

# Mesh generation for confined fusion plasma simulation

Fan Zhang<sup>1</sup> · Robert Hager<sup>2</sup> · Seung-Hoe Ku<sup>2</sup> · Choong-Seock Chang<sup>2</sup> ·  
Stephen C. Jardin<sup>2</sup> · Nathaniel M. Ferraro<sup>3</sup> · E. Seegyong Seol<sup>1</sup> · Eisung Yoon<sup>1</sup> ·  
Mark S. Shephard<sup>1</sup>

Received: 2 June 2015 / Accepted: 8 September 2015 / Published online: 21 September 2015  
© Springer-Verlag London 2015

**Abstract** XGC1 and M3D-C<sup>1</sup> are two fusion plasma simulation codes being developed at Princeton Plasma Physics Laboratory. XGC1 uses the particle-in-cell method to simulate gyrokinetic neoclassical physics and turbulence (Chang et al. *Phys Plasmas* 16(5):056108, 2009; Ku et al. *Nucl Fusion* 49:115021, 2009; Admas et al. *J Phys* 180(1):012036, 2009). M3D-C<sup>1</sup> solves the two-fluid resistive magnetohydrodynamic equations with the C<sup>1</sup> finite elements (Jardin *J comput phys* 200(1):133–152, 2004; Jardin et al. *J comput Phys* 226(2):2146–2174, 2007; Ferraro and Jardin *J comput Phys* 228(20):7742–7770, 2009; Jardin *J comput Phys* 231(3):832–838, 2012; Jardin et al. *Comput Sci Discov* 5(1):014002, 2012; Ferraro et al. *Sci Discov Adv Comput*, 2012; Ferraro et al. *International sherwood fusion theory conference*, 2014). This paper presents the software tools and libraries that were combined to form the geometry and automatic meshing procedures for these codes. Specific consideration has been given to satisfy the mesh configuration and element shape quality constraints of XGC1 and M3D-C<sup>1</sup>.

**Keywords** Geometric model · Automatic mesh generation · Tokamak fusion reactor

---

In memory of RPI Ph.D. Student Benjamin (Ben) Llewellyn Ball whose untimely death did not allow him to complete his work on this project.

---

✉ E. Seegyong Seol  
seols@rpi.edu

<sup>1</sup> Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

<sup>2</sup> Princeton Plasma Physics Laboratory, Princeton, NJ 08540, USA

<sup>3</sup> General Atomics, San Diego, CA 92121, USA

## 1 Introduction

Mesh-based methods are extensively applied to study the behaviors of the plasmas in tokamak geometries [1, 11, 12]. The challenges of meshing for fusion reactor simulations lie in satisfying the specific requirements from the combination of physics of interest and the computational methods applied to model the physics. A fully automatic meshing procedure is needed to most effectively meet specific constraints from different fusion plasma simulation codes.

XGC1 [13] applies the particle-in-cell method to solve the gyrokinetic Vlasov–Maxwell system and it focuses on the physics phenomena at the plasma edge. The motion of particles is tracked and the fields that provide the driving force are computed on the mesh [1–3]. In case of meshes for XGC1, the magnetic flux surfaces form a set of curves on the 2D cross section. Based on the combination of the complex physics and numerical methods used in XGC1 [1–3], the meshing requirements include (i) mesh edges must align with the magnetic flux surfaces, (ii) mesh vertices on the magnetic flux surfaces must be placed in a specific manner [3] to follow the motion of particles through the magnetic field, (iii) the mesh should be one-element deep between adjacent magnetic flux surfaces, (iv) the layered mesh between surfaces needs to be improved at particular areas such as the X-point [14, 15], and (v) the mesh should be generated on the real geometries of fusion reactors.

M3D-C<sup>1</sup> [4–10] studies non-linear magnetohydrodynamic (MHD) instabilities of the plasmas in the tokamak. C<sup>1</sup> finite elements are applied to solve the fourth-order PDEs that arise when a stream function/potential representation for the velocity and magnetic potential vector fields are combined with the MHD equations. The meshing requirements are (i) initial 2D mesh generation on the toroidal cross-section geometry, (ii) mesh adaptation on the

toroidal cross section, and (iii) 3D geometric model and distributed mesh construction out of multiple 2D models and meshes. An initial unstructured mesh on the cross section is generated with the controlled mesh size, and then improved by anisotropic mesh adaptation. The 3D mesh with wedge elements is created by connecting triangular mesh faces on 2D cross sections.

This paper discusses a set of procedures to meet the particular meshing requirements of the XGC1 and M3D- $C^1$  plasma physics codes as follows: Sect. 2 describes the geometry definition in the fusion plasma codes. Section 3 discusses software design and algorithms applied to generate meshes in a controlled manner. Sections 4 and 5 present resulting meshes and the closing remarks, respectively.

## 2 Geometric model definition

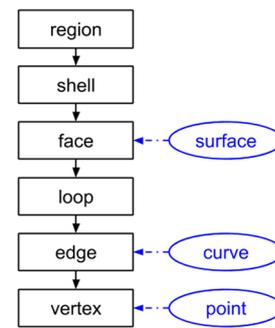
In a geometry-based analysis environment, an effective representation of the analysis domain is a non-manifold boundary topology with associated shape information [16]. A geometric model boundary representation is a hierarchy of regions, shells, faces, loops, edges, vertices, (Fig. 1) and use entities for vertices, edges, loops, and faces [17] that can effectively define the adjacencies seen in analysis model idealizations of physical domains. The geometric model is a necessary input for the reliable automatic meshing [18].

The mesh is a discretized representation of the geometric model. It consists of the four types of topological entities, which are regions (3D), faces (2D), edges (1D), and vertices (0D), with controlled size, shape, and distribution [19–21]. With respect to the simulation, a mesh entity that does not bound any higher dimensional entities is termed an *element*.

