Tracking Adaptive Moving Mesh Refinements in 3D Curved Domains for Large-Scale Higher Order Finite Element Simulations

Xiaojuan Luo[1], Mark S. Shephard[1], Lie-Quan Lee[2], Cho Ng[2], and Lixin Ge[2]

[1] Rensselaer Polytechnic Institute, Troy, NY 12180
xluo@scorec.rpi.edu, shephard@scorec.rpi.edu
[2] Stanford linear Accelerator Center (SLAC), Menlo Park, CA 94025
liequan@slac.stanford.edu, cho@slac.stanford.edu, lge@slac.stanford.edu

Summary. When applying higher order finite elements to curved 3D domains in large-scale accelerator simulations, complexities that arise include needing valid curved finite elements and the capability to track the movement of mesh refinement in the critical domains. This paper presents a procedure which combines Bézier mesh curving and size driven mesh adaptation technologies to address those requirements. The intelligent selection of local mesh modifications to eliminate invalid curved elements and properly control the size distribution are the two key technical components. The procedure has been successfully applied by SLAC to generate 3D moving curved meshes in the large-scale electromagnetic modeling of next generation accelerator designs. The results demonstrated that valid curvilinear meshes not only make the time domain simulations more reliable but also improve the computational efficiency up to 30%.

1 Introduction

Higher order finite elements [1], which are well known for faster rates of convergence in terms of computational efficiency, can provide an effective approach for large-scale simulations. When applying higher order finite elements to three-dimensional curved domains, the elements must be properly curved to maintain the rate of convergence [2]. The common approach to the construction of such meshes is to apply a straight-sided mesh generation procedure [3, 4] and then curve the mesh edges and faces on the curved domain boundaries to the proper orders. This approach takes advantage of the conventional unstructured mesh generators to deal with the complexity of model geometry. However, the resulting meshes may become invalid because the curving of the mesh entities to model boundaries can lead to negative determinant of Jacobian in the closures of curved elements. Effective and efficient correction of those invalid elements is critical in curvilinear mesh construction and for its usage with higher order finite elements.

The electromagnetic simulation tools developed at Stanford Linear Accelerator Center (SLAC), supported by DOE SciDAC program [5, 6, 7] have

successfully taken advantage of higher order finite elements to perform analysis needed for the design of the next generation linear accelerators, for example, short-range wakefield calculations. In those simulations, a small region near the beam requires sufficient refinement to resolve high frequency while the rest of the domain can have a large mesh size. This refinement region must move with the particles beam through the curved domains in the time dependent simulations to achieve acceptable computational efficiency. Since the domains are curved and higher order finite elements are used, the refined meshes must also be curved to provide a sufficiently higher order geometric approximation to ensure that the solution can effectively achieve the desired level of accuracy. The uniform refinement using smaller mesh size throughout the entire domain can produce over-refined meshes outside of the critical particle beam domains while larger mesh size can generate too coarse meshes which often become invalid during the curving procedure. These lead either to infeasibly large problem size, inaccurate results, or possible failure of the simulations.

To enable the higher order finite elements in large-scale simulations, a moving mesh adaptation in curvilinear domains that includes the combination of a general mesh curving tool [8, 9] and size driven mesh adaptation [10, 11] is presented. A full set of curved mesh modification procedures play an important role in ensuring the resulting curved meshes are valid and with minimal number of elements for the desired accuracy.

The outline of this paper is as follows. Section 2 describes a general mesh curving procedure to construct valid curvilinear meshes for three-dimensional curved domains. The procedure employs Bézier polynomial to represent the higher order geometric shapes for curved mesh entities. The extension of the size driven mesh adaptation procedure in curved domains is discussed in Section 3. Analyses results applied by SLAC for linear accelerator design are shown in Section 4. Conclusions and future works are given Section 5.

2 Mesh Curving

A flexible distributed mesh data structure [12] is employed in this paper to support the mesh adaptation in curved domains. The mesh data structure applies a general topology and classification of the entities with respect to the geometric model entity that the mesh entity is on [13]. M_i^d and G_i^d are used to describe the mesh and model topological entity of dimension d, $d = 0, 1, 2, 3$ represent mesh and model vertex, edge, face, and region respectively.

The approximation of the mesh to curved geometric domains is maintained by assigning appropriate Bézier higher order geometric shapes to mesh edges and faces on curved domain boundaries. The Bézier topology-based mesh geometry shape is constructed using Bernstein polynomials which possess a number of advantageous properties including [14]:

- The Convex Hull Property - A Bézier curve, surface, or volume is contained in the convex hull formed by its control points.

- Computationally efficient algorithms for degree elevation and subdivision are available which can be used to refine the shape's convex hull as well as adaptively refine the mesh's shape.

Those properties are useful to form the validity check algorithm for Bézier higher order curved elements and to determine local mesh modifications to correct invalid elements which have negative determinant Jacobian in the element closures due to the curving of mesh entities. The resulting curved meshes can always guarantee that each element has positive determinant of Jacobian in its closures.

2.1 Bézier Topology-Based Higher Order Shape Representation

Bernstein polynomials provide an effective means to define Bézier hierarchic higher order shapes for topological mesh entities in their parametric coordinates. A *qth* order Bézier mesh entity can be represented as [9],

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{|\boldsymbol{i}|=q} B_{|\boldsymbol{i}|}(\boldsymbol{\xi}) b_{|\boldsymbol{i}|} \boldsymbol{\xi}^{|\boldsymbol{i}|} \tag{1}$$

where $B_{|\boldsymbol{i}|}\boldsymbol{\xi}^{|\boldsymbol{i}|}$ are the Bernstein polynomials defined in the mesh entity parametric coordinate system as shown in Table 1. Note that the independent parametric coordinates for a topological mesh entity should be ξ_1, (ξ_1, ξ_2) and (ξ_1, ξ_2, ξ_3) respectively with the constraints that $\xi_2 = 1 - \xi_1$, $\xi_3 = 1 - \xi_1 - \xi_2$ and $\xi_4 = 1 - \xi_1 - \xi_2 - \xi_3$. $b_{|\boldsymbol{i}|}$ are the control points used to define the shapes of the Bèzier mesh edges, faces and regions. Figure 1 shows the control points for a quadratic curved mesh edge, triangle face and tetrahedral region.

