}{B_\phi} = \frac{\partial \psi}{\partial R} \frac{R}{I(\psi)} \\ \frac{dL_\phi}{dt} &= 1 \end{aligned} \quad (5)$$

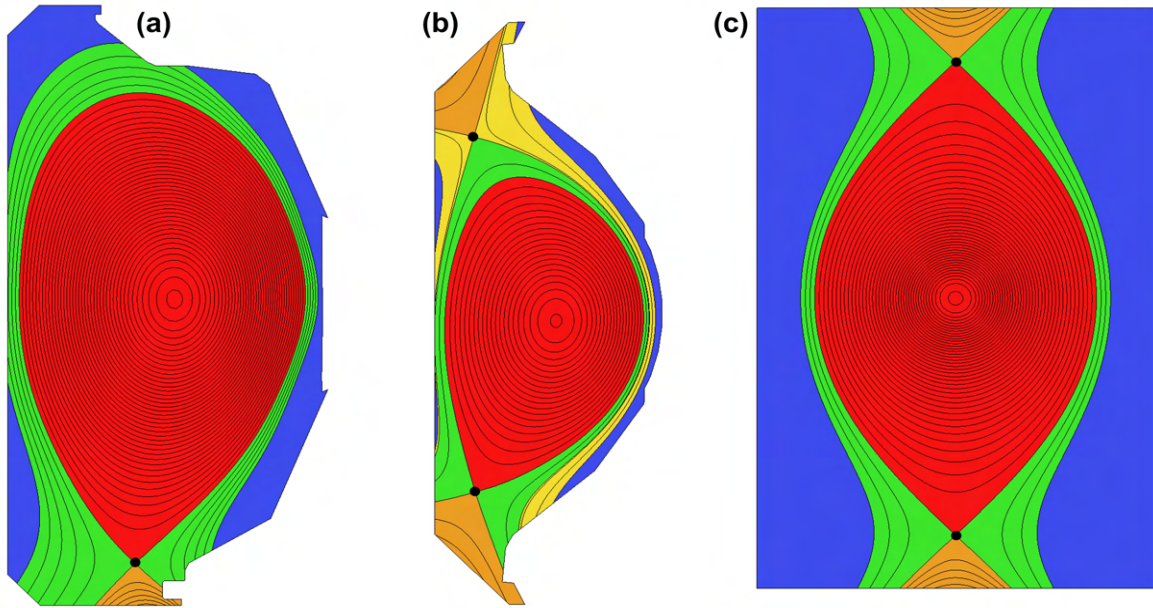


Figure 7. Model examples (a) DIII-D with one X-point, (b) Korea Superconducting tokamak Advanced Research (KSTAR) with two X-points, (c) mock double null equilibrium with two X-points on the same flux curve.

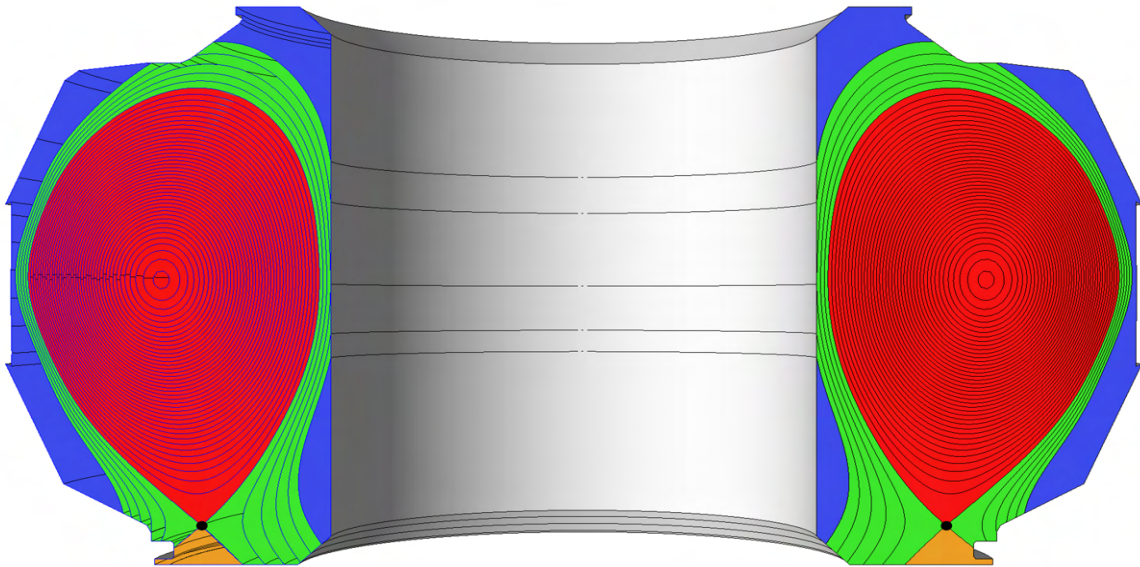


Figure 8. The cross-sectional view of a 3D model generated from a 2D poloidal model with one X-point.

The numerical integration of the magnetic field line in equation 5 using Runge-Kutta 4th Order (RK4) method results in points $[L_R(t), L_Z(t), \phi]$ on the flux curves that are approximately field-following. For every value of ϕ , a point $[L_R(t), L_Z(t)]$ is projected to the flux curve on the original poloidal cross-section.

The mesh vertex placement on the tokamak wall edges is based on the distance between mesh vertices on the flux curves at the model vertex at the intersection of the flux curve with the wall and an input wall maximum spacing. When the spacing on flux curves at the model vertices at the wall intersection is finer than the maximum wall spacing,

the distance between mesh vertices on the wall edge is graded smoothly from the model vertex until reaching the maximum wall spacing.