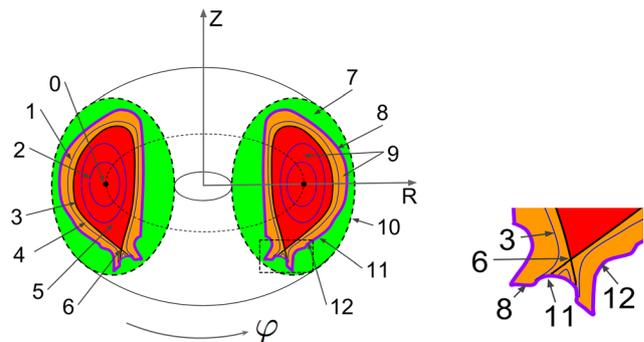
### 2.1 Geometric description

Tokamak devices use a magnetic field to confine the plasmas for sustainable fusion reactions. The basic tokamak geometry is a torus that is symmetric along the toroidal direction. Magnetic field lines wind around the torus and form magnetic flux surfaces [14]. The meshing procedure first generates the 2D geometric model and then generates a 2D mesh based on the geometry.

Figure 2 illustrates the 2D geometric model of the toroidal cross section [14]. In order to comply with different meshing requirements of the plasma simulation codes, the geometry of the tokamak contains the combination of the physics and physical components. In the left of Fig. 2, the physics components are labeled with the numbers from 0 to 6 and physical components are labeled with the numbers from 7 to 12.



**Fig. 1** Topological entities (*rectangles*) and associated shape information (*ellipses*) in the geometric model



**Fig. 2** (*Left*) geometric components of the fusion reactor and (*right*) the close-up of the area near the X-point. The coordinate system is  $(R, Z, \varphi)$  or  $(r, \theta, \varphi)$ , where  $R$ ,  $r$ ,  $\varphi$  and  $\theta$  are major radius, minor radius, toroidal angle, and poloidal angle, respectively. The model components include the (0) magnetic axis, (1) open magnetic flux surfaces, (2) closed magnetic flux surfaces, (3) separatrix, (4) scrape-off layer, (5) plasma core, (6) X-points, (7) vacuum vessel, (8) wall area, (9) plasma area, (10) vacuum boundary, (11) outer wall boundary, and (12) inner wall boundary

A separatrix (3 in Fig. 2) adjacent to the magnetic axis (0 in Fig. 2) splits the face into two distinct areas, the scrape-off layer (4 in Fig. 2) and the plasma core (5 in Fig. 2). The scrape-off layer is the area between the separatrix and the wall boundary. In axisymmetric equilibria of the plasmas, the magnetic flux surfaces (2 and 3 in Fig. 2) in the plasma core are nested toroids, which form non-intersecting loops on the 2-dimensional toroidal cross section between the magnetic axis and the separatrix. The magnetic surfaces diverge at the X-point (6 in Fig. 2) that is a saddle point of the magnetic flux field on the separatrix. The flux surfaces intersect the wall in the area of the scrape-off layer. In addition to the plasma area, the geometry can include a vacuum vessel (7 in Fig. 2) and a finite-thickness wall (8 in Fig. 2). The vacuum vessel is the outermost area of the fusion device surrounding the wall. The plasma area (9 in Fig. 2) is the interior area bounded by the limiter or the

boundary of the material wall [14]. There may exist one or more separatrices in the plasma area.

The geometry of XGC1 contains the plasma area with the scrape-off layer (4 in Fig. 2) separated by the separatrix (3 in Fig. 2) and the plasma core (5 in Fig. 2). It also includes magnetic flux surfaces (1 and 2 in Fig. 2) in the plasma area and the inner wall boundary to meet the requirement of placing mesh vertices. Multiple identical planes are placed around the torus axis.

The geometry of M3D-C<sup>1</sup> is a 3D torus made up of the vacuum vessel (7 in Fig. 2), the material wall with or without a finite thickness,  $\delta_h$ , (8 in Fig. 2), and the plasma area (9 in Fig. 2). Each plane forms a cross section of the tokamak, so multiple planes are placed around the torus axis, each with an identical mesh as is also the case with XGC1. The outer boundary of the vacuum vessel is a simple loop (10 in Fig. 2) that encloses the wall area.

### 2.2 Topological representation

Figure 3 illustrates the topological structure of model faces in XGC1 geometry. Faces interior to the separatrix are bounded by two loops that correspond to a pair of adjacent closed magnetic flux surfaces (faces B and E) or a pair defined by the magnetic axis and the adjacent closed magnetic flux surface (face A). Faces exterior to the separatrix are bounded by one loop that corresponds to a collection of open magnetic flux surfaces and a portion of the wall boundary (faces D, F, G, and H).

The geometry of M3D-C<sup>1</sup> is a torus composed of multiple wedge segments separated by the planes placed around the major axis of the torus. There are up to three model regions between two adjacent planes, which represent the plasma (Fig. 4a), wall (Fig. 4b), and vacuum (Fig. 4c) areas. The simulation domain of M3D-C<sup>1</sup> is either the plasma area or all three areas. Each model region is bounded by a shell that consists of two faces on the two

planes and faces joining two planes. There are up to three model faces on any given plane or cross section, which are bounded by the loops on the boundary of the material wall and the vacuum (Fig. 4).

### 2.3 Shape definition

On the toroidal cross section, the key shape information is the geometry of curves. There are two kinds of curves in the geometric model. The first are physical domain curves that define the inner/outer walls of the reactor and the vacuum boundary. The second are physics curves that define the features interior to the reactor. A physics curve corresponds to a specific magnetic flux surface.

#### 2.3.1 Physical curves

The geometry of the reactor wall curves is defined by either CAD model input, analytic functions, or splines fitted to an ordered set of points. Figure 5 illustrates a wall curve with C<sup>2</sup> continuity by interpolating an ordered set of points with the cubic B-splines [22] in M3D-C<sup>1</sup>.

