Isogeometric High Order Mesh Generation

Teseo Schneider[a], Daniele Panozzo[b], Xianlian Zhou[c,*]

[a] *Department of Computer Science, University of Victoria, Victoria BC V8P 5C2, Canada*
[b] *Courant Institute of Mathematical Sciences, New York University, New York City, NY 10012, USA*
[c] *Department of Biomedical Engineering, New Jersey Institute of Technology, Newark, NJ 07103, USA*

Abstract

In this paper, we present a new posteriori method to generate high order curved meshes directly from linear meshes through a recently developed poly-spline isogeometric (IGA) method. Given an input structured or unstructured linear quadrilateral or hexahedral mesh, our method constructs quadratic poly-spline IGA bases for each element and ensures continuity and smoothness across neighboring elements. With these smooth IGA bases, each element can be sampled into a high order element of arbitrary order while maintaining consistently continuous and smooth interfaces between elements. Several fitting algorithms are developed to adjust the IGA control points to ensure the final mesh interpolates or approximates the initial boundary closely. Our method requires no CAD geometry of the initial mesh and guarantee C^1 smoothness in regular regions and C^0 continuity across interfaces of spline-incompatible elements. The method is compared with an elastic analogy based approach and is shown to be superior in terms of computational performance and overall mesh quality and smoothness. The current method can also be extended for higher order IGA basis construction and smooth refinement of meshes that are important for high order physics simulations.

Keywords: High Order Mesh Generation, Isogeometric Analysis, Poly-spline Finite Element Method, Elastic Analogy

1. Introduction

High order discretization and numerical methods are garnering considerable attention from the scientific computation community due to their potential to achieve high order accuracy in solving various physics problems. Compared to traditional linear or low order meshes and numerical schemes, high order approaches tend to achieve higher accuracy with reduced computational cost in certain scenarios [1]. Within the structure or solid mechanics community, high order Galerkin method with finite element discretization [2, 3, 4] is very popular, whereas within the computational fluid dynamics (CFD) and electromagnetic dynamics

*corresponding author
Email addresses: teseo@uvic.ca (Teseo Schneider), panozzo@nyu.edu (Daniele Panozzo), alexzhou@njit.edu (Xianlian Zhou)

communities the preferred method is the high order Discontinuous Galerkin (DG) method [5, 6, 7, 8, 9]. In general, high order methods are easier to implement on structured quadrilateral or hexahedral meshes due to the regularity of element arrangement. For unstructured meshes, high order methods are more challenging but still offer a key to the improvement of simulations over complex geometric configurations, such as high lift systems and landing gear [7], turbulence and transition [10].

One of the main challenges of high order methods is the difficulty in generating high order meshes, as noted by Vincent and Jameson [11], which limited the adoption of high order techniques in both academia and industry. Generating unstructured high order meshes has been a focal point of research in tackling this challenge. There are two main categories of methods for high order mesh generation: direct methods [12] and posteriori methods [13, 14, 15, 16]. The direct method takes as input CAD models and generates high order meshes directly. As noted by Geuzaine et al. [17], robust high order mesh generation procedures are far from being trivial: snapping high order vertices on the CAD geometry may cause elements to become invalid and complex mesh untangling procedures may be needed to restore mesh validity. For example, Toulorge et al. [18] developed a technique using constrained optimization where element Jacobians are constrained to lie in a prescribed range to untangle high order meshes. On the other hand, the posteriori method takes as input a high-quality linear (low order) mesh that was already generated and then add high order nodes on top of it. The simplest approach is to add the high order nodes on the straight linear edges or faces of the low order elements. However, high order meshes generated in this way are not geometrically smooth and cannot fully take advantage of the setup and accuracy of high order numerical schemes[19]. Many of the previous studies [20, 21, 5, 6] have confirmed that physically curved elements that approximate the curved domain boundaries adequately are required to take advantage of the accuracy of high order methods.