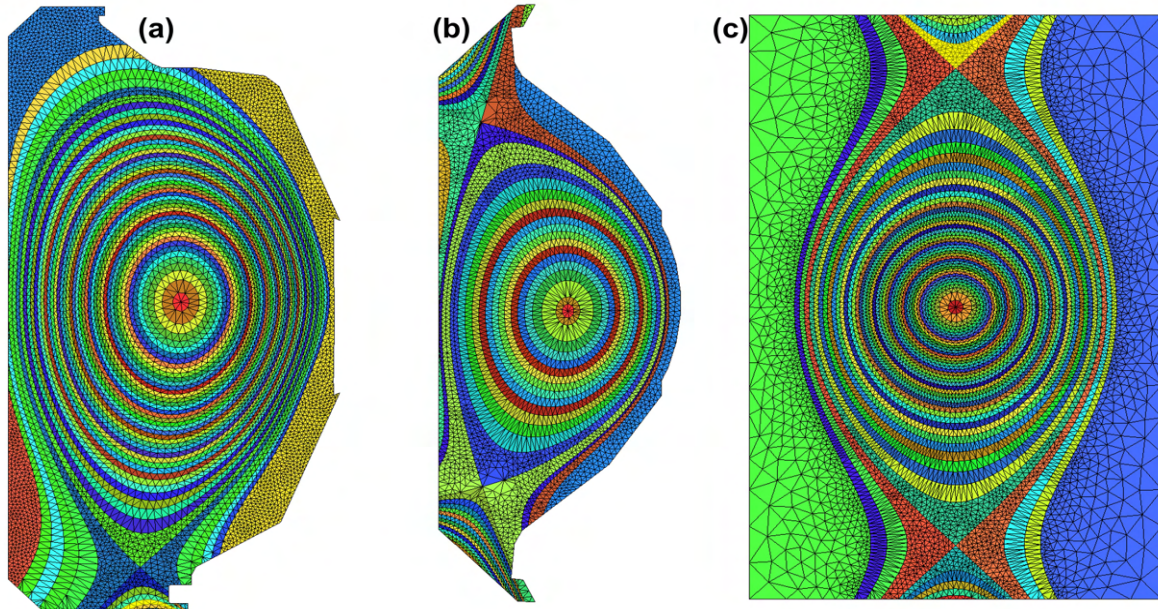


Figure 9. Mesh examples (a) DIII-D with one X-point, (b) Korea Superconducting tokamak Advanced Research (KSTAR) with two X-points, (c) mock double null equilibrium with two X-points on the same flux curve.

The meshes produced at the level of mesh coarseness shown in figure 9 are generally satisfactory. However, the meshes required for the recent XGC production simulations demand a much tighter spacing of mesh vertices on the flux curves relative to the spacing between flux curves. This produces elements with a much higher aspect ratio between the flux curves. Although the base meshing procedure produces the desired mesh in most of the areas between flux curves, the one-element deep meshing procedure produces unacceptable elements near X-points and in areas where open flux curves approach the tokamak walls (see figure 10). In particular, the application of the one-element deep procedure results in the creation of a poor quality mesh in those areas.

Even the application of the local edge split and swap mesh modifications (previously used around the X-points) did not provide a satisfactory mesh, given the high aspect ratio elements between flux curves. To provide the level of mesh resolution and element shape quality desired in those areas of the domain, procedures that modify the geometric model to provide additional model faces for the application of appropriate meshing operations are required. The procedures to update the geometric model and additional mesh generation operations are described in the next section.

4. Model Topology Update and Mesh Generation

Although maintaining the one-element deep field-following mesh between flux curves is important for the area between the closed flux curves and most of the area around the separatrix curve and between open flux curves, the XGC models of the physics behavior near X-points and tokamak walls allows the use of more general mesh configurations. This provides the opportunity to employ alternative mesh generation procedures in those areas that yield higher quality meshes as desired to ensure accurate and stable solution results of the mesh-based PDE solution.

Conceptually, the approach to produce the desired meshes in the problem areas is to decompose the model faces between open flux curves and around X-points into multiple faces. One face, which would consist of most of the original base model face, would continue to employ the one-element-deep meshing procedure. The remaining portions of the original face employ a general-purpose unstructured mesh generator that is suited to producing well-controlled meshes, such as the one that is already applied to model faces bounded by a single flux curve and a portion of the

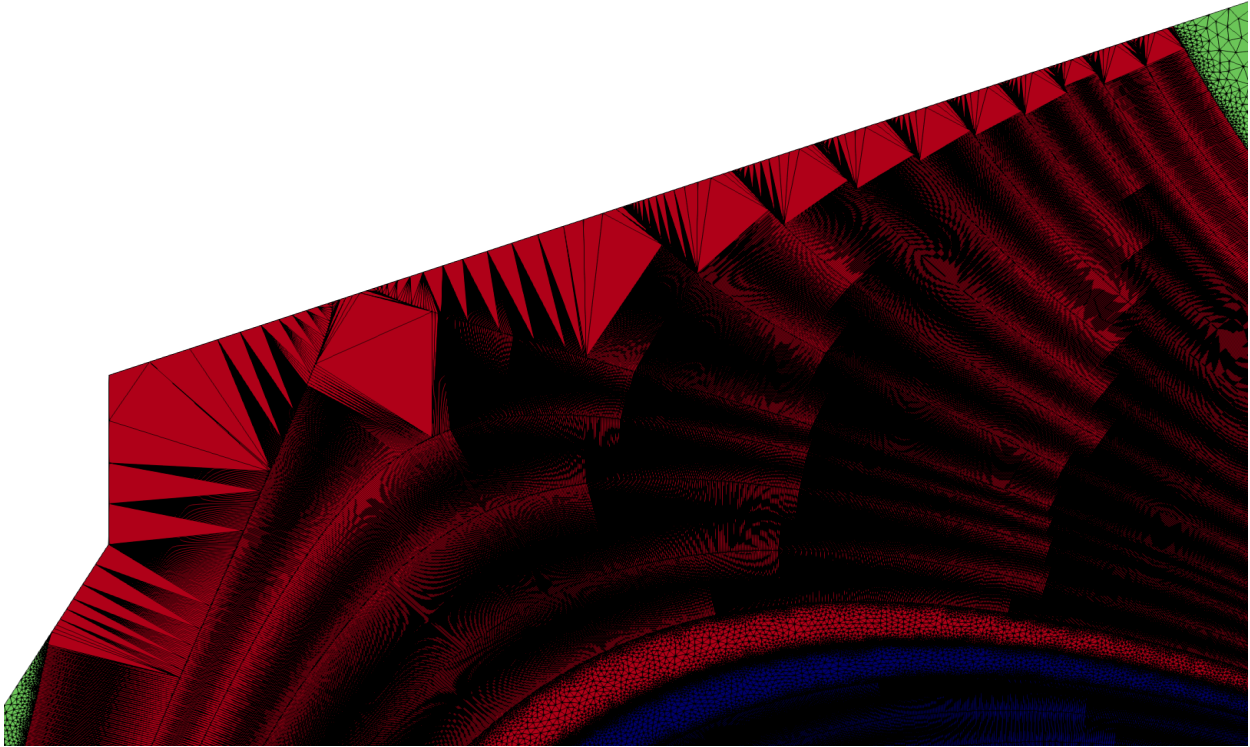


Figure 10. National Spherical Torus Experiment (NSTX) mesh areas with flux curves interacting with the wall curve producing unsatisfactory meshes when applying the original meshing procedures discussed in reference [6].