#### 2.3.2 Physics curves

The intersection of the magnetic flux surfaces and a plane with a constant  $\varphi$  value form the physics curves, which are flux surfaces composed of common magnetic field lines. Magnetic fields in axisymmetric equilibria can be described by the poloidal magnetic flux field,  $\psi = \psi(R, Z)$ , and the field related to the poloidal current density,  $I(\psi)$  [14]. Given  $\psi = \psi(R, Z)$  and  $I(\psi)$ , the magnetic field  $\mathbf{B}$  is defined as [14]

$$\mathbf{B} = -\frac{1}{R} \frac{\partial \psi}{\partial Z} \hat{R} + \frac{1}{R} \frac{\partial \psi}{\partial R} \hat{Z} + \frac{I(\psi)}{R} \hat{\varphi}. \tag{1}$$

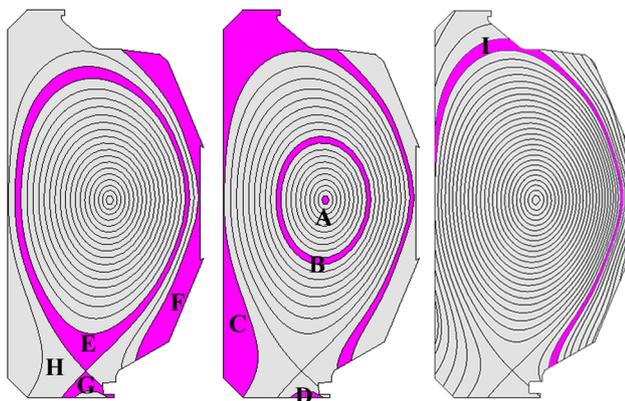


Fig. 3 Geometric faces on the toroidal cross section in XGC1

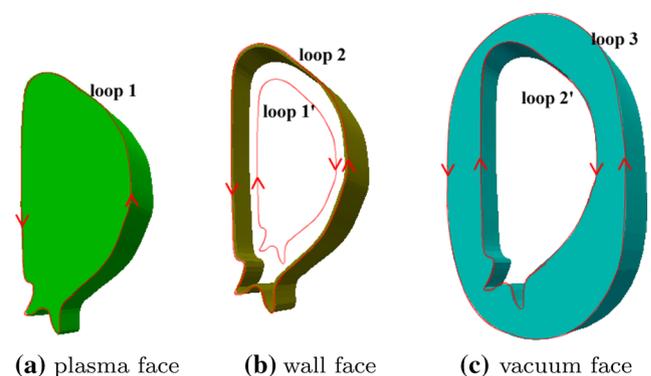
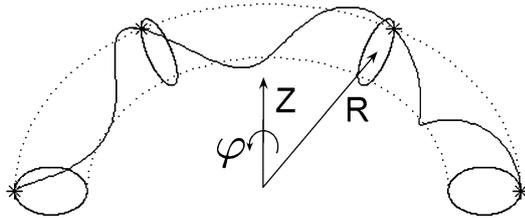
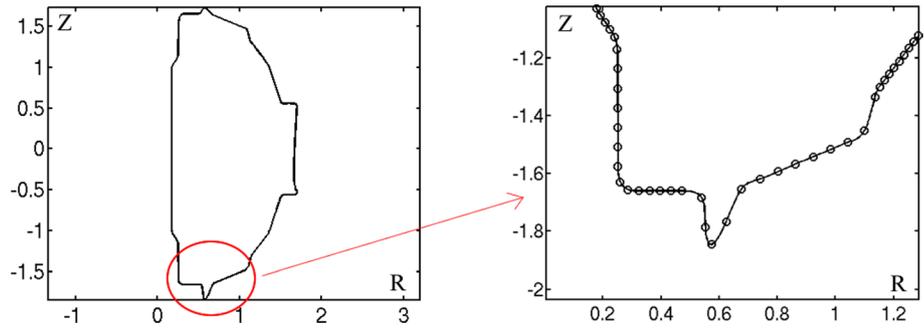


Fig. 4 Geometric faces on the plane (loop 1' of the wall face is offset to be distinct from loop 2)

**Fig. 5** Wall curve of NSTX [23] by the cubic spline interpolation with  $C^2$  continuity.



**Fig. 6** Magnetic field line on a closed magnetic flux surface

The value of  $\psi$  does not change along the direction of  $\mathbf{B}$  from the definition. Therefore, each magnetic flux surface associates with a constant  $\psi$ .

The 3D field lines along the constant magnetic flux surfaces are defined in the parametric form as  $\mathbf{L}(t) = [L_R(t), L_Z(t), L_\phi(t)]$ . Given a set of the constant magnetic flux surfaces,  $\{\psi_i\}$ , the field lines are as follows:

$$\begin{aligned} \frac{dL_R}{dt} &= \frac{RB_R}{B_\phi} = -\frac{\partial \psi}{\partial Z} \frac{R}{I(\psi_i)} \\ \frac{dL_Z}{dt} &= \frac{RB_Z}{B_\phi} = \frac{\partial \psi}{\partial R} \frac{R}{I(\psi_i)} \\ \frac{dL_\phi}{dt} &= 1. \end{aligned} \tag{2}$$

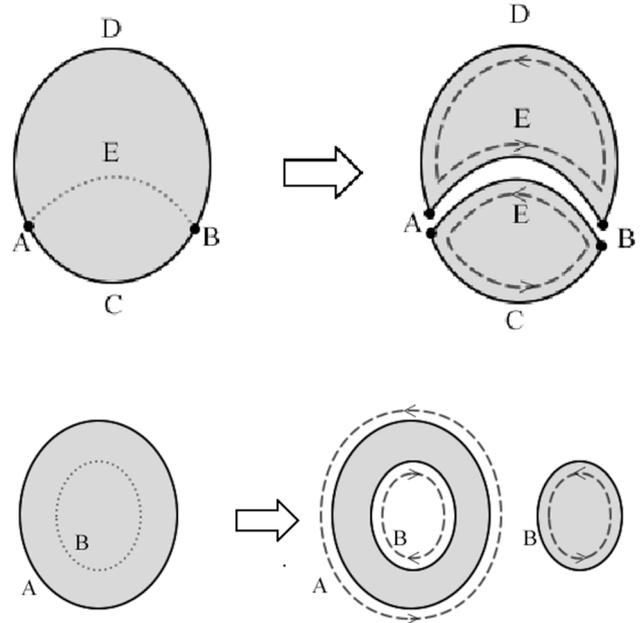
Eq. 2 describes how the poloidal component of  $\mathbf{B}$  changes according to a unit change of  $\phi$ . Figure 6 illustrates the 3D field line on a closed magnetic flux surface.

