Engineering with Computers
DOI 10.1007/s00366-010-0179-5

ORIGINAL ARTICLE

Moving curved mesh adaptation for higher-order finite element simulations

Xiao-Juan Luo · Mark S. Shephard ·
Lie-Quan Lee · Lixin Ge · Cho Ng

Received: 15 October 2008 / Accepted: 24 March 2009

Abstract Higher-order finite element method requires valid curved meshes in three-dimensional domains to achieve the solution accuracy. When applying adaptive higher-order finite elements in large-scale simulations, complexities that arise include moving the curved mesh adaptation along with the critical domains to achieve computational efficiency. This paper presents a procedure that combines Bézier mesh curving and size-driven mesh adaptation technologies to address those requirements. A moving mesh size field drives a curved mesh modification procedure to generate valid curved meshes that have been successfully analyzed by SLAC National Accelerator Laboratory researchers to simulate the short-range wakefields in particle accelerators. The analysis results for a 8-cavity cryomodule wakefield demonstrate that valid curvilinear meshes not only make the time-domain simulations more reliable, but also improve the computational efficiency up to 30%. The application of moving curved mesh adaptation to an accelerator cavity coupler shows a tenfold reduction in execution time and memory usage without loss in accuracy as compared to uniformly refined meshes.

Keywords Mesh adaptation · Bézier mesh curving · Higher-order finite elements

X.-J. Luo · M. S. Shephard (✉)
Scientific Computation Research Center,
Rensselaer Polytechnic Institute,
Troy, NY 12180, USA
e-mail: shephard@scorec.rpi.edu

X.-J. Luo
e-mail: xluo@scorec.rpi.edu

L.-Q. Lee · L. Ge · C. Ng
SLAC National Accelerator Lab,
Menlo Park, CA 94025, USA
e-mail: liequan@slac.stanford.edu

L. Ge
e-mail: lge@slac.stanford.edu

C. Ng
e-mail: cho@slac.stanford.edu

1 Introduction

Higher-order finite elements [1], which are well known for the faster rates of convergence in terms of computational efficiency, can provide an effective approach to perform 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 construct such curved 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 proper orders. This approach is able to take advantage of the conventional unstructured mesh generators to deal with the complexity of model geometry. However, the resulting meshes often become invalid because curving the straight-sided 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 researchers in SLAC National Accelerator Laboratory have successfully taken advantage of higher-order finite elements to perform electromagnetic simulations in designing next generation linear accelerators, for example,

Published online: 27 February 2010

short-range wakefield calculations [5–7]. Those simulations require sufficient refinement around a beam region to resolve local high frequencies, while the rest of the domain can have a large mesh size. This refinement region must move along with the beam through the curved domains in the time-dependent simulations to achieve acceptable computational efficiency. When considering that the domains are curved and higher-order finite elements are used, the refined meshes must also be curved to provide a sufficiently geometric approximation to 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 beam domains, while larger mesh size can generate too coarse meshes that often become invalid during the curving procedure. As an example, Fig. 1 shows a beam region (300 μm) in a linear collider (ILC) coupler short-range wakefield simulation whose beam pipe radius is 39 mm. The mesh will have over 100 million tetrahedral elements if the beam size is used to generate uniform-refined mesh. Those lead either to unfeasible large problem size, inaccurate results, or possible failure of the simulations.

To enable such higher-order finite elements in large-scale simulations, a moving curved mesh adaptation procedure that combines a general Bézier mesh curving [8, 9] and size-driven mesh adaptation [10, 11] is presented. The application of curved local mesh modification operations and proper mesh size control are essential in ensuring the resulting curved meshes are valid with the least number of elements for the desired accuracy.

The outline of this paper is as follows. Section 2 describes a Bézier mesh curving procedure to construct valid curvilinear meshes for three-dimensional curved domains. The procedure employs Bézier polynomials to represent the higher-order geometric shapes for curved mesh entities. The extension of size-driven mesh adaptation procedure to account for curved elements is discussed in Sect. 3. Analysis results applied by SLAC for linear accelerator design are shown in Sect. 4.