Table 1. $\boldsymbol{\xi}$, $|\boldsymbol{i}|$, $B_{|\boldsymbol{i}|}$ and $\boldsymbol{\xi}^{|\boldsymbol{i}|}$for topology mesh entity

	$\boldsymbol{\xi}$	$\lvert\boldsymbol{i}\rvert$	$B_{\lvert\boldsymbol{i}\rvert}$	$\boldsymbol{\xi}^{\lvert\boldsymbol{i}\rvert}$
Edge	(ξ_1, ξ_2), $\xi_1 + \xi_2 = 1$	$\lvert\boldsymbol{i}\rvert = i + j$	$\frac{q!}{i!j!}$	$\xi_1^i \xi_2^j$
Triangle	(ξ_1, ξ_2, ξ_3), $\xi_1 + \xi_2 + \xi_3 = 1$	$\lvert\boldsymbol{i}\rvert = i + j + k$	$\frac{q!}{i!j!k!}$	$\xi_1^i \xi_2^j \xi_3^k$
Tetrahedron	$(\xi_1, \xi_2, \xi_3, \xi_4)$, $\xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$	$\lvert\boldsymbol{i}\rvert = i + j + k + l$	$\frac{q!}{i!j!k!l!}$	$\xi_1^i \xi_2^j \xi_3^k \xi_4^l$

In the case that a straight-sided mesh and its associated geometry CAD model are given, the control points for those mesh entities on the curved model edges/faces are determined based on Bézier curve and surface interpolation method to evaluate the model geometry at a set of discrete parametric locations. Common approaches often use the uniformly distributed parametric space points. However, alternative methods, such as chord length method or curvature-based procedure, will be used to improve the geometric approximation [14]. In the case that a curved mesh with different shape representation method is given, the control points are computed by converting the given shapes to Bézeir shapes. For example, a quadratic mesh edge using Lagrange interpolating method to represent its curved shape has three control points $\mathbf{l_1}$, $\mathbf{l_2}$ and $\mathbf{l_3}$, the control points

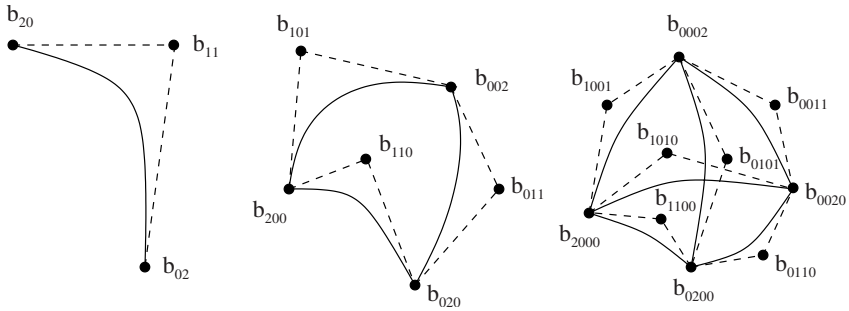


Fig. 1. Bézier control points for a curved mesh edge, face and region

for the Bézier shape of the edge, $\boldsymbol{x} = \mathbf{b}_{20}\xi_1^2 + 2\mathbf{b}_{11}\xi_1(1-\xi_1) + \mathbf{b}_{02}(1-\xi_1)^2$, can be computed by solving the following equations,

$$\begin{array}{lll} \boldsymbol{x}(\xi_1 = 0) & = \mathbf{b}_{02} & = \mathbf{l}_1 \\ \boldsymbol{x}(\xi_1 = 1/2) & = \frac{\mathbf{b}_{20}}{4} + \frac{\mathbf{b}_{11}}{2} + \frac{\mathbf{b}_{02}}{4} & = \mathbf{l}_2 \\ \boldsymbol{x}(\xi_1 = 1) & = \mathbf{b}_{20} & = \mathbf{l}_3 \end{array} \tag{2}$$

Therefore, $\mathbf{b}_{02} = \mathbf{l}_1$, $\mathbf{b}_{20} = \mathbf{l}_3$ and $\mathbf{b}_{11} = (4\mathbf{l}_2 - (\mathbf{l}_1 + \mathbf{l}_3))/2$.

2.2 Validity Check of Bézier Higher Order Curved Elements

When the finite element basis is increased in applying the higher order finite elements, the integration rules of the finite element approximation must be properly improved to ensure that the numerical integration error does not become the dominant error. The improvement of the integration rules requires evaluation of the geometric mapping of the curved element at new integration locations or knowledge that the Jacobian is positive throughout the element. To avoid the need to constantly recheck the integration points, a general element validity check is performed. This validity check for Bézier higher order curved elements takes advantage of its convex hull property to ensure that a valid curved element always has positive determinant of Jacobian in its closures, which is independent of the basis functions, the polynomial orders, or the applied integration rules [8].

Given a *qth* order Bézier tetrahedron mesh region described in Eq.1, the Jacobian matrix of the geometric mapping with respect to the independent parametric coordinates (ξ_1, ξ_2, ξ_3) is,

$$J = \left[\frac{\partial \mathbf{x}}{\partial \xi}\right] = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_1}{\partial \xi_3} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \\ \frac{\partial x_3}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_3} \end{bmatrix} \tag{3}$$

where $\mathbf{x} = (x_1, x_2, x_3)$ Therefore, the determinant of the Jacobian J is,

$$det(J) = (\frac{\partial \boldsymbol{x}}{\partial \xi_1} \times \frac{\partial \boldsymbol{x}}{\partial \xi_2}) \cdot (\frac{\partial \boldsymbol{x}}{\partial \xi_3}) \tag{4}$$

where $\frac{\partial \boldsymbol{x}}{\partial \xi_i}$ are the three partial derivatives of $\boldsymbol{x}^q$ which are $(q-1)th$ order Bézier functions. Therefore, the resulting determinant of Jacobian is a Bézier polynomial function with order $3(q-1)$,