To generate physically curved high order elements, many posteriori methods have been proposed. Some methods are based on local or global modification of geometric entities. For example, Fortunato et al. [22] developed a high order unstructured curved mesh generation by solving the Winslow equations within the entire domain. Jiao and Wang [23] developed a weighted averaging of local fittings (WALF) for high order surface meshing. A free program, meshCurve, implemented by Ims et al. [15] uses an algorithm developed by Jiao and Bayyana [24] to detect feature curves of the linear mesh and then reconstructs the high order mesh using the WALF algorithm. Karman et al. [25] proposed a weighted condition number (WCN) method for mesh optimization to generate high order meshes by splitting high order elements into a collection of linear sub-elements and perturbing node vectors for the new surface nodes and finally smoothing the mesh. This method was shown to correctly mesh examples where some other methods failed, including mean-value coordinate interpolation and radial basis function based smoothing techniques.

Different from the above geometry-based approaches, another line of research sought to move the high order nodes based on Lagrangian solid mechanics [26] or elastic analogy. Poya et al. [16] developed a posteriori high order curved mesh generation approach using boundary node surface projection and solid

mechanics for interior deformation. The method imposes boundary displacement on high order nodes that are projected onto the true CAD surface and deform the internal nodes based on elasticity. More recently, Yang et al. [19] developed a posteriori high order mesh generation method through moving the boundary high order nodes to the smooth boundary subdivision curves [27] or surfaces [28, 29] and solving the deformation of interior mesh with elasticity as well. Unlike the Poya's method, the method by Yang et al.[19] requires no CAD surface of the initial geometry and thus provides more flexibility in curving lower order meshes.

The posteriori elastic analogy approach not only offers a unique advantage over the geometry based methods, as it guarantees no tangling or inversion of elements when the elasticity solution converges, but it also comes with a few disadvantages. Often noted, the elastic analogy approach produces smooth elements around the boundary, but the overall smoothness quickly fades away from the boundary since the deformation energy is absorbed mostly by the local areas. Another disadvantage is the computation time required for solving a solid mechanics problem. For example, using the finite element method (FEM) to solve elastic deformation of a large mesh is computationally intensive and sometimes could be impractical.

In this paper, we present a new posteriori method to generate high order curved meshes. Our method is based on the recently developed poly-spline FEM [30] which employs a novel poly-spline isogeometric (IGA) [4] interpolation. The current method takes as input linear or low order meshes (structured or unstructured quadrilateral or hexahedral meshes) and constructs quadratic poly-spline IGA bases for each element. In this way, C^1 continuity and smoothness across neighboring elements are ensured for regular grid-like regions and C^0 continuity is ensured across interfaces of spline-incompatible elements. Within each IGA element, a regular high order element of arbitrary order can be generated by simply sampling the element's parametric space and the shared (interface) nodes between elements are guaranteed to be coincident. In what follows, we first present the basics of poly-spline IGA interpolation and corresponding high order mesh generation procedure, and then describe several fitting algorithms to adjust the IGA control points to ensure the final mesh interpolate or approximate the initial boundary closely. We further demonstrate the effectiveness of this method with several 2D and 3D examples and compare it with an elastic analogy based approach in terms of computational performance and overall mesh smoothness. Lastly, we discuss the advantages and limitations of this method and potential future directions.

2. Method

2.1. Isogeometric interpolation

Isogeometric analysis [31, 4, 3] is a finite element analysis framework that employs the shape functions used to describe the geometry of the computational domain as the interpolation functions for the finite element field variable. The geometrical shape functions often use the B-Splines or Non-Uniform B-Splines (NURBS) bases for exact and efficient geometry representation in CAD. The physical representation of the

geometry is a mapping from the parametric space (where the spline bases are defined) to the physical space through linear combination of control points and spline bases. These spline bases serve as the shape functions and have global high order smoothness despite compact local supports. IGA analysis directly integrates CAD systems with high order analysis to improve the solution accuracy and reduces the computational cost [4]. IGA analysis is relatively easy to define and implement for fully regular (lattice like block or multi-block structure) grids but challenging to extend to geometrically complex or high-genus models. It is still considered as an open problem to generate IGA-compatible meshes from a given general boundary surface [32, 33]. New development tends to use more advanced geometrical representation, such as T-Splines to handle geometry of complex topology [34, 35, 36] and S-Splines to simplify local refinement [37]. However, these new geometry representations are more complicated than NURBS and have not yet been widely adopted by both academia and industry.