tokamak wall. The application of this procedure alone is not satisfactory in that it still has the problem of an immediate transition from a high aspect ratio mesh configuration to a more general low aspect ratio element configuration as shown in figure 11. This problem is eliminated by the introduction of a doubling transition procedure that is applied to transition from the high aspect ratio one-element deep elements to a lower aspect ratio unstructured mesh.

4.1. Splitting Model Face between Flux Curves

Given the base poloidal plane geometric model, the set of model faces between two flux curves that include portions of the tokamak wall (figure 12a) and/or X-points (figure 12b) are identified as the model faces that need to be decomposed into a set of model faces in which the one-element deep procedure, the doubling transition procedure (described below), and general unstructured mesh generation are applied. The goal is to maintain the one-element deep structure for as much of the area between flux curves as possible while being sure to transition to lower aspect ratio elements in the specific areas where the mesh would not be satisfactory with the application of only the one-element deep procedure. The key factors in determining the portions of the base model face between two flux curves to which the doubling transition and unstructured mesh procedures need to be applied are a function of the portion of the tokamak wall geometry that is part of the base model face between two flux curves, or how much wider the base model face between flux curves is in the vicinity of an X-point.

The model faces are decomposed into a set of new model faces by introducing model edges across the face between two flux curves. Two new model edges are inserted to split the face into three model face types; one-element-deep mesh face, transition mesh face(s), and unstructured mesh face(s). The first model edge is introduced on the face at a distance ' d ' from the tokamak wall or X-points. The distance d is a function of the length of the tokamak wall on the base model face or the width of the base model face in the vicinity of the X-point for the model faces that are adjacent to X-point. The model edge is introduced on the face by inserting two model vertices v_1 and v_2 on the flux curves ψ_1 and ψ_2 respectively such that the new model edge is bounded by both vertices. This model edge splits the face into two model faces, an unstructured mesh face and a one-element-deep mesh face as shown in figure 13a. The second

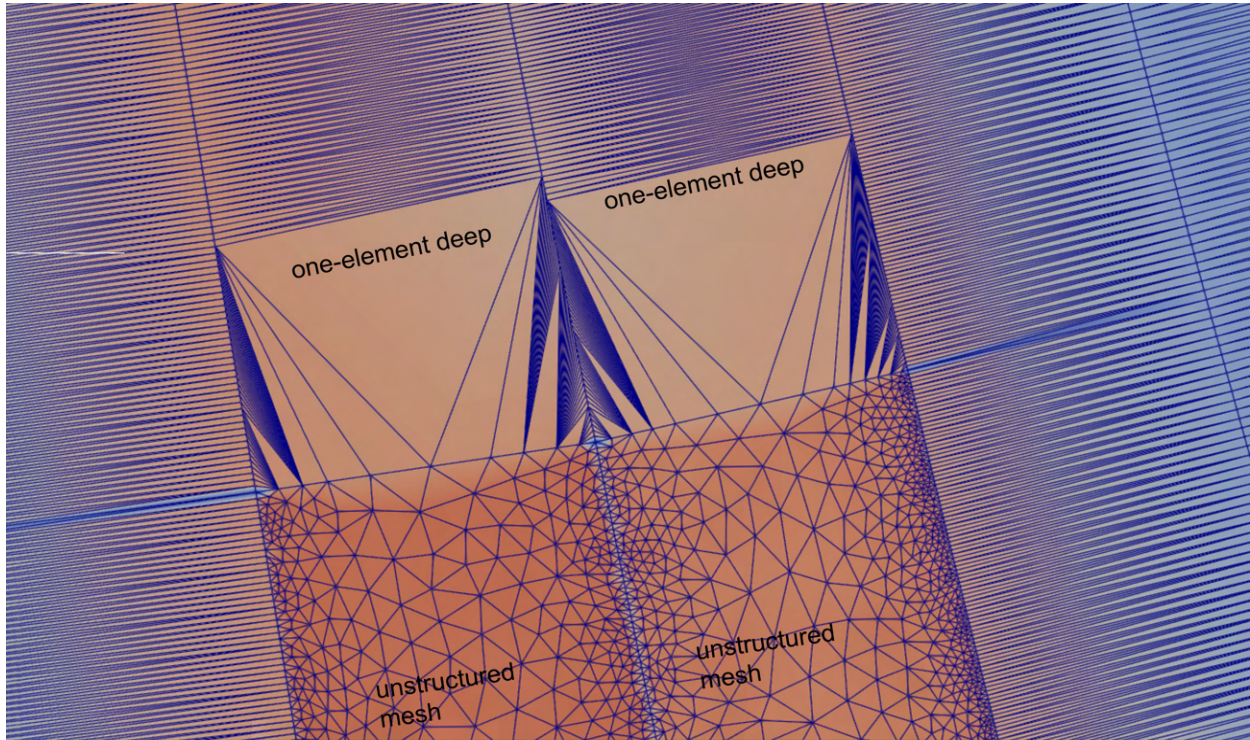


Figure 11. The immediate transition from one-element between flux curves to an unstructured mesh using the given flux curve mesh vertex spacing will yield unsatisfactory meshes, as shown here.