The ordinary differential equation defined by Eq. 2 is integrated by the Runge–Kutta method on a uniform grid. The two flux fields,  $\psi(R, Z)$  and  $I(\psi)$ , are defined by fitting the experimental data with splines [24, 25].

Given the definition of magnetic field line in Eq. 2, the physics curves are obtained by replacing  $L_\phi(t)$  with a constant value.

### 2.4 Geometric model construction

Figure 7 illustrates two basic topological splits that are applied iteratively to construct the geometric model on the toroidal cross section. A model face is either split by the



**Fig. 7** Basic topological splits (the loops on the right side are shown with an offset)

model edges that connect two model vertices on the boundary of the model face or by the model edges that form a loop interior to the model face.

If the model face is to be split by a list of model edges associated with a magnetic flux surface, the value of  $\psi$  is specified to create the curves associated with the model edges. The value of  $\psi$  between adjacent magnetic flux surfaces is changed by  $\delta\psi$  that is specified by the user. The exceptional case is at the magnetic axis point where  $\frac{d\psi}{dr} \sim 0$ . The new value of  $\psi$  near the magnetic axis point is determined based on the physical distance between the adjacent flux surfaces.

### 3 Meshing procedure

In the automatic mesh generation procedure, mesh control parameters are specified onto the entities of the 2D

geometric model defining a cross section. The purposes of mesh control parameters are (i) meeting the mesh layout constraints, (ii) having the desired mesh gradation, (iii) controlling mesh quality, and (iv) meeting the needs of the simulation procedure. The full set of mesh control parameters includes

- $d_i$  to control spacing mesh vertices on a magnetic flux surface.
- $\delta\psi$  to control spacing of the magnetic flux surfaces.  $\delta\psi$  specifies the change in the values of  $\psi$  between adjacent magnetic flux surfaces. The number of magnetic flux surfaces used is proportional to  $\frac{\psi_{\max}-\psi_{\min}}{\delta\psi}$ , where  $\psi_{\max}$  and  $\psi_{\min}$  are the maximum and minimum values of  $\psi$  in the domain.
- Mesh size control on the model entities.
- Element shape control that determines the desired shape of mesh elements.
- Anisotropic mesh size field that drives the mesh adaptation.
- The mesh generation procedure consists of four software components: (i) unstructured triangulation, (ii) layered mesh generation, (iii) general mesh modification, and (iv) toroidal mesh extrusion of the 2D mesh to create a 3D mesh of the full reactor.

### 3.1 Unstructured triangulation

Unstructured triangulation creates a graded mesh in the portion of domain where there is no need to generate a one-element-deep mesh. A number of tools are available to generate unstructured triangular meshes. The Symmetric *MeshSim* [26] is used in mesh generation for XGC1 and M3D-C<sup>1</sup> due to its benefits of well-controlled graded meshes based on a geometry-based specification and the ability to be incorporated as a component of an overall meshing procedure responsible for meshing selected portion of the domain. In case of XGC1, *MeshSim* is used to mesh the geometric faces between the open magnetic flux surfaces and the wall boundary (faces D and F, and part of face C in Fig. 3). In case of M3D-C<sup>1</sup>, it is used to create an initial mesh of the entire 2D domain.

### 3.2 Layered mesh generation

In XGC1, the motion of particles is driven by the electromagnetic field and the particle orbits mainly follow the magnetic flux surfaces in axisymmetric equilibria of the plasmas since the magnetic field perturbation is small. The motion of particles changes the charge and current density that determines the field by the gyrokinetic Poisson and Ampere’s equations [1, 3]. Meshes following the equilibrium field lines improve the efficiency of parallel particle

tracking, and the accuracy of derivative calculation along the field line. Therefore, in case of the mesh in XGC1, mesh vertices and edges on the model edge and interior area need to follow the field lines associated with the magnetic flux surfaces. The constant electromagnetic turbulence contours also follow the equilibrium magnetic field closely. Thus, the meshes following magnetic field line also improve the efficiency of the field solvers.

The vertex placement on the toroidal cross section is as follows. Assume the number of planes placed around the major axis of the torus is  $n$ . For the magnetic flux surface with  $\psi = \psi_0$ , an initial vertex,  $\mathbf{L}(0) = [R_0, Z_0, 0]$ , is picked up on the plane with  $\varphi = 0$  and it satisfies  $\psi(R_0, Z_0) = \psi_0$ . The vertices on the curve (Eq. 2) are placed at the sequence of the parameters defined as

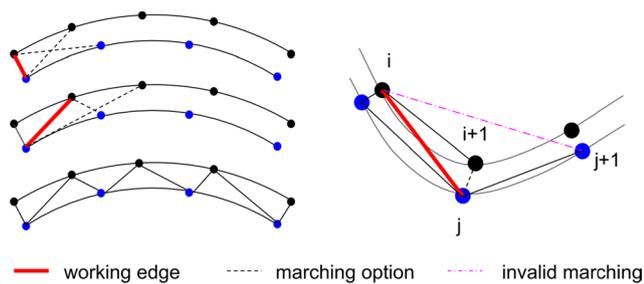
$$t_i = \frac{i}{2n\pi}, \quad i = 0, 1, \dots, N - 1. \tag{3}$$

The sequence is terminated and an approximately closed curve is formed based on the mesh vertex spacing requirement,  $d_i$ , or it reaches the wall boundary. Note that  $\varphi$  of each vertex is replaced by a constant value and the vertices are projected to the same cross section.

The parameter,  $d_i$ , is specified to set a tolerance of vertex spacing. The sequence of the vertices defined by Eq. 3 is refined by bisecting the interval of parameters if the distance between vertices is greater than  $d_i$ . For example, bisecting the interval  $[t_i, t_{i+1}]$  places an additional point  $\mathbf{L}\left(\frac{t_i+t_{i+1}}{2}\right)$  between points,  $\mathbf{L}(t_i)$  and  $\mathbf{L}(t_{i+1})$ . The sequence of the vertices is coarsened if the distance between vertices falls below  $d_i$ .