Recently, a novel poly-spline FEM (PolyFEM) has been developed by Schneider et al. [30] to address the difficulties of generating IGA-compatible meshes for solids with arbitrary boundary surfaces. Compared to existing methods on converting unstructured quadrilateral and hexahedral meshes to T-splines for IGA applications [38, 39], the PolyFEM method shares a similar goal but uses standard spline bases. These spline bases are easier to construct and widely supported in existing modeling systems. In additional, IGA implementation in PolyFEM with the standard spline bases, instead of the more complex T-splines, is simpler and more straightforward.

The PolyFEM method takes as input a linear mesh as the starting point for high order IGA basis construction. In 3D, the linear mesh could be a mix of regular hexahedral elements and polyhedral elements, such as those generated with hex-dominate meshers [40, 41]. Well-shaped regular elements get the simplest and most efficient basis functions, while complex polyhedral elements used to handle the transitions between regular regions are formulated with harmonic bases. These regular elements can be spline compatible or incompatible. For a spline compatible element, such as a hexahedral element with triquadratic $(3 \times 3 \times 3)$ neighbors, only a single node is introduced per element. For an incompatible hexahedral element, standard or conventional quadratic element construction is used. That is, 27 local nodes are introduced within an incompatible hexahedral element. To ensure continuity between the neighboring spline compatible and incompatible elements, constraints are introduced between the spline control points and conventional element high order nodes [30]. Consequently, some of these local element nodes can be eliminated or treated as virtual nodes in the implementation. Formally, for every hexahedra H in a spline compatible region, we define the basis functions as the standard triquadratic spline basis with uniform knot vectors. For spline compatible boundary hexahedra we increase the knot multiplicity and add additional bases on the faces/edges/vertices of H. For instance, if H is a corner element, we will add one node on the three boundary faces, one node on the three boundary edges, and one node on the vertex and increase the knot multiplicity in the boundary direction. The geometric mapping of spline-compatible elements is defined as the linear combination of the

nodes' positions and their bases. PolyFEM suggests placing nodes in the barycenters (i.e., cell barycenter for internal nodes and face/edge barycenter for boundary elements); in this work, we focus on a better placement of such nodes. For the spline incompatible region, we will be using standard triquadratic Lagrange bases with 27 nodes per elements. The main difficulty of this construction is in the transition between the regions where the bases (i.e., transition form a smooth spline to a Lagrange basis) is not C^0. Schneider et al. [30] suggest constraining the Lagrange nodes to ensure that they match the spline side. Practically, this is achieved by changing the standard local-to-global mapping for the nodes in the interface: the local Lagrange nodes on the interface will be mapped to unique global nodes with a specified weight, thus ensuring at least C^0 continuity of the basis.