$$det(J) = \sum_{|\boldsymbol{i}|=r} C_{|\boldsymbol{i}|} c_{|\boldsymbol{i}|} \boldsymbol{\xi}^{|\boldsymbol{i}|} \tag{5}$$

where $r = 3(q-1)$. $C_{|\boldsymbol{i}|}$ and $c_{|\boldsymbol{i}|}$ can be expressed using the coefficients $B_{|\boldsymbol{i}|}$ and $b_{|\boldsymbol{i}|}$ in Eq. 1. As an example, the quadratic tetrahedral region shown in Figure 1 can be expressed as,

$$\begin{aligned} \boldsymbol{x} = & B_{2000} b_{2000} \xi_1^2 + B_{0200} b_{0200} \xi_2^2 + B_{0020} b_{0020} \xi_3^2 + B_{0002} b_{0002} \xi_4^2 \\ & + B_{1100} b_{1100} \xi_1 \xi_2 + B_{1010} b_{1010} \xi_1 \xi_3 + B_{1001} b_{1001} \xi_1 \xi_4 \\ & + B_{0110} b_{0110} \xi_2 \xi_3 + B_{0101} b_{0101} \xi_2 \xi_4 + B_{0011} b_{0011} \xi_3 \xi_4 \end{aligned} \tag{6}$$

Considering that $\xi_4 = 1 - \xi_1 - \xi_2 - \xi_3$, $B_{2000} = B_{0200} = B_{0020} = B_{0002} = 1$ and the rest coefficients $B's$ equal to 2, therefore,

$$\begin{aligned} \frac{\partial \boldsymbol{x}}{\partial \xi_1} = & 2\{\underbrace{(b_{2000} - b_{1001})}_{a_1} \xi_1 + \underbrace{(b_{1100} - b_{0101})}_{b_1} \xi_2 \\ & + \underbrace{(b_{1010} - b_{0111})}_{c_1} \xi_3 + \underbrace{(b_{1001} - b_{0002})}_{d_1} \xi_4\} \\ \frac{\partial \boldsymbol{x}}{\partial \xi_2} = & 2\{\underbrace{(b_{1100} - b_{1001})}_{a_2} \xi_1 + \underbrace{(b_{0200} - b_{0101})}_{b_2} \xi_2 \\ & + \underbrace{(b_{0110} - b_{0011})}_{c_2} \xi_3 + \underbrace{(b_{0101} - b_{0002})}_{d_2} \xi_4\} \\ \frac{\partial \boldsymbol{x}}{\partial \xi_3} = & 2\{\underbrace{(b_{1000} - b_{1001})}_{a_3} \xi_1 + \underbrace{(b_{0110} - b_{0101})}_{b_3} \xi_2 \\ & + \underbrace{(b_{0110} - b_{0020})}_{c_3} \xi_3 + \underbrace{(b_{0011} - b_{0002})}_{d_3} \xi_4\} \end{aligned} \tag{7}$$

The determinant of Jacobian is a cubic Bernstein polynomial and the coefficients $C_{|\boldsymbol{i}|}$ and $c_{|\boldsymbol{i}|}$ are listed in Table 2. a_i, b_i, c_i and d_i are the vectors defined by the corresponding control points shown in Eq. 6.

The convex hull property of Bézier polynomial indicated that the polynomial is bounded by its minimal and maximal control points [14]. So,

$$min(c_{|\boldsymbol{i}|}) \leq det(J) \leq max(c_{|\boldsymbol{i}|}) \tag{8}$$

Therefore, a curved tetrahedral region is valid in its closure as long as its $min(c_{|\boldsymbol{i}|}) > 0$.

Table 2. $C_{|\boldsymbol{i}|}$ and $c_{|\boldsymbol{i}|}$ for $det(J)$ of a quadratic tetrahedral region

$\|\boldsymbol{i}\|$	$C_{\|\boldsymbol{i}\|}$	$c_{\|\boldsymbol{i}\|}$	$\|\boldsymbol{i}\|$	$C_{\|\boldsymbol{i}\|}$	$c_{\|\boldsymbol{i}\|}$
3000	8	$(a_1 \times a_2) \cdot a_3$	0300	8	$(b_1 \times b_2) \cdot b_3$
0030	8	$(c_1 \times c_2) \cdot c_3$	0003	8	$(d_1 \times d_2) \cdot d_3$
2100	8	$(a_1 \times b_2 + a_2 \times b_1) \cdot a_3$	1200	8	$(a_1 \times b_2 + a_2 \times b_1) \cdot b_3$
...	...	...	...	...	...

2.3 Effective Procedure to Generate Valid Curvilinear Meshes

The mesh curving procedure can deal with a starting straight-sided mesh and an existing curved mesh with invalid elements. In the case of a straight-sided mesh, the procedure computes the Bézier control points for the mesh edges/faces on curved domain boundaries and curves them incrementally. In the case of a given curved mesh with different initial higher order shapes representation, for example, Lagrange interpolation, the shapes are converted to Bézier form and the invalid elements are detected and corrected incrementally. Central to both of the approaches is the selection of effective local mesh modification operations to eliminate the invalid elements till the resulting curvilinear meshes are valid.