Given the placement of the curves and vertices, triangular elements on the geometric faces between adjacent curves are created by a *one-element-deep marching procedure*. The procedure starts with an edge designated as an initial working edge. The two marching options of creating new elements in the marching direction are evaluated based on the validity of the element and the shape indicator. The validity requires that the new edge must fall between the working edge and the edge on the surface in the marching direction. The marching option with a better element shape is chosen to form a new element. The edge created between the layers becomes the new working edge. The procedure continues until the last element is created.

Figure 8 illustrates the one-element-deep marching procedure. Assume the working edge is  $(i, j)$ . The new edge created by the marching procedure can be either  $(i, j + 1)$  or  $(j, i + 1)$  and the corresponding new element is  $(i, j, i + 1)$  or  $(j, i, j + 1)$ . According to the validity requirement, edges  $\{(i, j), (i, j + 1), (i, i + 1)\}$  or  $\{(j, i), (j, i + 1), (j, j + 1)\}$  must be placed clockwise or counter-clockwise. Therefore, element  $(i, j, i + 1)$  or  $(j, i, j + 1)$  must satisfy  $(\mathbf{v}_{i,j} \times \mathbf{v}_{i,j+1}) \cdot (\mathbf{v}_{i,j+1} \times \mathbf{v}_{i,i+1}) > 0$



**Fig. 8** One-element-deep marching procedure to generate *triangular* mesh faces between *curves*

or  $(\mathbf{v}_{j,i} \times \mathbf{v}_{j,i+1}) \cdot (\mathbf{v}_{j,i+1} \times \mathbf{v}_{j,j+1}) > 0$  to be valid, where  $\mathbf{v}_{j,i}$  defines the vector from point  $i$  to point  $j$ . Element  $(j, i, j + 1)$  is invalid although it gives a better shape indicator; thus, it will not be chosen as the new element in the marching procedure. Note that the current point placement controlled by  $d_i$  leads to the similar local mesh size on the two adjacent curves.

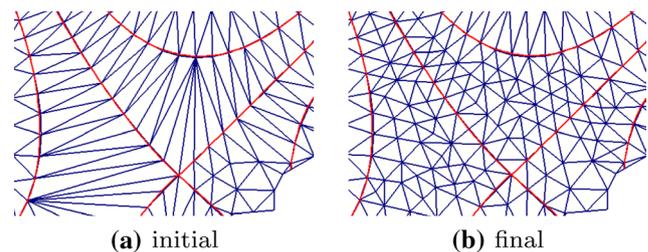
### 3.3 Mesh modification

General mesh modifications allow the users to adapt the unstructured mesh to match an anisotropic mesh size field defined over the initial mesh [27–29]. In order to evaluate the mesh quality, the mesh element is transformed by the metric tensor that defines the desired mesh size field [30] such that the mesh modification is controlled by the modification criterion and the desired mesh size field. The quality of the element size and shape is evaluated in the metric space. The type of the local mesh modification is chosen based on the evaluation of different operations [28].

In this paper, the mesh adaptation procedure is used to improve both the initial unstructured mesh and the layered mesh in the selected areas, and/or to adapt them to control mesh discretization errors. In case of XGC1, the layered mesh generation controlled by  $\delta\psi$  and  $d_i$  results in elements with poor shapes near the X-point. Therefore, mesh modification is used to improve the mesh quality and element shapes near the X-point. Figure 9 illustrates the layered mesh before and after mesh modification near the X-point. In M3D-C<sup>1</sup>, an initial mesh is obtained by the unstructured triangulation controlled by the mesh size parameters specified on the model entities and mesh adaptation is performed during the analysis. In this case, error indicators, or given functions, are used to define a new anisotropic mesh size field and a combined set of mesh modification operations are applied to convert the current mesh into one that satisfies the new mesh size field [31].

### 3.4 3D mesh construction

In parallel simulations, a mesh is split into multiple *parts* for the purpose of distribution to processes. Therefore, each



**Fig. 9** Improved mesh near the X-point

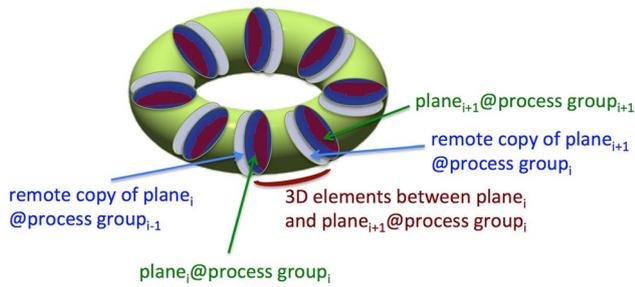
part consists of the set of mesh entities that is assigned to a process. For efficient manipulation, a part is uniquely identified within an entire mesh by a global part ID. Based on the adjacency relations, mesh entities on inter-part boundaries are duplicated to connect entities across parts such that they describe *part boundaries*. With the addition of part boundaries, each part is treated as a serial mesh. Each mesh entity duplicated on a inter-part boundary maintains a set of *remote copies* that is the memory location of mesh entity duplicated in the other part. The remote copy information is updated as the mesh partitioning changes dynamically, which is required by mesh modification or load balancing [32, 33].

In a 2D M3D-C<sup>1</sup> analysis with  $P$  processes, the mesh is distributed into  $P$  parts. In a 3D M3D-C<sup>1</sup> simulation,  $N$  copies of the same 2D mesh are loaded into  $N * P$  processes, where  $N * P$  processes are divided into  $N$  process groups such that each process group loads the 2D mesh onto a set of  $P$  processes. Within a process group, each process is assigned with a rank  $p$ ,  $0 \leq p < P$ , and a process group is uniquely identified as plane  $i$ ,  $0 \leq i < N$ . For each plane  $i$ , the backward plane is the plane  $i - 1$ , and the forward plane is the plane  $i + 1$ . For plane  $0$ , the backward plane is plane  $N - 1$ . For the plane  $N - 1$ , the forward plane is the plane  $0$ . In order to switch the mesh from 2D to 3D, on each plane  $i$ , a remote copy of the forward plane is created and then quadrilateral mesh faces and wedge elements are created using the entities on plane  $i$  and the remote copy of the forward plane. The total number of 3D elements created is the number of triangular faces in the 2D mesh times the number of planes. Figure 10 illustrates a 3D mesh constructed with eight planes.