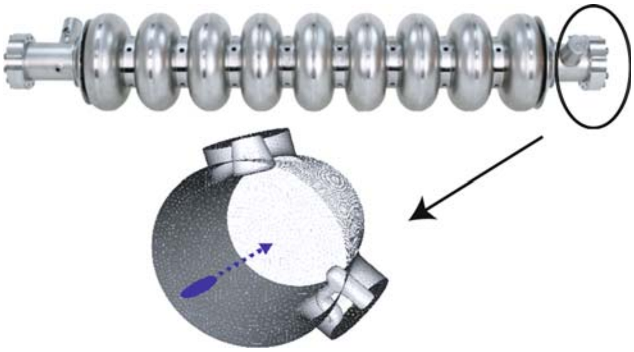


Fig. 1 Beam region in an ILC coupler short-range wakefield simulation

2 Mesh curving

A flexible distributed mesh data structure [12] is employed to support the moving curved mesh adaptation. 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 mesh approximation of the curved geometric domains is maintained by assigning appropriate Bézier higher-order geometric shapes to mesh edges and faces on curved domain boundaries. The topology-based Bézier mesh geometry shape is constructed using Bernstein polynomials that 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 that 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 modification operations to most effectively correct invalid elements due to the curving of mesh entities to the model boundaries. The resulting curved meshes guarantee that each element has positive determinant of Jacobian in its closure that cannot be accomplished by other standard finite element basis, such as Lagrange basis.

2.1 Topology-based Bézier 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],

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

where $B_{|i|}\xi^{|i|}$ 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 edge, face and tetrahedron should be

ξ_1, (ξ_1, ξ_2) and (ξ_1, ξ_2, ξ_3), respectively. Therefore, $\xi_2 = 1 - \xi_1$, $\xi_3 = 1 - \xi_1 - \xi_2$ and $\xi_4 = 1 - \xi_1 - \xi_2 - \xi_3$. $b_{|i|}$ are the control points used to define the curved shapes of the Bèzier mesh edges, faces and regions. Figure 2 shows the control points for a quadratic curved mesh edge, triangle face and tetrahedral region.

Given a straight-sided mesh and its associated geometry CAD model, the control points for those mesh entities on the curved model edges/faces are determined based on the Bézier curve and surface interpolation method by evaluating the model geometry at a set of discrete parametric locations. Common approaches often use the uniformly distributed parametric points. However, alternative methods, such as chord length method or curvature-based procedure, can be used to improve the geometric approximation [14]. For a curved mesh with a different initial shape representation method, the control points are computed by converting the given shapes to Bézier form. As an example, a Lagrange quadratic mesh edge with three interpolating control points $\mathbf{l_1}$, $\mathbf{l_2}$ and $\mathbf{l_3}$ can be converted to a Bézier shape defined as $x = \mathbf{b}_{20}\xi_1^2 + 2\mathbf{b}_{11}\xi_1(1-\xi_1) + \mathbf{b}_{02}(1-\xi_1)^2$. The Bézier control points can be computed as follows:

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

Therefore, $\mathbf{b}_{02} = \mathbf{l}_1$, $\mathbf{b}_{20} = \mathbf{l}_3$ and $\mathbf{b}_{11} = (4\mathbf{l}_2 - (\mathbf{l}_1 + \mathbf{l}_3))/2$. For other forms such as spline/Nurbs shapes, the conversion to Bézier requires careful construction of the mapping equations.

2.2 Validity check of Bézier higher-order curved elements

When applying adaptive higher-order finite element method in which the approximation basis is often increased, the integration rules must be properly improved to ensure that the numerical integration error does not become the dominant error. The improvement of the integration rules requires evaluating the determinant of Jacobian at new integration locations. Without knowledge that the determinant of Jacobian is positive throughout the element closure a priori, the curved elements must compute the determinant of Jacobian at those new locations. In the case that negative determinants of Jacobian occur, the curved elements are invalid and must be corrected. To avoid the constant rechecking of the validity of a curved element for different integration rules, a general algorithm independent of the basis functions, the polynomial orders or the applied integration rules is desired. The convex hull property of Bézier polynomials is used to check the validity of curved elements that ensure the determinant of Jacobian is always positive in the element closures [8].