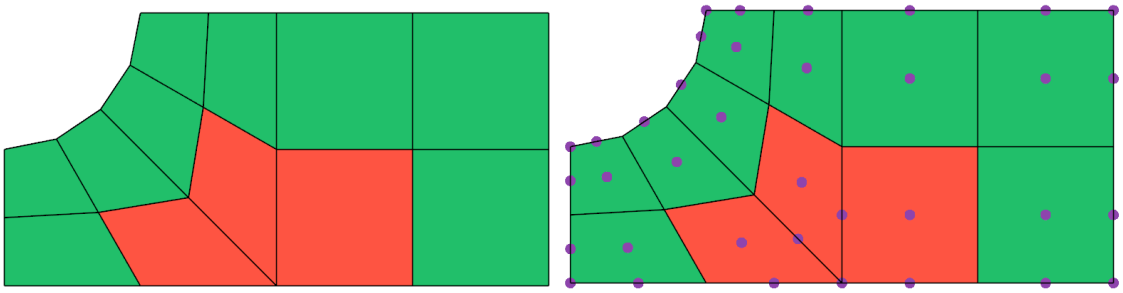
We note that the decreased smoothness of transition from spline elements to traditional high order elements. These transitional regions usually occupy a small portion of the volume, thus their effects do not have a major impact on the solution quality. For a mesh with only spline compatible elements, PolyFEM is a full IGA method. Since only one (center) node is used for each IGA element, the total number of nodes is in the same order as that of the initial linear mesh. For brevity, we refer to the PolyFEM bases/elements/method as the IGA bases/elements/method hereafter. In what follows, we will focus on mesh examples dominated by spline compatible (IGA) elements without arbitrary polyhedrons.

In Fig. 1, we show an example of a 2D IGA mesh. The mesh is a quarter of a plate geometry that contains a circular hole in the middle. The elements in green are the spline compatible IGA elements and the ones in red are the spline incompatible elements. For each IGA element and its neighbors, a cell center node is introduced to form a regular 3×3 lattice. All IGA elements in this example are also boundary elements, additional control points are introduced on boundary edges to maintain 3×3 structures, ensure biquadratic polynomial reproduction, and approximate the boundaries correctly. For each incompatible element, 9 (3×3) local nodes are introduced within the element, but some nodes at the IGA element interfaces are eliminated due to the continuity constraints. These elements also have their own independent center nodes that are coincident with those introduced during the construction of IGA bases for their IGA neighbors. In fact, the displayed center node of each spline incompatible element covers two nodes at the same initial position.

2.2. Sampling based high order mesh generation

Once the biquadratic (2D) or triquadratic (3D) IGA element bases are constructed, it is straightforward to generate high order meshes of an arbitrary order by sampling of each element's parametric space. In 2D, a quadrilateral element has a parametric space $\mathbf{u} = \{u, v\}$ with $0 \leq u, v \leq 1$. To generate a regular quadratic element, we can sample the space with the following 9 parametric coordinates

$$\{\{0,0\},\{1,0\},\{1,1\},\{0,1\},\{0.5,0\},\{1,0.5\},\{0.5,1\},\{0,0.5\},\{0.5,0.5\}\}. \tag{1}$$



(a) (b)

Figure 1: A quarter of the plate-hole mesh. (a) The initial linear mesh. (b) The initial placement of the IGA control points (purple dots) are overlaid on the linear mesh.

where the first four are the corners, the middle four are at the middles of the edges, and the last one is the element or cell center.

To find the physical nodal positions $\mathbf{\Phi}$ of these sampling coordinates, we can evaluate them with the following equation:

$$\mathbf{\Phi}(\mathbf{u}_i) = \sum_{k=1}^{nc} B_k(\mathbf{u}_i)\mathbf{C}_k \tag{2}$$

where nc is the number of element nodes (control points for IGA elements) and B_k is the basis function of the $k-$th node $\mathbf{C}_k$ evaluated at the parametric coordinate $\mathbf{u}_i$ $(i = 1, ..., ns)$ with ns denoting the number of sampling positions. For a regular quadratic IGA element, $nc = 9$, $\mathbf{C}_k$ are the control points including the central node of the current element and its eight neighbors, and B_k are the biquadratic spline basis functions. A similar procedure can be used to generate meshes of higher orders with dense sampling of the parametric space, and the procedure works for both 2D and 3D meshes.