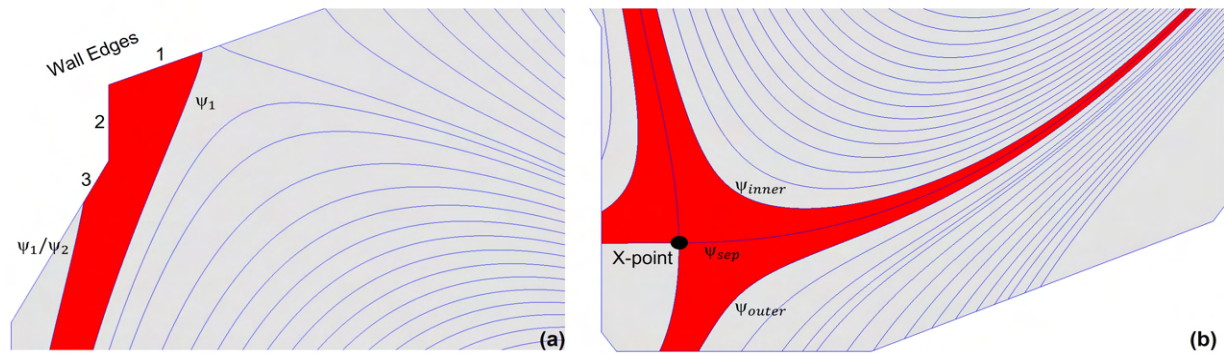


Figure 12. Types of model faces identified for doubling transitioning meshes, (a) a model face bounded by two flux curves ψ_1 and ψ_2 and is adjacent to the wall edge, (b) model faces bounded by two flux curves ψ_{sep} , ψ_{inner} , and ψ_{sep}, ψ_{outer} respectively and are adjacent to the X-point.

model edge is introduced at a distance ‘ d_l ’ from the first model edge, where d_l depends on the number of layers N required for the transition from the highly anisotropic one-element-deep mesh to a more isotropic mesh. The number of transition layers N depends on the approximate aspect ratio (A_r) of mesh elements on the one-element-deep mesh face before transitioning starts and is defined as $N = \log_2(A_r)$. The distance d_l is evaluated by moving N number of field-following points on the flux curve from the starting point (v_1 in the figure 13b). This second model edge is bounded by new model vertices v_a and v_b as shown in figure 13b.

To ensure the field-following property of the flux curves, every new model vertex added to the flux curves is placed at one of the field-following points (already specified on flux curves in the base model) on the flux curve.

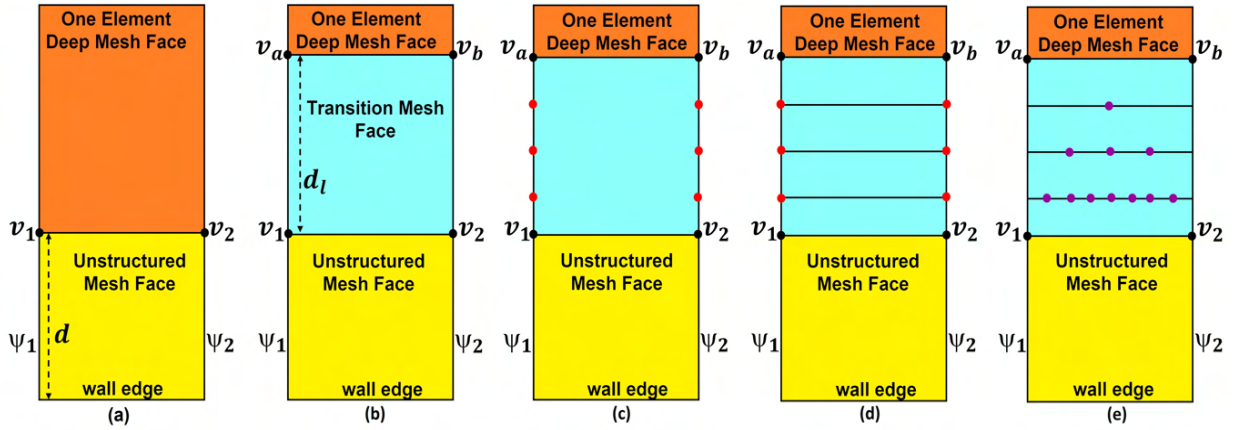


Figure 13. Schematic view of 5 steps of decomposing a based model face into three model faces to be meshed by the one-element-deep, doubling transition, and unstructured meshing procedures. (a) Introduction of the face (yellow) for unstructured meshing, (b) Introduction of the face (blue) needed for the doubling transition mesh, (c) Vertices (red) added to the flux curves, (d) Transition layers added, (e) Vertices (purple) specified on the layers used by the doubling transition procedure.

4.2. Doubling Transition Procedure

The smooth transitioning of the mesh from the high aspect ratio elements produced by the one-element deep meshing procedure to unstructured mesh is executed by applying a doubling transition. The doubling transition is applied as many times as needed to produce a number of mesh vertices between the flux curves that will allow the effective application of general unstructured mesh generation for the remainder of the face between the two flux curves. Figure 13c-13e indicates the process of defining a set of mesh edges and vertices that supports going from the high aspect ratio elements between the flux curves to a number of mesh edges across the face yielding elements with a more acceptable aspect ratio for use by a general triangulation procedure.

Figure 14 shows an example of the process of going from the one-element-deep mesh (area a) to a satisfactory unstructured mesh (area c) by the application of four layers of the doubling transition operator (area b). The number of doubling transitioning layers required for smooth transition depends on the ratio of the distance between the field-following mesh vertices on the flux curves to the distance between two flux curves at the location where the doubling transition operator is applied. The goal is to add the transition layers until the spacing of mesh vertices on the last transition layer is roughly equal to the spacing between the field-following mesh vertices on the flux curves.

4.3. Comparison Meshes

The splitting of model faces between flux curves, followed by the application of the doubling transition operator, aims to produce satisfactory meshes for a full range of desired tokamak geometries and mesh configurations. The base mesh generation procedures presented in reference [6] have limitations in producing meshes of satisfactory quality due to the need for high aspect ratio elements between flux curves for effective XGC simulations. Figure 15 shows a close-up of a portion of a production simulation mesh produced by the base mesh generation procedure compared to the mesh generated using the modifications and extensions indicated in this section.