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

	ξ	$\|i\|$	$B_{\|i\|}$	$\xi^{\|i\|}$
Edge	(ξ_1, ξ_2)	$\|i\| = i + j$	$\frac{q!}{i!j!}$	$\xi_1^i \xi_2^j$
Triangle	(ξ_1, ξ_2, ξ_3)	$\|i\| = i + j + k$	$\frac{q!}{i!j!k!}$	$\xi_1^i \xi_2^j \xi_3^k$
Tet	$(\xi_1, \xi_2, \xi_3, \xi_4)$	$\|i\| = i + j + k + l$	$\frac{q!}{i!j!k!l!}$	$\xi_1^i \xi_2^j \xi_3^k \xi_4^l$

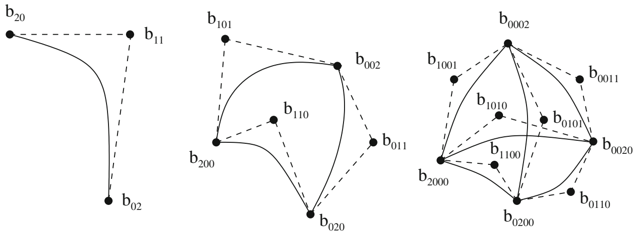


Fig. 2 Bézier control points for a curved mesh edge, face and region

Given a qth order Bézier tetrahedron 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,

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

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

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

where $r = 3(q-1)$. $C_{|i|}$ and $c_{|i|}$ can be expressed using the coefficients $B_{|i|}$ and $b_{|i|}$ in Eq. 1. As an example, the quadratic tetrahedral region shown in Fig. 2 can be expressed as,

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

When 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,

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

The determinant of Jacobian is a cubic Bernstein polynomial and the coefficients $C_{|i|}$ and $c_{|i|}$ 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,

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

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

2.3 Effective procedure to generate valid curvilinear meshes

The mesh curving procedure can start with a straight-sided mesh or a curved mesh with invalid elements. In the case that a straight-sided mesh is given, the procedure computes the Bézier control points for the mesh edges/faces on curved domain boundaries and curves them incrementally. For a given curved mesh with different representation higher-order shapes, 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.

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

$\|i\|$	$C_{\|i\|}$	$c_{\|i\|}$
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	$(a_1 \times b_2 + a_2 \times b_1) \cdot b_3$
–	–	–

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 coefficient, $c_{|i|} \le 0$, as shown in Eq. 5. The key mesh entities are defined as those whose control points appear in the computation of the negative coefficients $c_{|i|}$. As an example, Fig. 3 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 indicate that M_0^0, M_0^1, M_1^1 and M_2^1 are key mesh entities and applying local 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 reshape as shown in Fig. 4, [9]. Comparing to the straight-sided mesh, the validity check algorithm discussed in Sect. 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, $c_{|i|} \le 0$, in computing the determinant of Jacobian;

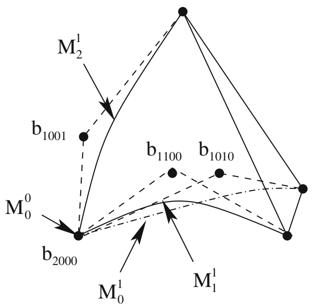
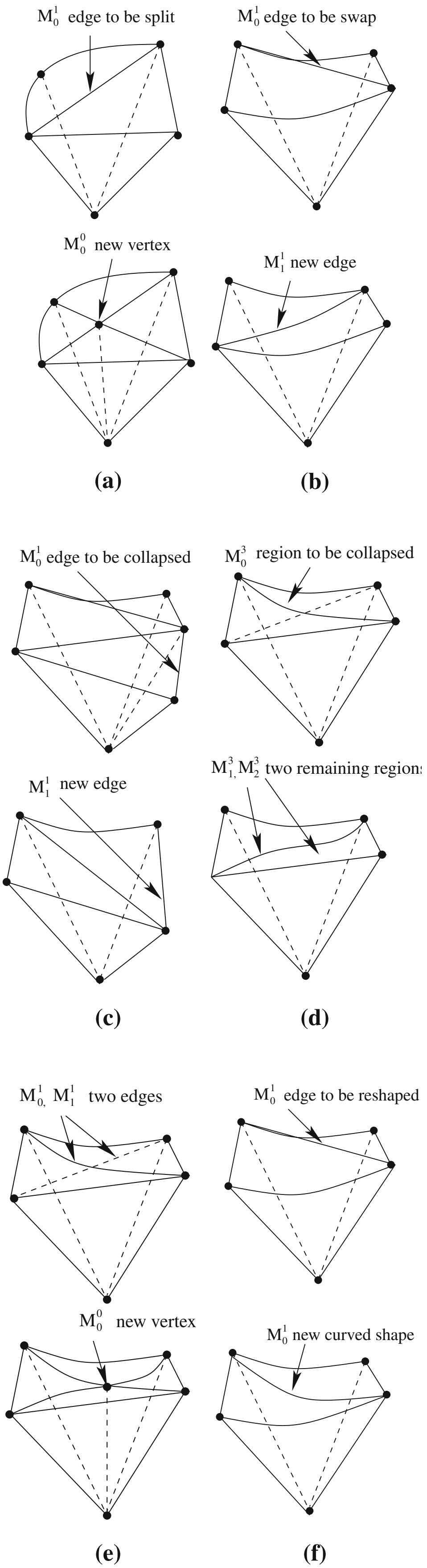