2.3. Automatic fitting with least square

Because of the non-Kronecker delta property of B-Spline basis functions, the evaluated curve or surface of the boundaries will not necessarily interpolate the linear nodes based on the initial placement of IGA control points. For a very fine initial linear mesh, this issue is not apparent due to relatively small errors. However, for a coarse initial mesh, the error can be considerable (e.g. see Fig. 2e). For posteriori high order mesh generation, it is often desired to have the smooth boundaries interpolating or approximating the initial linear mesh nodes as close as possible. To correct this non-interpolating issue, a fitting algorithm was developed to optimize the positions of IGA nodes such that the differences between the evaluated smooth boundaries and the linear boundaries is minimized. The algorithm aims to preserve the validity of the mesh, ensuring that a valid curved mesh with positive Jacobians is generated as long as the input mesh is also valid.

Considering one IGA element, we evaluate the position of the selected sampling coordinates ($\mathbf{u}_i$) and aim to bring these positions closer to their linear counterparts ($\mathbf{P}_i$). For this element, the overall difference between the evaluated positions and corresponding linear positions is described by

$$d = \sum_{i=1}^{ns} w_i |\mathbf{\Phi}(\mathbf{u}_i) - \mathbf{P}_i|^2 \tag{3}$$

where w_i is a weight associated with $i-th$ sampling position to indicate its importance. For example, higher weights can be assigned to nodes that on the boundary. Since the evaluation of $\mathbf{\Phi}(\mathbf{u}_i)$ depends on the central nodes from both the current element and its neighboring elements, minimizing this difference must be considered in a global setting. Therefore, we sum all differences as

$$D = \sum_{i=1}^{ne} d_i \tag{4}$$

where ne is the total number of elements and d_i is the difference defined in Eq. 3 for element i.

Regarding the sampling, the simplest choice will be the corner nodes, since the goal is to interpolate all corresponding linear nodes. However, according to our tests, choosing corner nodes alone sometimes may give oscillatory solutions with high curvature in the middle of the element borders despite a good fitting of its corners. Considering this, we include the middles nodes (thus the full list of sampling points in Eq. 1 for 2D) into the equation. Since the linear middle nodes resides on the straight edges or faces of the linear mesh, it is undesired to be tightly fitted to these nodes. Therefore, we assign them lower weights than those of the corner nodes. Increasing the weights for the middle nodes from zeros to larger numbers will reduce the curvature oscillation greatly, but will also gradually move the evaluated IGA nodes away from the linear corner ones (by a small amount).

We also developed an alternative frame rotation based method, which aims to explicitly eliminate the oscillatory curvature, to avoid ad-hoc weights adjustment. Since curvature is associated directly with the rotation of local frames derived from the geometrical (curvilinear) mapping, we can define an objective that minimizes the rotation of local frames. First, at each sampling point, a local frame consisted of curvilinear basis vectors, defined as the partial derivatives of the spatial IGA mapping, is created. These basis vectors are not necessarily orthogonal to each other or unit vectors. In computational mechanics literature [42, 43], these curvilinear basis vectors are used directly to define strains associated with stretching, bending, and shear and the elastic energy associated with them. We choose not to normalize and orthogonalize these vectors into conventional frames since doing so will also introduce dependence on mesh density for the objective function to be introduced soon. For the linear mesh, the nodal frames on the element interfaces are discontinuous and cannot be determined unequivocally. Therefore, we use the frames defined on the initial IGA mesh (before the optimization) which are smooth and continuous across elements (thus unique at any sampling position). The objective function to minimize the frame rotation within each element is

defined as

$$r = \sum_{i=1}^{ns} \left(w_i \sum_{j=1}^{dim} |(\mathbf{\Phi}_{,j}(\mathbf{u}_i) - \mathbf{\Phi}^0_{,j}(\mathbf{u}_i)|^2 \right) \tag{5}$$

where the superscript 0 denotes the initial geometry, and the subscript $,j$ denotes the partial derivative along the $j\text{-}th$ coordinate direction, that is,

$$\mathbf{\Phi}_{,j}(\mathbf{u}_i) = \sum_{k=1}^{nc} B_{k,j}(\mathbf{u}_i)\mathbf{C}_k \tag{6}$$

where $B_{k,j}$ is the derivative of the spline basis function B_k with respect to the $j\text{-}th$ coordinate.