5. Providing Mesh Information for Efficient XGC Execution

The execution of operations within an XGC simulation requires ordering and structuring the mesh information as needed to support the effective execution of the XGC computations.

5.1. Mesh Data for XGC Mesh Based Operations

The ability of XGC to execute the field-following related calculations, such as flux surface averaging and field-following Fourier filtering, requires specific ordered data related to the set of mesh vertices on each of the flux curves.

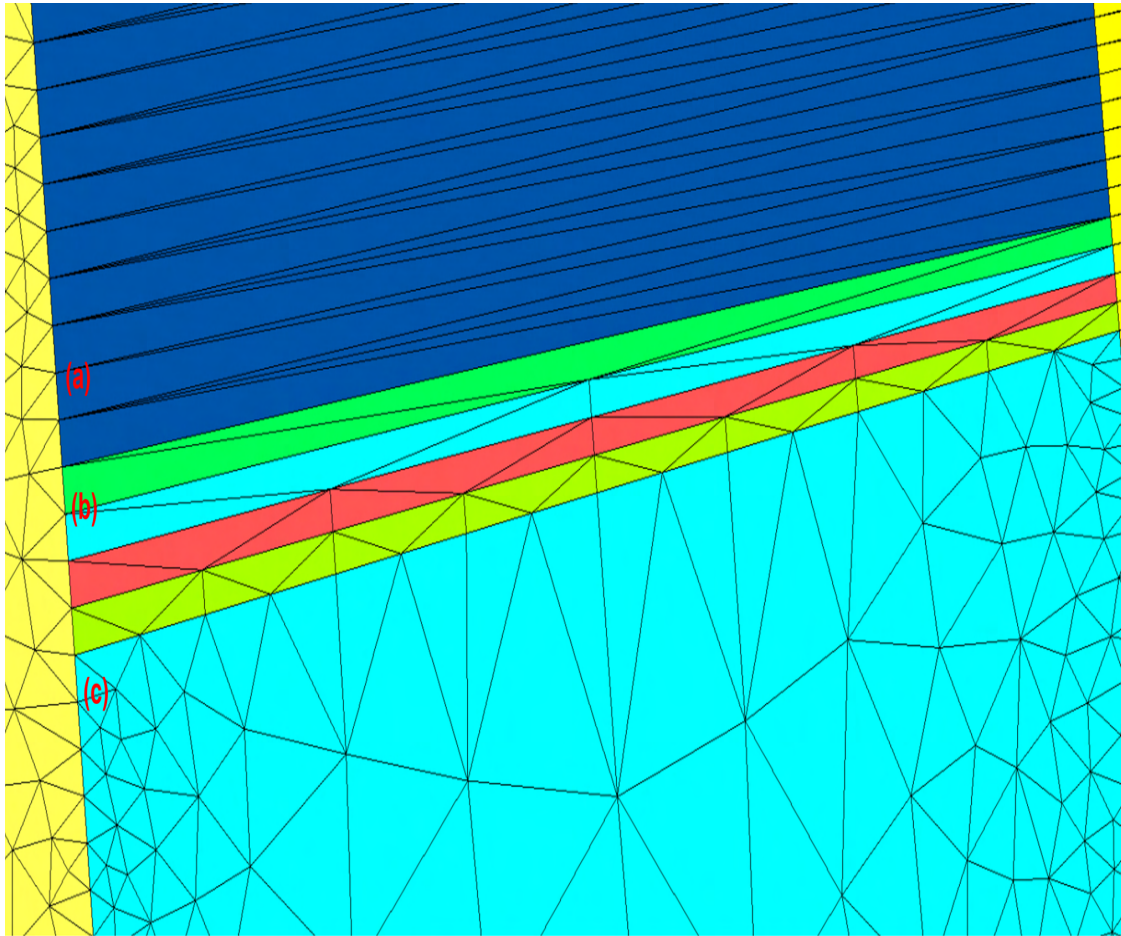


Figure 14. Close-up of the use of the doubling transition mesh (area b), to smoothly transition the mesh from the anisotropic elements in the one-element between flux curves (area a) to the unstructured mesh (area c).

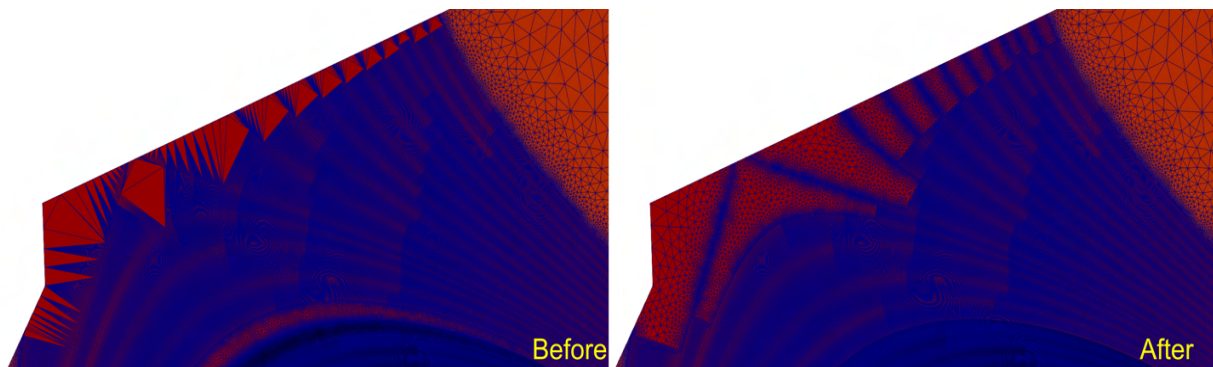


Figure 15. Comparison of previous and current meshing procedures on a National Spherical Torus Experiment (NSTX) production simulation mesh. (Before) re-shows the portion of the mesh from figure 10 meshed with the base meshing procedure, while (After) shows that portion of the domain meshed with the updated meshing procedure presented in this paper.

In addition, other operations take advantage of the one-element between flux curves property of the mesh faces placed between flux curves. As discussed in section 4, the one-element deep property is lost for portions of the mesh between

flux curves adjacent to X-points and for open flux curves as they approach the tokamak walls. In these areas, specific knowledge of the mesh vertices that have been placed between the flux curves to support the generation of acceptable meshes is needed.