Fig. 3 The computation of det(J) indicates that the mesh entities M_0^0, M_0^1, M_1^1 and M_2^1 are key mesh entities



◀ **Fig. 4** 3D-curved local mesh modification operations. **a** Edge split, **b** edge swap, **c** edge collapse, **d** region collapse, **f** split + collapse, **g** edge reshape

- determine if the invalidity is caused by pairs of neighboring mesh faces or edges classified on the boundary such that angles of 180° are created. In those cases, $c_{lil} \leq 0$ only happens at $i, j, k, l = r$. Apply either split (see Fig. 4a) or swap operations (see Fig. 4b) to introduce additional entities to subdivide those larger angles and 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 $c_{lil} \leq 0$ happens at $i, j, k, l \neq r$. Apply either region split (see Fig. 4d) or split + collapse (see Fig. 4e) to remove the invalid curved element;
- if neither of above two steps is successful, examine the applications of the remaining operations (see Fig. 4c, f) to correct the invalid curved elements;
- if the invalid curved element cannot be corrected using those local mesh operations, refinement is applied and all newly created invalid curved mesh entities will be added to the list to be processed. Subdivision creates more options for applying operations later.

This last refinement step is critical to ensure that all invalid elements can always be corrected as the refinement can produce finer elements and more possibilities to perform local operations. Most of the examples tested by this iterative procedure often only require one or a couple refinements to make the resulting mesh valid. Figure 5 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. Twenty 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 Sect. 4.

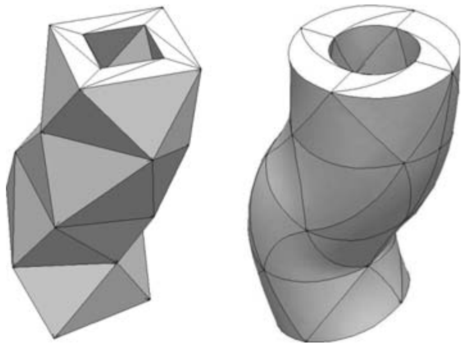


Fig. 5 Straight-sided mesh (*left*) and curved mesh (*right*) for a 3D curved domain

3 Moving curved mesh adaptation 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 that the mesh is curved.

- The validity check algorithm described in Sect. 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, Fig. 6 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.

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 that 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 newly 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 [15–17] 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 beams currently reside. The specification of this refinement information is dictated by the initial locations of the 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 beam domain and the coarse domain can lead to bad quality resulting meshes. The left mesh in Fig. 7 shows an adapted curved mesh which uses sizes 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 that clearly demonstrated that the control of the mesh gradation is needed.