Again, in a global setting, we sum all differences of local frames as

$$R = \sum_{i=1}^{ne} r_i. \tag{7}$$

The frame rotation difference has to be used in combination with the position difference, and the combined objective is written as

$$Q = D + R = \sum_{j=1}^{ne} \left(\sum_{i=1}^{ns} \left(w_i^d |\mathbf{\Phi}(\mathbf{u}_i) - \mathbf{P}_i|^2 + w_i^r \sum_{j=1}^{dim} |\mathbf{\Phi}_{,j}(\mathbf{u}_i) - \mathbf{\Phi}^0_{,j}(\mathbf{u}_i)|^2 \right) \right) \tag{8}$$

where w_i^d and w_i^r are weights for the position and frame differences, respectively.

Minimization of Q can be solved with a least square method, which results in a linear system of equations

$$\mathbf{A}^T\mathbf{A}\mathbf{C} = \mathbf{A}^T\mathbf{B}, \tag{9}$$

where

$\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_{nn}\}^T$ are the unknown positions and nn is the total number of nodes,

$\mathbf{B} = \{\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_{ne}\}^T$ with

$\mathbf{B}_i = \{w_1^d\mathbf{P}_1, ..., w_{ns}^d\mathbf{P}_{ns}, w_1^r\mathbf{\Phi}^0_{1,1}, ..., w_1^r\mathbf{\Phi}^0_{1,dim}, ..., w_{ns}^r\mathbf{\Phi}^0_{ns,1}, ..., w_{ns}^r\mathbf{\Phi}^0_{ns,dim}\}^T$, and

$\mathbf{A}$ is a sparse matrix filled with quantities such as $w_i^d B_k$ and $w_i^r B_{k,j}$ at proper locations. Solving this linear system of equations (Eq. 9) will result in the new positions of the IGA mesh nodes ($\mathbf{C}$).

The automatic fitting procedure obtained by minimizing the Q in Eq. 8 cannot guarantee mesh validity. In some cases, a mesh generated from this procedure may create some degenerated elements whose Jacobian are negative. To ensure mesh validity, we propose to conduct element Jacobian check after fitting to ensure positive Jacobian for all elements. If there are elements found with zero or negative Jacobian, a bi-section line search will be performed to find the maximum allowed nodal displacement along the direction computed from the initial fitting. By doing so, all positive Jacobians can be guaranteed if the initial mesh contains no elements with zero or negative Jacobian. The examples presented in the result section were all obtained with this method.

2.4. Additional element degeneration energy

While the above method ensures validity, it does so by sacrificing mesh quality, as the filtering criteria are not directly represented in the energy function. To further increase mesh quality, we propose to add an additional element degeneration energy to Q in Eq. 8 such that

$$Q = D + R + \sum_{i=1}^{ne} S_i \tag{10}$$

where S_i is the element degeneration energy for the $i-th$ element. One choice of such an energy is the symmetric Dirichlet energy [44] defined as

$$S = 2\frac{||\det \mathbf{J}||^2}{\det^2 \mathbf{J}} \tag{11}$$

where $\mathbf{J}$ is the geometric Jacobian of the element. The energy increases as the Jacobian approaches zero, diverges at $\det \mathbf{J} = 0$ as it cannot be evaluated, and is assumed to be undefined when $\det \mathbf{J} < 0$. Therefore, during energy evaluation, we shall guarantee the mesh is valid with $\det \mathbf{J} > 0$ for all elements. To ensure that, we always perform a Jacobian check and a line search if necessary before the energy evaluation. Note, the Jacobian is defined between the spatial domains of the deformed element and its initial element (instead of the canonical parametric element). As a result, the energy will be scale or mesh density independent and can preserve boundary layer elements with high aspect ratios.