The TOMMS mesh generator employs a mesh data structure that maintains explicit knowledge of the association of each mesh entity (mesh vertex, edge, and face for a 2D mesh) with the appropriate model entity (model vertex, edge, and face for a 2D model) [14]. This association information is referred to as mesh classification. The mesh data structure also contains a complete mesh topology so it can efficiently provide information of any mesh entity adjacency (e.g., the mesh vertices bounding a mesh edge, the edges bounded by a mesh vertex, the mesh edges bounding a mesh face, etc.). This information supports the efficient determination of the specific mesh information needed by XGC, which includes:

1. Mesh vertices bounding each mesh face (the standard mesh element connectivity information).
2. Ordered list of mesh vertices on each flux curve as shown in red in the figure 16.
3. Unstructured mesh vertices classified on the model faces that are bounded by two flux curves and also on the model faces that are bounded by a single flux curve and wall curve as shown in blue and green respectively in the figure 16.
4. Mesh vertices on the wall curve as shown in maroon in the figure 16.
5. Mesh vertex classified on the X-point model vertex as shown in orange in the figure 16.

The ordered list of mesh vertices on each flux curve segment is provided using the mesh classification information. The procedure uses a traversal of the mesh edges and collects the set of mesh edges and their bounding vertices on the closure of the model edges that are part of the flux curves. In the case of closed flux curves, the flux curve is represented by a single closed model edge. In this case, the ordered list of mesh vertices is determined by executing a traversal starting at the mesh vertex classified on the single model vertex for that closed model edge. Using mesh vertex to edge mesh adjacency, the mesh edge classified on that model edge using the mesh vertex is found. The other mesh vertex bounding that mesh edge is then the next mesh vertex. The process is continued until the starting mesh vertex is reached. In the case of an open flux curve, the starting point is a mesh vertex classified on a model vertex that is classified on a model edge that is a wall edge. In this case, the traversal terminates when the next vertex found is classified on a wall curve.

The mesh vertices that are classified on the model faces bounded by two flux curves must be collected and associated with the appropriate flux curve. To determine if a model face is between two flux curves, the model edges bounding that model face are traversed, and if two different ψ values are found for the model edges that are part of flux curves in the face loop, then the model face is between those two curves. The mesh vertices classified on the model face between two flux curves are collected and evaluated to determine their association with the flux curve it is closest to.

5.2. Ordering Mesh Data for Efficient Cache Performance

The computational efficiency of numerical operations on GPU-accelerated parallel computers is strongly dictated by the time spent accessing the data required for the execution of stimulation. The “closer” the data is, the more efficient the access will be, with access to data in the various levels of cache memory being orders of magnitude faster than access to data in main memory. Within XGC, the placement of mesh data in memory is based on the labeling that data is given by the mesh generation procedure. Thus, it is desirable to assign labels to the mesh vertices and faces that will optimize memory access.

Due to the field-following aspects of the XGC operations, a potentially advantageous labeling of the mesh vertices and faces is to label the vertex at the core O-point first, and then selecting a mesh vertex on the first flux curve and labeling it followed by labeling the other vertices on that flux curve in order, based on the edge adjacent to the last labeled vertex, until all mesh vertices on that flux curve are labeled. As the mesh vertices on the flux curve are being labeled, the mesh faces that are bounded by the mesh vertices on the flux curve are also labeled. Afterward, the mesh vertices on the next flux curve and the mesh faces between the two flux curves are labeled, starting with a mesh vertex that bounds an edge with the first mesh vertex labeled on the previous flux curve bounds. This process is easily supported for the closed flux curves up to the last closed flux curve before the first separatrix using the classification and mesh adjacency information stored in the mesh data structure used in TOMMS. After that point, the loss of the