The procedure described in [18] is adopted to define 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

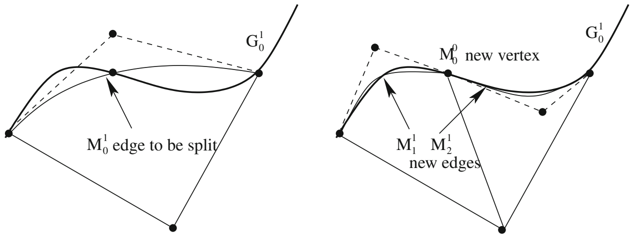


Fig. 6 Before (*left*) and after (*right*) refine of a quadratic curved mesh edge M_0^1 on model edge G_0^1. New mesh edges M_1^1, M_2^1 have been appropriately curved to the model boundaries

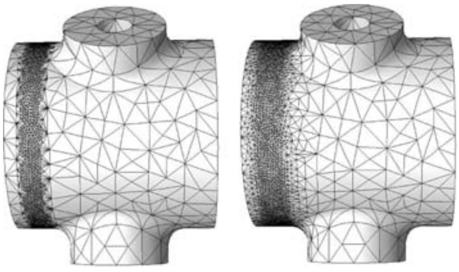


Fig. 7 Curved meshes without (*left*) and with (*right*) mesh size gradation control

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

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,

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

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,

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

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 Fig. 7b, β is adopted as 2.0. More moving adaptive curved meshes are shown in Sect. 4.

4 Analysis results

4.1 Finite-element time-domain method for electromagnetics

A brief introduction of finite-element time-domain (FETD) method for electromagnetic simulation is given in this section. Ampere's and Faraday's laws along with constitutive relations yield the inhomogeneous vector wave equation for the electric field,

$$\nabla\times\left(\frac{1}{\mu}\nabla\times\mathbf{E}\right)+\varepsilon\frac{\partial^2}{\partial t^2}\mathbf{E}=-\frac{\partial}{\partial t}\mathbf{J} \tag{13}$$

To avoid time differentiation of electric current density $\mathbf{J}$, it can be integrated in time to obtain the following equation,

$$\nabla\times\left(\frac{1}{\mu}\nabla\times\int_{-\infty}^{\mathbf{t}}\mathbf{E}\mathrm{d}\tau\right)+\varepsilon\frac{\partial^2}{\partial t^2}\int_{-\infty}^{\mathbf{t}}\mathbf{E}\mathrm{d}\tau=-\mathbf{J} \tag{14}$$

where $\mathbf{E}$ is the electric field intensity, $\mathbf{J}$ is the electric current density, and ε and μ are the electric permittivity and magnetic permeability.

With finite-element spacial discretization, $\int_{-\infty}^{t}\mathbf{E}\mathrm{d}\tau$ in Eq. 14 is expanded by a set of hierarchical Nedelec [19] basis functions $\mathbf{N}_i(\mathbf{x})$,

$$\int_{-\infty}^{t}\mathbf{E}(\mathbf{x},\tau)\mathrm{d}\tau=\sum_i e_i(t)\cdot\mathbf{N}_i(\mathbf{x}) \tag{15}$$

The vector wave equation is discretized to a set of second-order ordinary differential algebraic equations,

$$\mathbf{M}\frac{\mathrm{d}^2\mathbf{e}}{\mathrm{d}t^2}+\mathbf{Ke}=\mathbf{f} \tag{16}$$

where matrices $\mathbf{M}$, $\mathbf{K}$, and vector $\mathbf{f}$ are

$$\mathbf{M}_{ij}=\int_\Omega\varepsilon\mathbf{N}_i\cdot\mathbf{N}_j\mathrm{d}\Omega \tag{17}$$

$$\mathbf{K}_{ij}=\int_\Omega\frac{1}{\mu}(\nabla\times\mathbf{N}_i)\cdot(\nabla\times\mathbf{N}_j)\mathrm{d}\Omega \tag{18}$$

$$\mathbf{f}_i=\int_\Omega\mathbf{N}_i\cdot\mathbf{J}\mathrm{d}\Omega \tag{19}$$

The Newmark-beta scheme, which is unconditionally stable when beta is larger than or equal to 0.25, is used to approximate the above second-order differential equations of field Eq. 16. The resulting implicit time marching scheme is given as follows,

$$\begin{aligned}&\left(\mathbf{M}+\beta(\Delta t)^2\mathbf{K}\right)\mathbf{e}^{n+1}=\left(2\mathbf{M}-(1-2\beta)(\Delta t)^2\mathbf{K}\right)\mathbf{e}^n\\&-\left(\mathbf{M}+\beta(\Delta t)^2\mathbf{K}\right)\mathbf{e}^{n-1}-(\Delta t)^2\left(\beta\mathbf{f}^{n+1}+(1-2\beta)\mathbf{f}^n+\beta\mathbf{f}^{n-1}\right)\end{aligned} \tag{20}$$