With the added element degeneration energy, minimization of Q in Eq. 10 can no longer be solved with the least square method due to the nonlinear nature of the element degeneration energy. To solve this unconstrained minimization problem, we used both limited-memory BFGS (LBFGS) method and a projected Newton method. However, both methods are much slower than the least square method because they typically take many iterations to converge. And within each iteration, one or more steps of line search may be needed to guarantee positive Jacobian of the intermediate meshes. We also utilized these two methods on the examples presented in the result section below. However, there are no observable differences in examples of generated high order meshes presented in this paper.

The projected Newton method typically converges with fewer iterations than the LBFGS method. However, it requires evaluation of both gradient and Hessian of the energy function, which is not trivial. We utilized an automatic differentiation method to compute the gradient and Hessian, which is more time-consuming when compared to analytical differentiation. To speed up the computation, one may consider a local confined optimization to minimize the element degeneration energy. That is, one can consider moving the position of one vertex at a time while keeping the rest of the mesh fixed. This strategy allows good parallelization, as we can do graph coloring on the mesh and then optimize one color at a time in parallel.

2.5. Elastic analogy based mesh generation

For comparison purposes, we also developed an elastic analogy based high order mesh generation approach. For a linear mesh, we first generate a regular high order mesh by sampling the linear elements,

which produces high order nodes on the straight edges (2D) or faces (3D). The elastic analogy method moves the high order nodes on the boundaries to reconstructed smooth boundary curves or surfaces, and then deforms the interior nodes based on the minimization of elastic energy. Different from other elastic analogy method reviewed earlier, the reconstructed smooth boundary curves or surfaces were obtained from the IGA fitting procedure outlined above. Our method does not require the CAD geometry like [16] and produces C^1 continuous boundary curves or surfaces almost everywhere. Both the IGA fitting and elastic analogy based mesh generation methods were implemented in the open source PolyFEM code [30] and the solid mechanics problem was solved with PolyFEM's elasticity solvers.

3. Results & Discussion

We first demonstrate the IGA based high order mesh generation method with the 2D example shown in Fig. 1 and the results are shown in Fig. 2. Fig. 2(a) shows a regular or conventional quadratic mesh of the quarter plate-hole geometry, refined or subdivided directly from the linear mesh. The resultant high order nodes reside on the straight edges of the linear elements. Fig. 2(b) shows the evaluated IGA mesh elements based on the initial placement of the control points in Fig. 1. In Fig. 2(c) and (d), we show two IGA meshes based on two different fitting procedures. The first one only considers the position differences (i.e. $Q = D$) and the second considers both position and frame differences. In both cases, $w_i^d = 1.0$ are used for corners of interior elements and $w_i^d = 10.0$ are used for element corners on the boundaries. The first case uses $w_i^d = 0.5$ for the middle nodes. The second case uses $w_i^r = 1.0$ for the frame differences for all nodes but ignores the positional differences of the middle nodes($w_i^d = 0$). As observed, after fitting, the IGA center control points within the spline incompatible elements, which are introduced during the basis construction of their IGA element neighbors, are no longer coincident with the local center nodes of these elements. The first case causes large movement of these control points that increases the curvature of the element boarders. The second case causes relatively small movement of these control points, as it can be seen from the three spline incompatible elements.