The computation of the determinant of Jacobian to detect invalid elements can provide useful information to determine key mesh entities and appropriate operations to correct the invalidity. The invalid elements are defined as those curved elements having at least one negative coefficients, $c_{|\boldsymbol{i}|} \leq 0$ as shown in Eq. 5. The key mesh entities are defined as those whose control points appear in the computation of the negative coefficients $c_{|\boldsymbol{i}|}$. As an example, Figure 2 shows an invalid quadratic tetrahedral region and the computation of the determinant Jacobian shows that coefficient $c_{3000} < 0$. Based on Eq.8 and Table 2, the control points $b_{2000}, b_{1100}, b_{1010}$ and b_{1001} have been used to compute the c_{3000} which indicates that M_0^0, M_0^1, M_1^1 and M_2^1 are key mesh entities and applying local

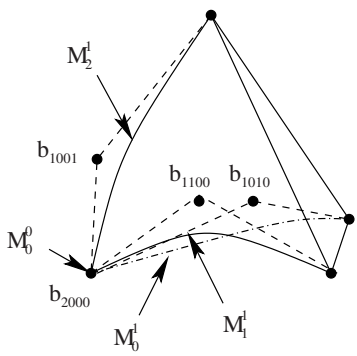


Fig. 2. The computation of det(J) indicates that the mesh entities M_0^0, M_0^1, M_1^1 and M_2^1 are key mesh entities

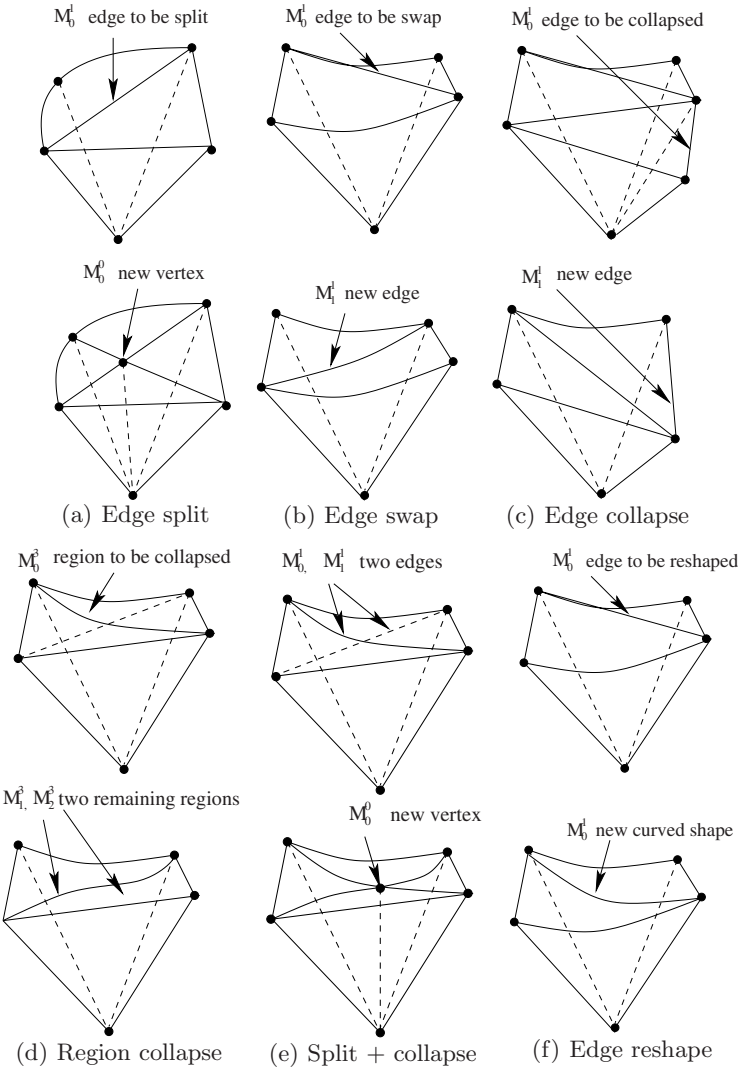


Fig. 3. 3D curved local mesh modification operations

mesh modifications on any of them can effectively make c_{3000} positive and the curved element valid.

The set of curved local mesh modifications applied to create valid curvilinear meshes include edge split, edge swap, edge collapse, region collapse, double split+collapse, and edge re-shape [9] as shown in Figure 3. Comparing to the straight-sided mesh, the validity check algorithm discussed in Section 2.2 is used to determine whether a curved local mesh modification operation can be applied.

Those operations are essential to ensure the reliability of the mesh curving procedure to create valid curved elements.

The procedure processes one curved mesh entity at a time as follows [8]:

- Determine the key mesh entities to apply local mesh operations based on the negative coefficients, $c_{|\boldsymbol{i}|} \leq 0$, in computing the determinant of Jacobian.
- Determine if the invalidity is caused by pairs of neighboring mesh faces or edges classified on the boundary such that angles of 180^0 are created. In those cases, $c_{|\boldsymbol{i}|} \leq 0$ only happens at $i, j, k, l = r$. Apply either split (see Figure 3(a)) or swap operations (see Figure 3(b)) to introduce additional entities to subdivide those larger angles to correct the invalid curved element.
- Determine if the invalidity is caused by pairs of opposite mesh edges coming too close to each other in one curved region, where $c_{|\boldsymbol{i}|} \leq 0$ happens at $i, j, k, l \neq r$. Apply either region split (see Figure 3(d)) or split+collapse (see Figure 3(e)) to remove the invalid curved element.
- If neither of above two steps is successful, examine the applications of the remaining operations (see Figure 3(c), 3(f)) to correct the invalid curved elements.
- If the invalid curved element can not be corrected using those local mesh operations, local refinement is applied and all new mesh entities will be added to the list to be processed. Subdivision creates more options for applying operations later.

Figure 4 shows the straight-sided and curved mesh for a 3D curved model to demonstrate the effectiveness of the developed procedure. The mesh has 139 regions and 31 curved regions are invalid. 20 local mesh modifications are applied to correct those invalid elements. Curved meshes for more complex domains used by SLAC for electromagnetic linear accelerator analysis are shown in section 4.