Note that the electric field $\mathbf{E}$ and the magnetic flux density $\mathbf{B}$ are then easily obtained from the solution vector $\mathbf{e}$,

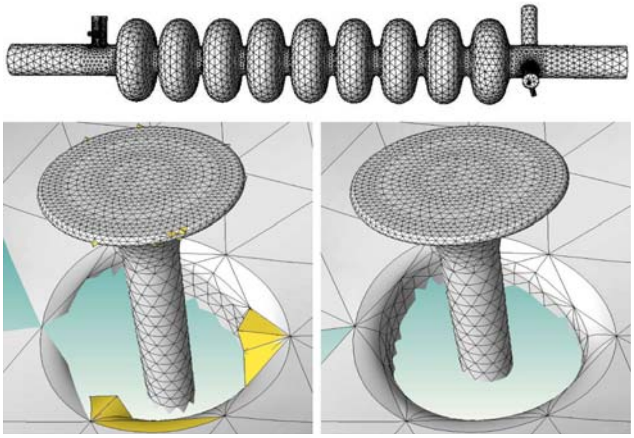


Fig. 8 The mesh for one cavity (*top*), close-up mesh before (*bottom left*) and after (*bottom right*) correcting the invalid curved elements marked as *yellow*

$$\mathbf{E}(\mathbf{x}) = \Sigma_i \partial_t \mathbf{e}_i \mathbf{N}_i(\mathbf{x}) \quad (21)$$

$$\mathbf{B}(\mathbf{x}) = -\Sigma_i \mathbf{e}_i \nabla \times \mathbf{N}_i(\mathbf{x}) \quad (22)$$

4.2 Valid curvilinear meshes for FETD electromagnetic simulations

SLAC performs simulations for the wakefield effects of an eight-cavity cryomodule for the proposed International 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 that have been converted to Bézier representations using Eq. 2. 515 invalid curved elements were detected and have been corrected using the procedure discussed in Sect. 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 8 shows the curved mesh for one cavity of the model and the close-up mesh before and after curving. Figure 9 shows how an edge collapse operation is applied to correct an invalid curved element 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 were generated. Figure 10 shows a snapshot of the electric field distribution excited by a beam in the ILC cryomodule.

The statistics for correcting the invalid curved regions is presented in Table 3. The data shows that the procedure used about 10 min 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.

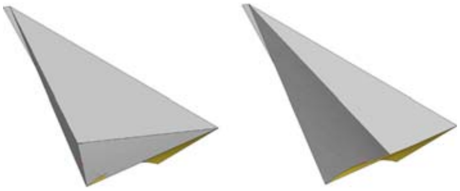


Fig. 9 Local mesh cavity before (*left*) and after (*right*) applying edge swap to correct the invalid element

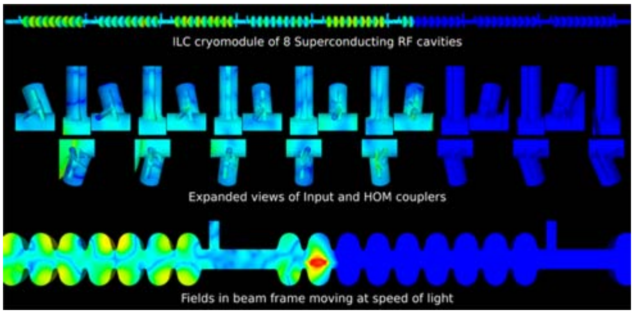

ILC cryomodule of 8 Superconducting RF cavities
Expanded views of Input and HOM couplers
Fields in beam frame moving at speed of light

Fig. 10 A snapshot of the electric field distribution excited by a beam in an eight-cavity cryomodule for the proposed international linear collider