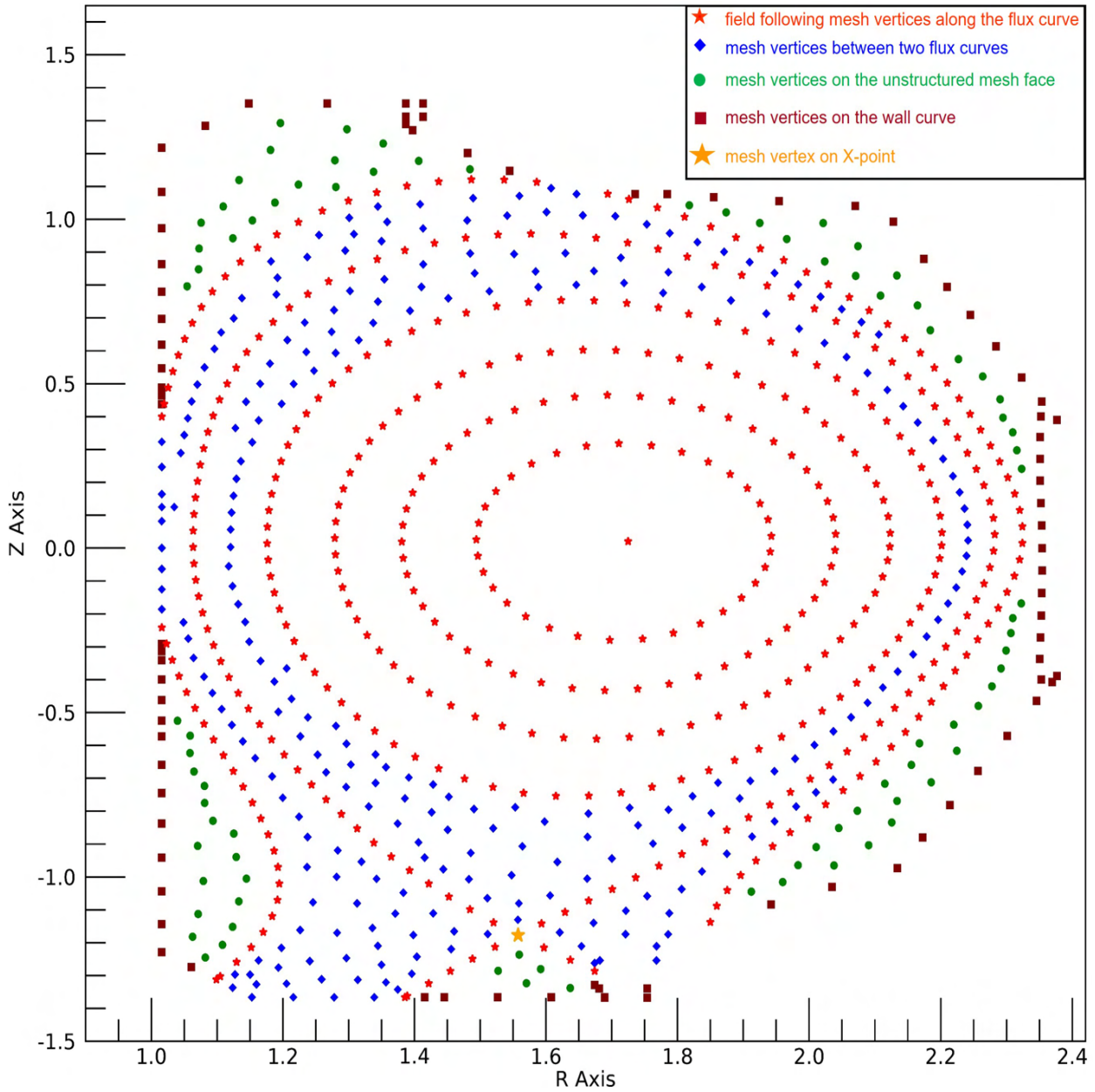


Figure 16. Classification of mesh vertices in XGC

one-element deep between flux curves property, as well as having general mesh between flux curves and wall, means that an alternative method must be applied to label the mesh vertices and faces outside the last closed flux curve before the first separatrix.

The desired labeling of the remaining mesh vertices and faces would continue to follow the strategy of labeling them “spiraling out” from the core O-point. The procedure implemented to do this is built on the adjacency-based ordering algorithm presented in reference [15]. That adjacency-based algorithm employs a queue of the mesh vertices such that the unlabeled mesh vertices on the other end of the edge bounding the recently labeled vertices are put in the queue.

In the application cases given in reference [15], that queue was initiated at the start of the process with a single

mesh vertex on the outer boundary of the domain. To have that algorithm produce the “spiraling out” labeling desired for the XGC meshes, that queue is initiated with the ordered set of mesh vertices on the last closed flux curve before the first separatrix. Since the procedures will label the entities in the queue in order and add unlabeled edge adjacent vertices to the queue, this ordering process will continue to label in a spiraling manner.

Figure 17 shows a coarse mesh example of the resulting reordering of the mesh vertices. Initial testing indicates that this ordering does improve memory access time. However, it had only a marginal effect on total solution time over using the previous less optimized ordering where the ordering of the mesh entities outside the last closed flux curve before the first separatrix was more random. It was determined that there was a small increase in load imbalance that countered the memory access improvements. Additional investigation is underway to consider potential changes to the load balancing scheme so take better advantage of the memory access benefits.