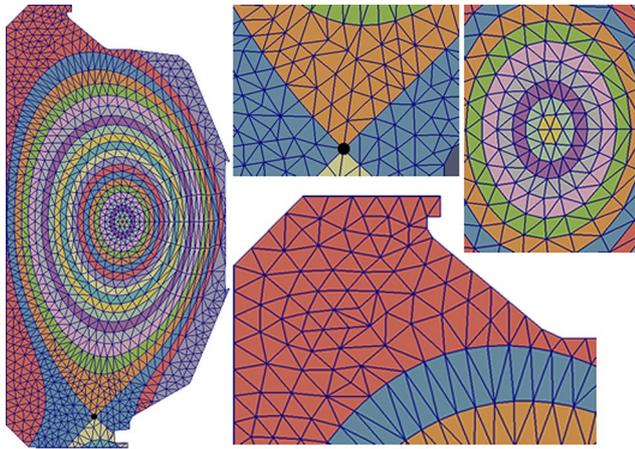
## 4 Examples

### 4.1 XGC1

Using the meshing control parameters discussed in Sect. 3, the steps for mesh generation are as follows: (i) the geometric model with the material wall boundary and magnetic flux surfaces interior to the domain is created, (ii)



**Fig. 10** 3D mesh constructed on eight process groups



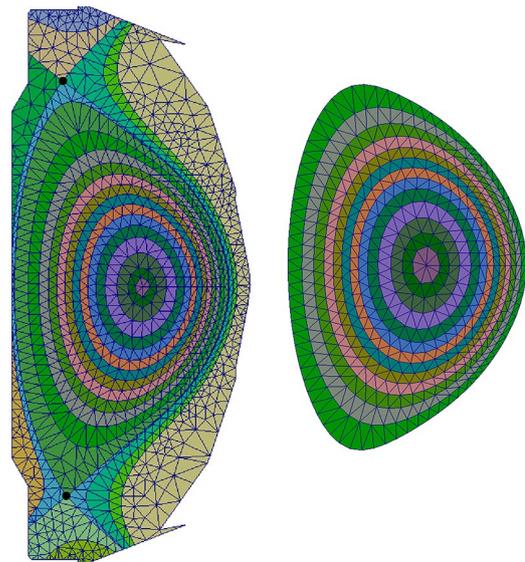
**Fig. 11** Example mesh with one X-point (3342 elements)

triangular elements are created on the model face with matched  $\psi$  curves that are oriented in the opposite directions, (iii) mesh modification is applied to improve the elements in poor shape near the X-point, and (iv) the rest of area is filled with unstructured elements.

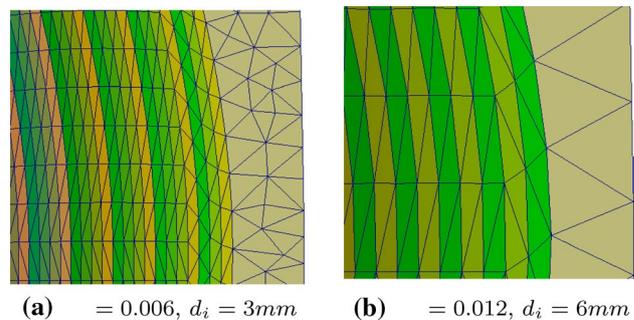
Figures 11 and 12 illustrate the example meshes with different numbers of X-point in the simulation domain. In both figures, the X-point is labeled with a solid circle. Figure 13 depicts how  $\delta\psi$  and  $d_i$  control the spacing of  $\psi$  curves and mesh vertices on each curve. Figure 14 illustrates how the targeted mesh size field affects mesh improvement in the X-point area.

### 4.2 M3D-C<sup>1</sup>

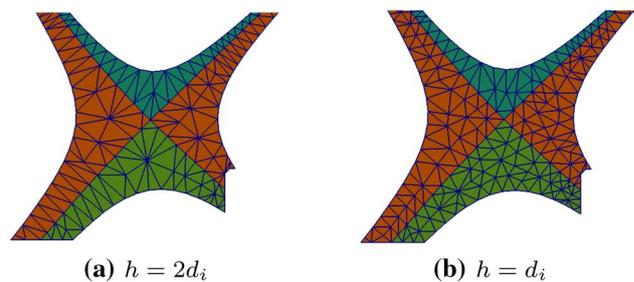
The simulation domain for M3D-C<sup>1</sup> consists of at most three areas that correspond to the plasma area, the finite-thickness material wall, and the vacuum vessel. In a 2D mesh, unstructured mesh elements are created and then adapted by an anisotropic size field. In a 3D mesh, the full torus is created by extrusion of the toroidal 2D meshes on toroidal cross sections.