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

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

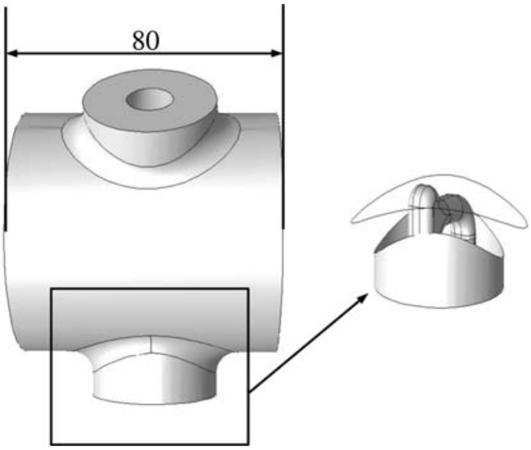


Fig. 11 Geometric model for the short-range wakefield simulation

4.3 Moving curved mesh refinements for short-range wakefield calculations

A series of moving adapted meshes in a curved domain were generated using the procedure described in Sect. 3 for short-range wakefield calculations by SLAC. Figure 11 shows the geometric model that has some complex components in the

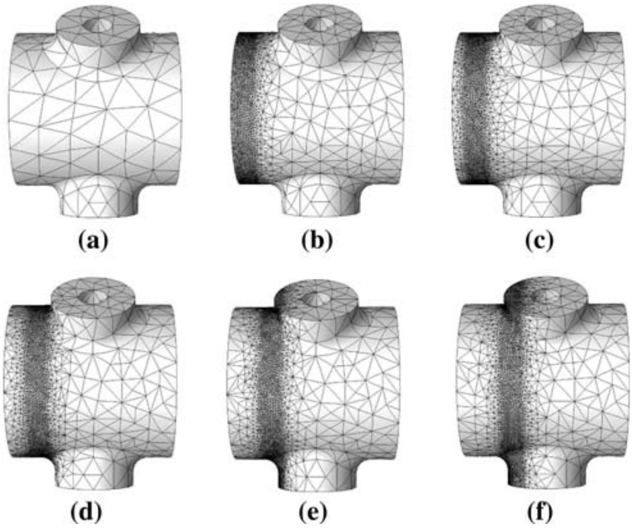


Fig. 12 Moving adapted meshes in curved domain for short-range wakefield simulation. **a** Initial mesh, **b** step 1, **c** step 2, **d** step 3, **e** step 4, **f** step 5

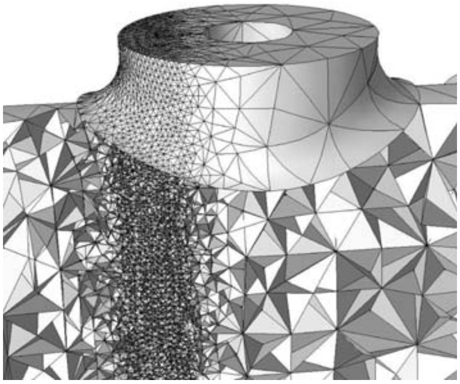


Fig. 13 Interior mesh for the adapted mesh at step 5

middle of the domain. The initial location of the beam is at the left end of the domain, the desired mesh size inside beam region is 1 and the size for the rest of the domains is 10. Figure 12 shows the moving adapted curvilinear meshes up to step 5. Figure 13 shows the interior adapted mesh at step 5. The Mesh size gradation control procedure discussed in Sect. 3 is applied with $\beta\ = 2.0$. Figure 15 shows the number of elements at each step that indicates that these adaptively refined meshes have around $1-1.15$ million elements when compared with the uniform refined mesh with 6.5 million elements if the mesh size inside the beam domain is applied in the entire domain. The increase in the number of elements in the middle of the domain is due to the complex geometries as shown in Fig. 11. Figure 14 shows the short-range wakefiled simulation using the moving curved adapted meshes. The results show a tenfold reduction in execution time and memory usage without loss in accuracy as compared to uniformly refined meshes.

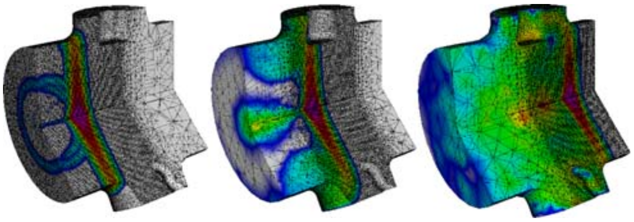


Fig. 14 Short-range wakefield simulation results with moving curved mesh adaptation