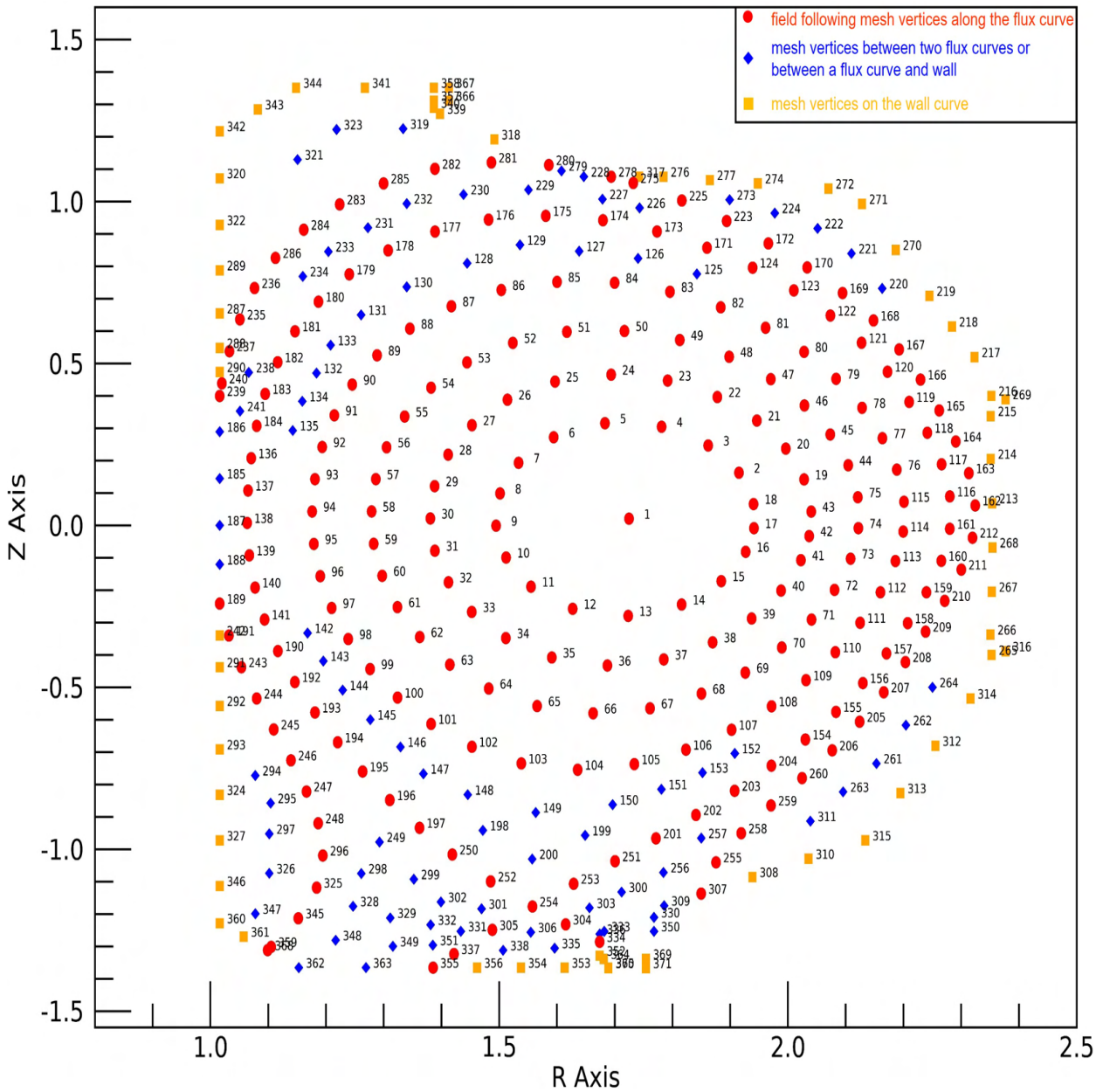


Figure 17. “Spiral ordering” of mesh vertices for a very coarse mesh example

6. Closing Remarks

This paper has presented a set of developments required to advance TOMMS to meet the mesh generation needs of the XGC gyrokinetic particle-in-cell code. The developments carried out include:

1. Extending the tokamak model construction process to support general combinations of O-points and X-points. This includes the development of methods for effective critical point search, modeling of the tokamak wall curve, and the construction of flux curves for a number of tokamak configurations. The critical point search method uses a combination of the Downhill Simplex method and Newton-Raphson method of root finding to ensure the full set of critical points. The wall modeling procedures employ a set of geometric properties to identify the key features of model vertices and model edges that are either straight lines or curves. The flux curve construction methods have been updated to support a general configuration of tokamak geometric features.
2. The mesh generation procedures were extended to support the creation of better-conditioned XGC meshes. These extensions first modified the model topology to split the faces between flux curves that included X-points or portions of the tokamak wall on their boundary into multiple faces where the additional faces were created near the X-point or where the flux curves approached the tokamak wall. As the X-point or wall face is approached, one of the newly created faces is meshed using a doubling transition procedure to go from high aspect ratio elements to low aspect ratio elements. This allowed the effective application of a general triangulation procedure to create a quality mesh in the remaining face that included the X-point or portion of the wall curve on its boundary.
3. Providing the mesh data in a form, and with an ordering, as needed for the efficient XGC PIC-based simulations.

The resulting modeling and meshing tool, TOMMS, is being applied by XGC users to produce the meshes required for critical fusion plasma physics studies.

Acknowledgment

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 authors and do not necessarily reflect the views of the U.S. Department of Energy.

References

- [1] N. M. Ferraro, S. C. Jardin, L. L. Lao, M. S. Shephard, F. Zhang, Multi-region approach to free-boundary three-dimensional tokamak equilibria and resistive wall instabilities, *Physics of Plasmas* 23 (5). doi:10.1063/1.4948722.
- [2] S. Shiraiwa, J. C. Wright, P. T. Bonoli, T. Kolev, M. Stowell, J. Hillairet, Rf wave simulation for cold edge plasmas using the mfem library, *EPJ Web of Conferences* 157. doi:10.1051/epjconf/201715703048.
- [3] O. Sahni, T. Younkin, D. Nath, V. Srinivasaragavan, G. Diamond, C. Smith, A. Hayes, M. Shephard, An unstructured mesh 3d pic code for impurity transport simulation in fusion tokamaks, *Journal of Computational Physics* Submitted.
- [4] E. D’Azevedo, S. Abbott, T. Koskela, P. Worley, S. Ku, S. Ethier, E. Yoon, M. Shephard, R. Hager, J. Lang, J. Choi, N. Podhorszki, S. Klasky, M. Parashar, C. Chang, The fusion code xgc: Enabling kinetic study of multiscale edge turbulent transport in iter, in: T. J. W. Tjerk P. Straatsma, Katerina B. Antypas (Ed.), *Exascale Scientific Applications: Scalability and Performance Portability*, CRC Press, 2017, Ch. 24, pp. 529–552, publisher Copyright: © 2018 by Taylor and Francis Group, LLC.
- [5] S.-H. Ku, C. Chang, M. Adams, J. Cummings, F. Hinton, D. Keyes, S. Klasky, W. Lee, Z. Lin, S. Parker, t. team, Gyrokinetic particle simulation of neoclassical transport in the pedestal/scrape-off region of a tokamak plasma, *Journal of Physics: Conference Series* 46 (2006) 87. doi:10.1088/1742-6596/46/1/012.
- [6] 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.
- [7] J. Luxon, A design retrospective of the diii-d tokamak, *Nucl. Fusion* 42 (5) (2002) 614. URL <https://stacks.iop.org/0029-5515/42/i=5/a=313>
- [8] X. Jiao, M. Heath, Feature detection for surface meshes, *Proceedings of 8th International Conference on Numerical Grid Generation in Computational Field Simulations*.
- [9] Pspline help, <https://w3.pppl.gov/pshare/help/pspline.htm>, online; accessed 24 May 2023.
- [10] J. A. Nelder, R. Mead, A simplex method for function minimization, *Computer Journal* 7 (1965) 308–313.

- [11] H. Press, S. Teukolsky, W. Vetterling, B. Flannery, Minimization or maximization of functions, in: *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992, Ch. 10, pp. 394–444, publisher Copyright: Copyright © 1988-1992 by Cambridge University Press. Programs Copyright © 1988-1992 by Numerical Recipes Software.
- [12] H. Press, S. Teukolsky, W. Vetterling, B. Flannery, Root finding and nonlinear sets of equations, in: *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992, Ch. 9, pp. 347–383, publisher Copyright: Copyright © 1988-1992 by Cambridge University Press. Programs Copyright © 1988-1992 by Numerical Recipes Software.
- [13] J. Stewart, Partial derivatives, in: *Multivariable Calculus*, Cengage Learning, 2010, Ch. 14, pp. 901–996, publisher Copyright: Copyright 2010 Cengage Learning.
- [14] M. W. Beall, M. S. Shephard, A general topology-based mesh data structure, *International Journal for Numerical Methods in Engineering* 40 (9) (1997) 1573–1596.
- [15] M. Zhou, O. Sahni, M. Shephard, C. Carothers, K. Jansen, Adjacency-based data reordering algorithm for acceleration of finite element computations, *Scientific Programming* 18 (2010) 107–123. doi:10.3233/SPR-2010-0301.