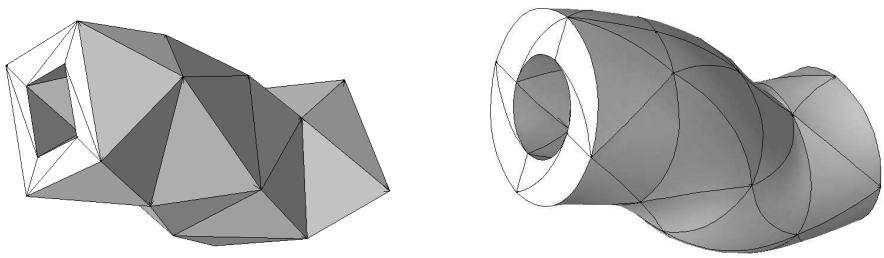


Fig. 4. Straight-sided mesh (left) and curved mesh (right) for a 3D curved domain

3 Moving Adaptive Mesh Refinements in 3D Curved Domains

The developed size driven mesh adaptation procedure [10] has been successfully applied in cardiovascular blood flow simulations [15], metal forming process [16],

wave propagation simulations [17], etc. where the results demonstrated substantial computational efficiency can be improved using the isotropic or anisotropic adapted meshes to effectively resolve solution fields. The procedure has been extended to deal with curved meshes for higher order finite elements in large-scale simulations. The extended procedure maintains the existing functionalities developed for straight-sided meshes such as vertex-based size field specifications and selective local mesh modification applications [10]. In addition, the following two steps have been added in the case where the mesh is curved.

- The validity check algorithm described in Section 2.2 must be applied when the affecting cavities for a local mesh modification operation have curved mesh entities. This step ensures that resulting curved meshes are valid after applying the selected local mesh operation.
- Any newly created mesh entities on the curved domain boundaries must be properly curved to the model boundaries which ensures that the geometric approximation of the resulting adapted meshes is maintained. As an example, Figure 5 shows how the procedure to split a quadratic curved mesh edge M_0^1 which is classified on the curved model edge G_0^1. The two newly created mesh edges M_1^1 and M_2^1 are also curved to the model edge G_0^1.

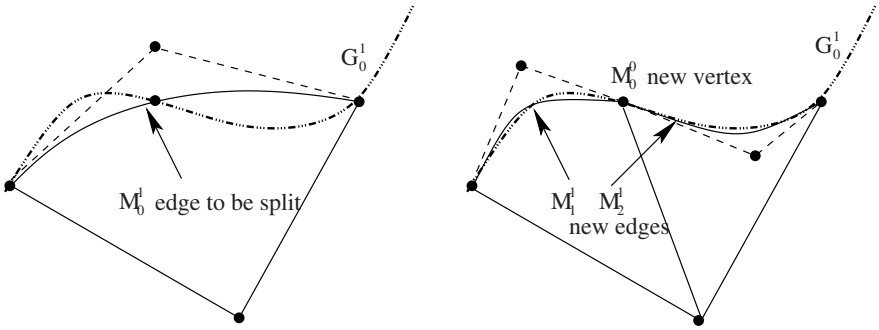


Fig. 5. Before (left) and after (right) refine a quadratic curved mesh edge M_0^1 on model edge G_0^1. New mesh edges M_1^1, M_2^1 have been appropriately curved to the model boundaries

In size driven mesh adaptation procedure, a mesh metric field, which can be either isotropic or anisotropic, is defined to specify the desired size of elements. The metric field is used to compute the edge length and directions of the current mesh with respect to this metric. A series of controlled mesh modification steps are applied to obtain a new mesh that satisfies the specified mesh metric field which consist of the following three steps [11]:

- Coarsening stage to eliminate the mesh edges that are shorter than the desired edge length in the metric field. This stage is accomplished by applying collapse operation on the identified shorter edges one at a time.

- Refinement stage to reduce the maximal mesh edge length to reach the desired edge length in the metric field. Edge-based refinement templates and application of local mesh modification to project the new created mesh vertices to the curved boundaries are iteratively applied until the adapted mesh satisfies the mesh size metric field requirements [10].
- Shape improvement stage to improve the quality of the resulting mesh using swap and/or vertex reposition operations.

For large-scale adaptive simulations, discretization error estimation is applied to construct size fields to control the mesh adaptation [15, 16, 17] in which the adapted meshes can conform to the size requirements. However, there are certain situations where other factors may also be applied to set the size field. For example, the size information being given for the short-range wakefield simulations performed by SLAC is supplemented to have a refined mesh in areas where particles currently reside. The specification of this refinement information is dictated by the initial locations of the particles beam and the desired mesh size around the beam which is often at least one order of magnitude smaller than the rest of the domains. The larger size difference between the finer particle domain and the coarse domain can lead to bad quality resulting meshes. The left mesh in Figure 6 shows an adapted curved mesh which uses size 1 and 10 to control the fine and coarse mesh in the model. The abrupt size field change causes meshes at the fine and coarse mesh interface not acceptable which clearly demonstrated that the control of the mesh gradation is needed.

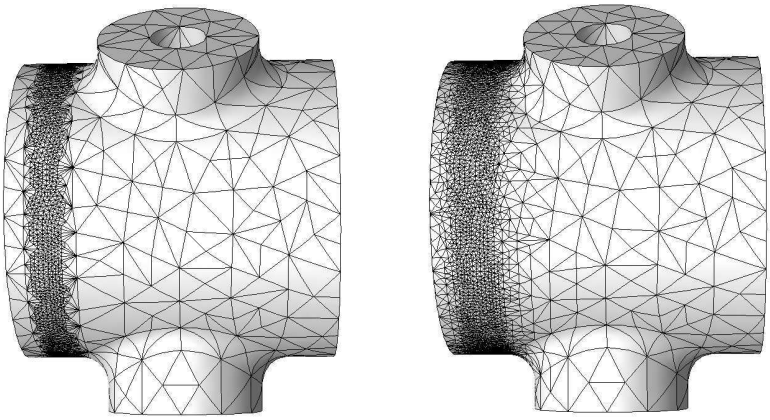


Fig. 6. Curved meshes without (left) and with (right) mesh size gradation control

The procedure described in [18] is adopted to control a smooth mesh size transition over the mesh. Central to the algorithm is that the ratio between the larger mesh size to the smaller mesh size at the two bounding mesh vertices of any mesh edge is under a prescribed factor β, where $\beta > 1$.