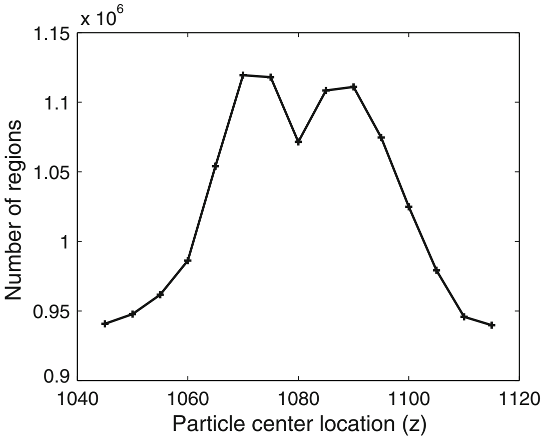


Fig. 15 Number of mesh regions at each step

5 Closing remarks

This paper has presented a procedure to track moving adaptive mesh refinement in curved domains that 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 US Department of Energy under DOE Grant number DE-FC02-06ER25769 and DE-AC02-76SF00515.

References

1. Babuska I, Szabo BA, Katz IN (1981) The p-version of the finite element method. Int J Numer Meth Eng 18(3):515–545
2. Luo XJ, Shephard MS, Remacle JF, O'Bara RM, Beall MW, Szabo BA, Actis R (2002) p-version mesh generation issues. In: Proceedings of eleventh meshing roundtable, Sandia National Labs, Ithaca, NY, pp 343–354
3. Simmetrix Inc. Enabling simulation-based design. http://www.simmetrix.com
4. CUBIT geometry and mesh generation toolkit (2008). http://cubit.sandia.gov

5. Akcelik V, Ko K, Lee LQ, Li ZH, Ng CK, Xiao LL (2008) Shape determination for deformed electromagnetic cavities. J Comput Phys 227(3):1722–1738
6. Xiao L, Adolphsen C, Akcelik V, Kabel A, Ko K, Lee LQ, Li Z, Ng CK (2007) Modeling imperfection effects on dipole modes in TESLA cavity. In: Proceedings of 2007 particle accelerator conference, Albuquerque, NM
7. Lee LQ, Akcelik V, Chen S, G LX, Prudencio E, Schussman G, Uplenchwar R, Ng C, Ko K, Luo XJ, Shephard MS (2007) Enabling technologies for petascale electromagnetic accelerator simulation. J Phys Conf Ser 78:012040
8. Luo XJ, Shephard MS, Obara RM, Nastasia R, Beall MW (2004) Automatic p-version mesh generation for curved domains. Eng Comput 20:265–285
9. Luo XJ (2005) An automatic adaptive directional variable p-version method in 3D curved domains. PhD thesis, Rensselaer Polytechnic Institute, New York
10. Li XR, Shephard MS, Beall MW (2003) Accounting for curved domains in mesh adaptation. Int J Numer Meth Eng 150:247–276
11. Li XR, Shephard MS, Beall MW (2005) 3D anisotropic mesh adaptation by mesh modification. Comp Meth Appl Mech Eng 194:4915–4950
12. Seegyoung E, Shephard MS (2006) Efficient distributed mesh data structure for parallel automated adaptive analysis. Eng Comput 22:197–213
13. Beall MW, Shephard MS (1997) A general topology-based mesh data structure. Int J Numer Meth Eng 40(9):1573–1596
14. Farin G (1992) Curves and surfaces for computer aided geometric design. Academic Press, New York
15. Sahni O, Muller J, Jansen KE, Shephard MS, Taylor CA (2006) Efficient anisotropic adaptive discretization of the cardiovascular system. Comput Methods Appl Mech Eng 195(41–43):5634–5655
16. Wan J (2006) An automated adaptive procedure for 3D metal forming simulations. PhD thesis, Rensselaer Polytechnic Institute, New York
17. Chevaugeon N, Hillewaert K, Gallez X, Ploumhans P, Remacle JF (2007) Optimal numerical parameterization of discontinuous Galerkin method applied to wave propagation problems. J Comput Phys 223(1):188–207
18. Borouchaki H, Hecht F, Frey PJ (1998) Mesh gradation control. Int J Numer Meth Eng 43(6):1143–1165
19. Sun DK, Lee JF, Cendes Z (2001) Construction of nearly orthogonal Nedelec bases for rapid convergence with multilevel preconditioned solvers. SIAM J Sci Comput 23(4):1053–1076