**Fig. 12** Example mesh with two X-points (2878 elements) and no X-point (1030 elements)

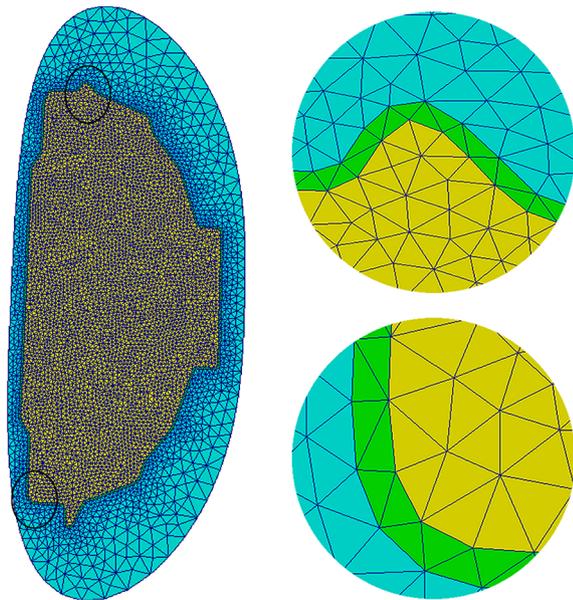


**Fig. 13** Two meshes with different field line placement ( $\delta\psi$ ) and vertex spacing ( $d_i$ ) parameters



**Fig. 14** Improved X-point area by different targeted mesh sizes ( $h$ )

Figure 15 depicts an example of the initial mesh on the NSTX model with a finite-thickness wall. The mesh size is specified at the model faces of the plasma, material wall, and the vacuum vessel to control the initial mesh. Figure 16



**Fig. 15** Initial mesh on the NSTX model with a finite-thickness wall

illustrates an example of parallel mesh adaptation in four processes with the boundary curve defined as

$$\begin{aligned} R(t) &= c_1 + c_2 \cos(t + c_3 \sin(t)) \\ Z(t) &= c_4 + c_5 \sin(t), \quad 0 \leq t \leq 2\pi \end{aligned} \tag{4}$$

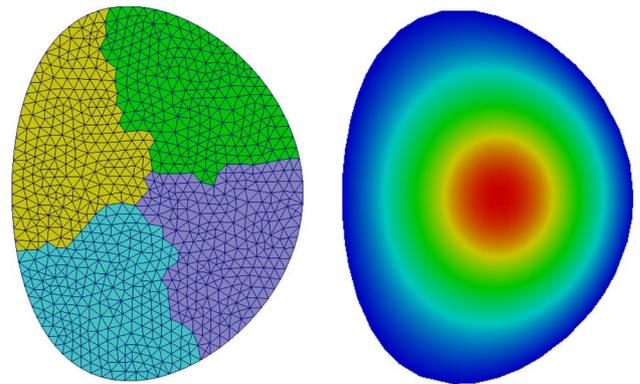
The mesh size field is calculated from the poloidal magnetic flux field  $\psi$ . A normalized field is defined as  $\tilde{\psi} = \frac{\psi - \psi_0}{\psi_l - \psi_0}$ , where  $\psi_l$  and  $\psi_0$  are the field values at the plasma boundary and the magnetic axis, respectively. The mesh size normal to the surface is  $h_1$  and the mesh size tangent to the surface is  $h_2$  that are defined as

$$h_i^{-1} = \tilde{h}_i^{-1} + \frac{1}{l_{ci} \left( 1 + \frac{\tilde{\psi} - \psi_c}{W_c} \right)^2}, \quad i = 1, 2 \tag{5}$$

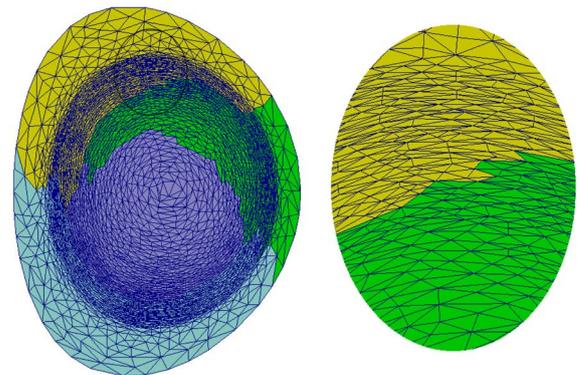
where  $l_{ci}$ ,  $\psi_c$ , and  $W_c$  are constants.  $\tilde{h}_i$  is defined as

$$\begin{aligned} \tilde{h}_i &= b_i \left[ 1 - e^{-|\frac{\tilde{\psi}}{a_i} - 1|^{a_2}} \right] + c_i, \quad \tilde{\psi} < a_1 \\ \tilde{h}_i &= d_i \left[ 1 - e^{-|\frac{\tilde{\psi}}{a_i} - 1|^{a_2}} \right] + c_i, \quad \tilde{\psi} > a_1 \end{aligned} \tag{6}$$

where  $b_i$ ,  $d_i$ ,  $a_i$  and  $c_i$  are constants. The constant parameters of the equations are determined such that the adapted mesh has finer size at the magnetic flux surface  $\tilde{\psi} = a_1$ . Since the solution to the physical equations varies more rapidly in the direction normal to the magnetic surfaces than within the surfaces, the directional mesh size fields are defined to represent this property. Figure 17 illustrates an example of a 3D mesh constructed with 64 planes. In Fig. 17a and c, different colors represent different process ranks.

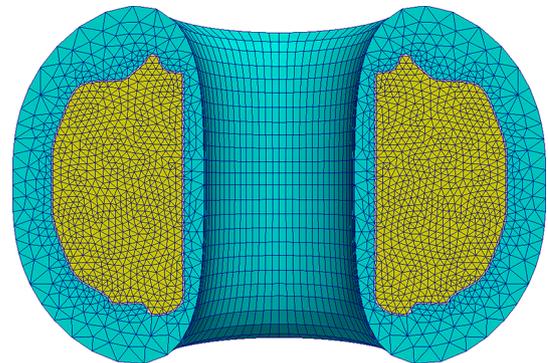


**(a)** initial mesh (1,678 elements) **(b)** solution field on the initial mesh



**(c)** adapted mesh and its close-up (5,746 elements)

**Fig. 16** Anisotropically adapted mesh in M3D-C<sup>1</sup>



**Fig. 17** Cross-cut view of a 3D mesh with 64 planes in M3D-C<sup>1</sup>

### 5 Closing remarks

This paper has presented a set of procedures for the automatic mesh generation with well-defined control parameters to satisfy the needs of two fusion plasma simulations codes, XGC1 and M3D-C<sup>1</sup>. Core capabilities include

- Employing a geometric model definition of the domain that represents physical and physics components that must be reflected in the resulting mesh.
- Straightforward specification of the needed mesh control information in terms of the geometric model.
- A component-based mesh generation procedure that satisfies the constraints of the simulation procedures while creating well-controlled graded meshes.