Let M_i^1 be a mesh edge, $M_{j_1}^0$ and $M_{j_2}^0$ are its two bounding mesh vertices, the given mesh sizes at each vertex are h_{j_1} and h_{j_2} which represent the desired edge lengths at those two vertices. We require that,

$$max\{\frac{h_{j_1}}{h_{j_2}}, \frac{h_{j_2}}{h_{j_1}}\}^{\frac{1}{L(M_1^1)}} \leq \beta \tag{9}$$

where $L(M_1^1)$ represents the length of the mesh edge M_i^1 with respect to the mesh size field as defined as,

$$L(M_1^1) = ||M_i^1|| \int_0^1 \frac{1}{H(t)} dt \tag{10}$$

where $||M_i^1||$ denotes the length of the mesh edge and $H(t)$ is a monotonic interpolation function along the mesh edge such that $H(0) = h_{j_1}$ and $H(1) = h_{j_2}$. For the piecewise linear mesh size field used in this paper to track the moving mesh adaptation in curved domains, the function $H(t)$ is,

$$H(t) = h_{j_1} + (h_{j_2} - h_{j_1})t \tag{11}$$

Therefore, Eq.10 gives,

$$L(M_1^1) = ||M_i^1|| \frac{log(h_{j_1}/h_{j_2})}{h_{j_1} - h_{j_2}}, h_{j_1} \neq h_{j_2} \tag{12}$$

Therefore, for any mesh edge which is not satisfied in Eq.9, the larger mesh size of its bounding mesh vertices is decreased to $min(h_{j_1}, h_{j_2})\beta^{L(M_i^1)}$ to meet Eq.9. The process is iteratively performed over the mesh when all of the mesh edges satisfy Eq.9. For the mesh shown in Figure 6(b), β is adopted as 2.0. More moving adaptive meshes are shown in section 4.

4 Analysis Results

4.1 Curvilinear Meshes for FETD Electromagnetic Simulations

SLAC performs simulations for the wakefield effects of an 8-cavity cryomodule for the proposed International Linear Collider (ILC) using the FETD method, which applies a set of higher order hierarchical Nedelec basis functions [19] for the finite element spacial discretization that requires the meshes to be curved. A curvilinear mesh with 2.974 million quadratic isoparametric tetrahedral elements is used in this FETD simulation. The initial curvilinear mesh uses Lagrange interpolation to represent the higher order shapes for those curved mesh edges which have been converted to Bézeir representations using Eq.2. 515 invalid curved elements were detected and have been corrected using the procedure discussed in Section 2. The valid curved mesh was exported by converting the Bézier shapes back to Lagrange shapes to be suitable for the analysis simulation system. Figure 7 shows the curved mesh for one cavity of the model and the close-up mesh before and after curving.

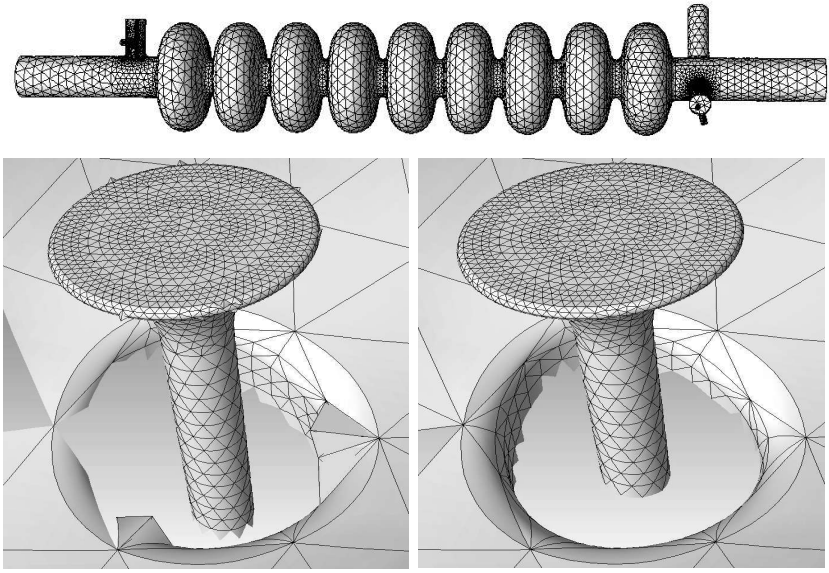


Fig. 7. The mesh for one cavity (top), close-up mesh before (bottom left) and after (bottom right) correcting the invalid curved elements marked as yellow

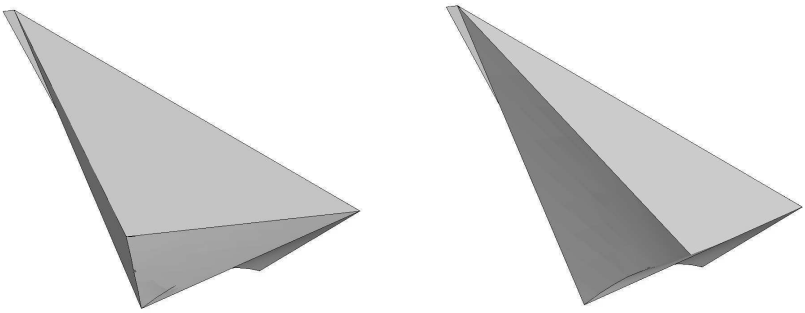


Fig. 8. Local mesh cavity before (left) and after (right) applying edge swap to correct the invalid element

Figure 8 shows how an edge collapse operation is applied to correct the invalid curved elements during curving process.

The mesh produced about 20 million degrees of freedom. The simulation used 256 Multi-stream processors on the Cray-X1E at Oak Ridge National Laboratory. It took a total runtime of 300 wall-hours through multiple jobs with checkpointing for a complete run and half terabtye of data was generated. Figure 9 shows a snapshot of the electric field distribution excited by a beam in the ILC cryomodule.

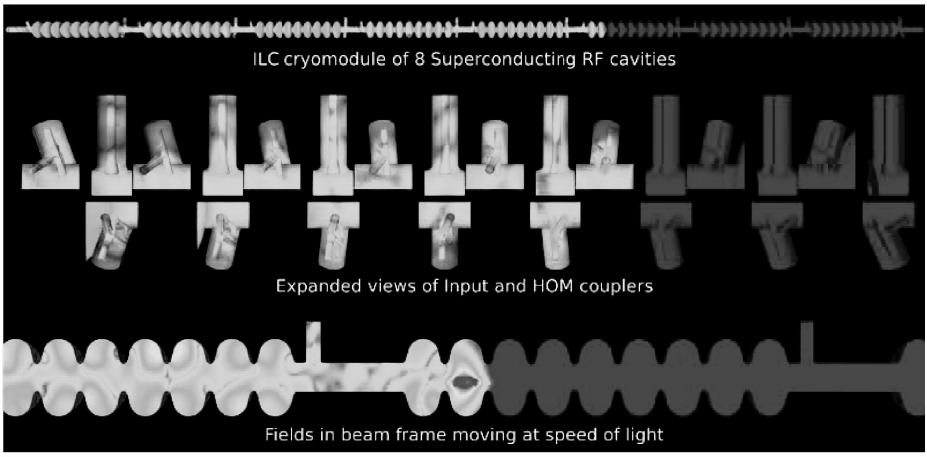


Fig. 9. A snapshot of the electric field distribution excited by a beam in an 8-cavity cryomodule for the proposed International Linear Collider

Table 3. Statistics for correcting the 2.97M mesh with 515 invalid curved regions

Time usage (sec)		Local mesh operations	
Import the mesh	381.162	Edge collapse	253
Create invalid region list	45.106	Region collapse	17
Correcting invalid regions	256.182	Edge swap	76
Export the mesh	64.911	Double edge split+collapse	13
		Recurving	32

The statistics for correcting the invalid curved regions is presented in Table 3. The data shows that the procedure used about 10 minutes to correct the invalid regions on a single processor linux workstation. The corrected curvilinear mesh not only leads to a stable time-domain simulation but also reduces the execution time per time-step by up to 30% due to better conditioned matrices, which is 90 wall-hours runtime efficiency improvement on the parallel computers.

4.2 Moving Adaptively Refined Meshes for Short-Range Wakefield Calculations

A series of moving adapted meshes in a curved domain were generated using the procedure described in Section 3 for short-range wakefield calculations by SLAC. Figure 10 shows the geometric model which has some complex components in the middle of domain. The initial location of the beam is at the left end of the domain, the desired mesh size inside the particle dense mesh is 1 and the size for the rest of the domains is 10. Figure 11 shows the moving adapted meshes up to step 5. Figure 12 shows the interior adapted mesh at step 5. Mesh size gradation control discussed in Section 3 is applied with $\beta = 2.0$. Figure 13 shows

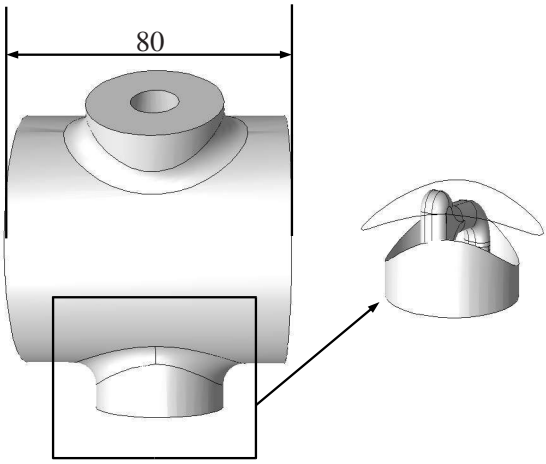
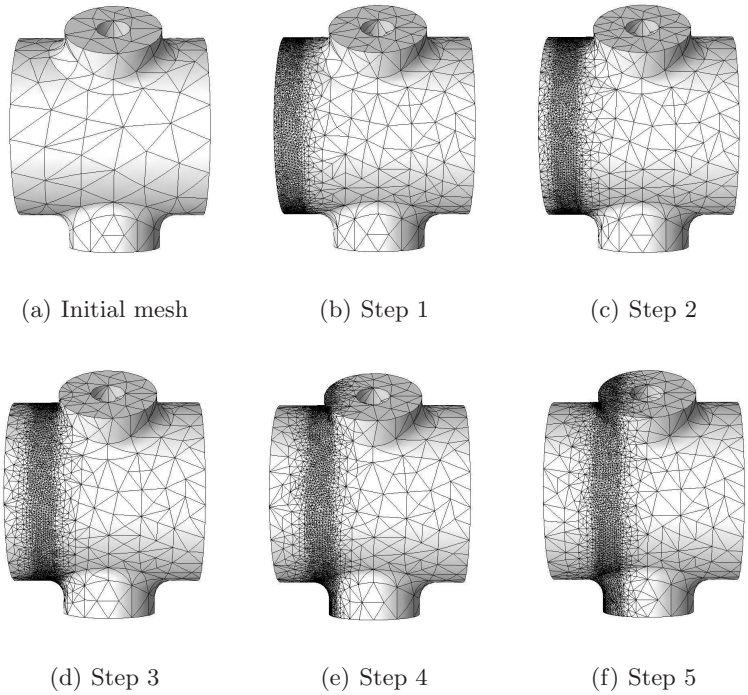


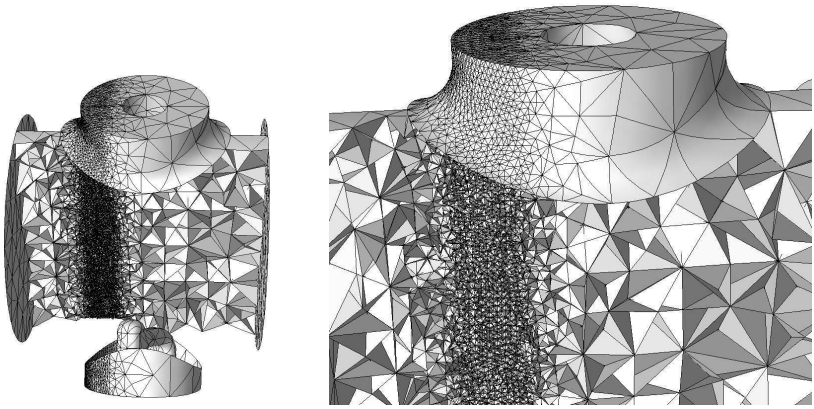
Fig. 10. Geometric model for the short-range wakefield simulation