**Acknowledgments** This work was supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences and Office of Advanced Scientific Computing Research under Award Numbers DE-SC0008449 (Partnership for Edge Physics Simulations), DE-SC0006618 (Center for Extended Magnetohydrodynamic Modeling), and DE-SC0006617 (Frameworks, Algorithms, and Scalable Technologies for Mathematics).

## References

1. Chang C, Ku S, Diamond P, Lin Z, Parker S, Hahn T, Samatova N (2009) Compressed ion temperature gradient turbulence in diverted tokamak edge. *Phys Plasmas* 16(5):056108
2. Ku S, Chang C, Diamond P (2009) Full-f gyrokinetic particle simulation of centrally heated global itg turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry. *Nucl Fusion* 49:115021
3. Adams MF, Ku SH, Worley P, D’Azevedo E, Cummings JC, Chang C (2009) Scaling to 150k cores: Recent algorithm and performance engineering developments enabling xgc1 to run at scale. *J Phys* 180(1):012036
4. Jardin SC (2004) A triangular finite element with first-derivative continuity applied to fusion MHD applications. *J Comput Phys* 200(1):133–152
5. Jardin SC, Breslau J, Ferraro NM (2007) A high-order implicit finite element method for integrating the two-fluid magnetohydrodynamic equations in two dimensions. *J Comput Phys* 226(2):2146–2174
6. Ferraro NM, Jardin SC (2009) Calculations of two-fluid magnetohydrodynamic axisymmetric steady-states. *J Comput Phys* 228(20):7742–7770
7. Jardin SC (2012) Review of implicit methods for the magnetohydrodynamic description of magnetically confined plasmas. *J Comput Phys* 231(3):822–838
8. Jardin SC, Ferraro NM, Breslau J, Chen J (2012) Multiple time-scale calculations of sawteeth and other global macroscopic dynamics of tokamak plasmas. *Comput Sci Discov* 5(1):014002
9. Ferraro NM, Jardin SC, Shephard MS, Bauer A, Breslau JA, Chen J, Delalondre F, Luo X, Zhang F (2011) Fluid modeling of fusion plasmas with M3D-C<sup>1</sup>. In: *SciDAC Conference 2011*, Denver, CO
10. Ferraro NM, Lao LL, Turnbull AD, Jardin SC, Sabbagh SA, Shephard MS, Zhang F (2014) Resistive wall model in M3D-C<sup>1</sup>. In: *2014 International Sherwood Fusion Theory Conference*
11. Jardin SC (2010) *Computational methods in plasma physics*. CRC Press, Boca Raton
12. Park W, Belova E, Fu G, Tang X, Strauss H, Sugiyama L (1999) Plasma simulation studies using multilevel physics models. *Phys Plasmas* 6(5):1796–1803
13. Center for edge physics simulation web page. <http://epsi.pppl.gov/>. Accessed 14 April 2015
14. Wesson J (2011) *Tokamaks*. Oxford University Press, Oxford
15. Chen F, Chen F (1984) *Introduction to plasma physics and controlled fusion. Volume 1: plasma physics*. Plenum Press, New York
16. Beall MW, Walsh J, Shephard MS (2004) A comparison of techniques for geometry access related to mesh generation. *Eng Comput* 20(3):210–221
17. Hoffmann CM (1989) *Geometric and solid modeling*. Morgan Kaufmann, San Mateo
18. Shephard MS, Georges MK (1992) Reliability of automatic 3D mesh generation. *Comput Methods Appl Mech Eng* 101(1):443–462
19. Beall MW, Shephard MS (1997) A general topology-based mesh data structure. *Int J Numer Methods Eng* 40(9):1573–1596
20. Garimella R (2002) Mesh data structure selection for mesh generation and FEA applications. *Int J Numer Methods Eng* 55(4):451–478
21. Celes W, Paulino G, Espinha R (2005) A compact adjacency-based topological data structure for finite element mesh representation. *Int J Numer Methods Eng* 64(11):1529–1556
22. Prautzsch H, Boehm W, Paluszny M (2002) *Bézier and B-spline techniques*. Springer, New York
23. National spherical torus experiment (NSTX). <http://www.pppl.gov/nstx>. Accessed 14 Apr 2015
24. EFIT equilibrium and reconstruction fitting code.” <https://fusion.gat.com/theory/Efit>. Accessed 14 Apr 2015
25. PSPLINE help <http://w3.pppl.gov/~pshare/help/pspline.htm>. Accessed 14 Apr 2015
26. Simmetrix web page. <http://www.simmetrix.com/>. Accessed 14 Apr 2015
27. Li X, Shephard MS, Beall MW (2003) Accounting for curved domains in mesh adaptation. *Int J Numer Methods Eng* 58(2):247–276
28. Li X, Shephard MS, Beall MW (2005) 3D anisotropic mesh adaptation by mesh modification. *Comput Methods Appl Mech Eng* 194(48):4915–4950
29. Alauzet F, Li X, Seol ES, Shephard MS (2006) Parallel anisotropic 3D mesh adaptation by mesh modification. *Eng Comput* 21(3):247–258
30. Bottasso CL (2004) Anisotropic mesh adaption by metric-driven optimization. *Int J Numer Methods Eng* 60(3):597–639
31. Shephard MS, Seol ES, Smith CW, Mubarak M, Ovcharenko A, Sahni O (2013) Methods and tools for parallel anisotropic mesh adaptation and analysis. In: Moitinho de Almeida JP, Díez P, Tiago C, Parés N (eds) *VI International Conference on Adaptive Modeling and Simulation ADMOS 2013*
32. Seol ES, Shephard MS (2006) Efficient distributed mesh data structure for parallel automated adaptive analysis. *Eng Comput* 22(3–4):197–213
33. Ibanez DA, Seol ES, Smith CW, Shephard MS (2015) PUMI: parallel unstructured mesh infrastructure, *ACM Trans Math Softw* (To appear)