(a) Initial mesh (b) Step 1 (c) Step 2

(d) Step 3 (e) Step 4 (f) Step 5

Fig. 11. Moving adapted meshes in curved domain for short-range wakefield simulation



(a) Interior (b) Close-up

Fig. 12. Interior mesh for the adapted mesh at step 5

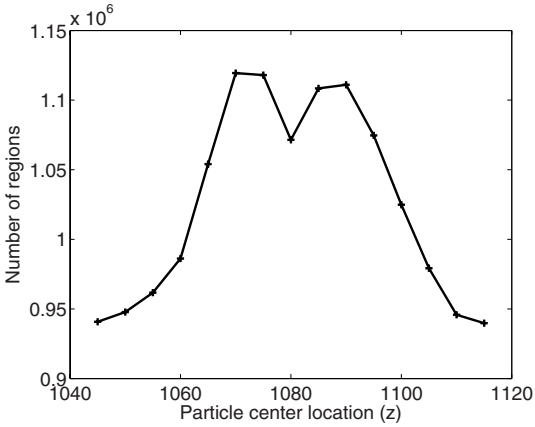


Fig. 13. Number of mesh regions at each step

the number of elements at each step which indicates that these adaptively refined meshes have around $1 \sim 1.15$ million elements compared to the uniform refined mesh with 6.5 million elements if the mesh size inside the particle domains is applied in the entire domain. The increase of the number of elements in the middle of the domain is due to the complex geometries as shown in Figure 10. The computation effort of short-range wakefield calculations using the moving adaptively refined meshes can be reduced by one order of magnitude compared to the uniformly-refined mesh.

5 Conclusion

This paper has presented a procedure to track moving adaptive mesh refinement in curved domains which is capable of generating suitable curvilinear meshes to enable large-scale accelerator simulations. The procedure combined a general mesh curving tool and size-driven mesh adaptation to produce valid curved meshes with substantially fewer elements and the analysis results demonstrated such meshes improved the computational efficiency and reliability. Future work will focus on the scalable parallelization of all steps for petascale simulations.

Acknowledgments

This work is supported by U.S. Department of Energy under DOE grant number DE-FC02-06ER25769 and DE-AC02-76SF00515.

References

1. Babuska, I., Szabo, B.A., Katz, I.N.: The p-version of the finite element method. Int. J. Numer. Meth. Engng. 18(3), 515–545 (1981)
2. Luo, X.J., Shephard, M.S., Remacle, J.F., O'Bara, R.M., Beall, M.W., Szabo, B.A., Actis, R.: p-version mesh generation issues. In: Proc.of 11th Meshing Roundtable, pp. 343–354 (2002)
3. CUBIT geometry and mesh generation toolkit, `http://cubit.sandia.gov`
4. Simmetrix Inc. Enabling simulation-based design, `http://www.simmetrix.com`
5. Akcelik, V., Ko, K., Lee, L.Q., Li, Z.H., Ng, C.K., Xiao, L.L.: Shape determination for deformed electromagnetic cavities. J. Comput. Physics 227(3), 1722–1738 (2008)
6. Xiao, L., Adolphsen, C., Akcelik, V., Kabel, A., Ko, K., Lee, L.Q., Li, Z., Ng, C.K.: Modeling imperfection effects on dipole modes in TESLA cavity. In: Proc. of 2007 Particle Accelerator Conference (2007)
7. Lee, L.Q., Akcelik, V., Chen, S., Gl, X., Prudencio, E., Schussman, G., Uplenchwar, R., Ng, C., Ko, K., Luo, X.J., Shephard, M.S.: Enabling technologies for petascale electromagnetic accelerator simulation. J. of Physics 78(Conference Series), 012040 (2007)
8. Luo, X.J., Shephard, M.S., Obara, R.M., Nastasia, R., Beall, M.W.: Automatic p-version mesh generation for curved domains. Engineering with Computers 20, 265–285 (2004)
9. Luo, X.J.: An Automatic Adaptive Directional Variable p-Version Method in 3D Curved Domains. PhD Thesis, Rensselaer Polytechnic Institute, New York (2005)
10. Li, X.R., Shephard, M.S., Beall, M.W.: Accounting for curved domains in mesh adaptation. Int. J. Numer. Meth. Engng. 150, 247–276 (2003)
11. Li, X.R., Shephard, M.S., Beall, M.W.: 3D anisotropic mesh adaptation by mesh modification. Comp. Meth. Appl. Mech. Engng. 194, 4915–4950 (2005)
12. Seegyoung, E., Shephard, M.S.: Efficient distributed mesh data structure for parallel automated adaptive analysis. Engineering with Computers 22, 197–213 (2006)
13. Beall, M.W., Shephard, M.S.: A general topology-based mesh data structure. Int. J. Numer. Meth. Engng. 40(9), 1573–1596 (1997)

14. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design. Academic Press, London (1992)
15. Sahni, O., Muller, J., Jansen, K.E., Shephard, M.S., Taylor, C.: Efficient anisotropic adaptive discretization of the cardiovascular system. Comput. Methods Appl. Mech. Engrg. 195(41-43), 5634–5655 (2006)
16. Wan, J.: An Automated Adaptive Procedure for 3D Metal Forming Simulations. PhD Thesis, Rensselaer Polytechnic Institute, New York (2006)
17. Chevaugeon, N., Hillewaert, K., Gallez, X., Ploumhans, P., Remacle, J.F.: Optimal numerical parameterization of discontinuous Galerkin method applied to wave propagation problems. J. Comput. Physics 223(1), 188–207 (2007)
18. Borouchaki, H., Hecht, F., Frey, P.J.: Mesh gradation control. Int. J. Numer. Meth. Engng. 43(6), 1143–1165 (1998)
19. Sun, D.K., Lee, J.F., Cendes, Z.: Construction of nearly orthogonal Nedelec bases for rapid convergence with multilevel preconditioned solvers. SIAM J. on Sci Comput. 23(4), 1053–1076 (2001)