In Fig. 2(e), we compare the four meshes by displaying their element boarders (wire frames) together. As it can be seen, the three IGA cases generate smooth curved elements. The first IGA mesh (green) without fitting does not fit the initial linear mesh well, with the differences clearly seen on the circular portion of the boundary. Both fitted IGA meshes visually interpolate the boundary nodes of the initial linear mesh. And the first (position only) fitting case (purple) also fits the interior element corners well but produces more interior element boarders with higher curvature. The second fitting case (red) produces a mesh that is closer to the initial (green) IGA mesh. The points shown in this figure are the quadratic high order nodes computed by evaluating the IGA mapping (Eq. 2) from the parametric sampling points (Eq. 1) to the physical space. To guarantee interpolation of the initial linear boundaries, we can easily move the high order corner nodes

(on the boundaries) back to its initial (linear) positions as a post-processing step if needed. In general, this will not decrease the quality of the final high order mesh since these node movements are relatively small.

In Fig.2(f), we show the comparison between the quadratic meshes generated from the first IGA fitting case (purple) and the elastic analogy approach (blue). The elastic analogy case uses the boundary condition obtained from the IGA fitting procedure and as a result the boundary nodes are coincident. For interior deformation, a linear elastic material model with Young's modulus $E = 1.5$ and Poisson's ratio $\mu = 0.49$ are used. As it can be seen from the figure, the movement of the interior nodes is not apparent, as the displacement decreases quickly away from the boundary. Overall, the IGA generated high order meshes are much smoother than the elastic analogy based mesh.

In Fig. 3, we demonstrate that the same IGA (position difference based) fitting and sampling procedure can be used to generate high order meshes for a 3D plate-hole example. In the figure, both quadratic and cubic meshes are shown. Each quadratic element was obtained from $3 \times 3 \times 3$ sampling points, whereas each cubic element was obtained from $4 \times 4 \times 4$ sampling points. Meshes of higher order can be obtained similarly.

In Fig. 4, we show the IGA fitted quadratic mesh of a 2D turbine blade enclosed in a flow domain. The initial mesh consists of only quadrilateral elements and has only a few spline incompatible elements (red) near the leading edge and the trailing edge (indiscernible in the figure). The initial mesh has 2,520 elements and 2,657 nodes and the initial IGA mesh has exactly the same number of elements with 2,830 control points (nodes). With only slightly more nodes, the IGA mesh can represent a much smoother mesh. In Fig. 4(b), the generated quadratic mesh based on element sampling is shown. The zoom-in views of four different meshes around the leading edge are shown in Fig. 5 for comparison. It includes the regular quadratic mesh, the IGA generated quadratic and cubic meshes, and the elastic analogy generated mesh. Clearly, the smoothness of the elastic analogy mesh deteriorates away from the boundary, while the IGA meshes maintain the smoothness much better.

In Fig. 6, we show a 3D linear mesh of the NACA 0012 airfoil, which was simply a one layer extrusion of a 2D mesh. The mesh was generated from an automatic meshing algorithm, which produced many spline incompatible elements (colored in red). In Fig. 7, the IGA fitted quadratic mesh of this NACA airfoil is shown. As it can be seen, the fitted mesh smooths both the interior (airfoil) and the outside boundaries. To demonstrate the usage of this generated high order mesh, we simulated the flow around the airfoil with the FLEXI simulation framework [45] that employs the high order discontinuous Galerkin method [46]. The simulations were set up non-dimensionally with Mach number $Ma = 0.4$ and Reynolds number $Re = 5000$. The inflow velocity has a magnitude of 1 and yields an angle of attack of $\alpha = 8°$. The airfoil wall has zero velocity and is assumed to be adiabatic. A sponge zone was used to remove artificial reflections from the outflow boundary so that a clean acoustic field can be retained. The initial condition is selected for density $\rho = 1.0$, velocity $\{u, v, v\} = \{0.990268069, 0.139173101, 0.0\}$ (unit velocity with an 8° angle), and

pressure $p = 4.4642857$. The end simulation time is 10 seconds at which the flow field stabilizes. The IGA fitted quadratic mesh was first exported to a file in the GMsh format[17] and then loaded into FLEXI. Flow simulations with discretization order from 1 (linear) to 4 were performed. For the linear discretization, the curved (middle) nodes were ignored, and the domain has straight edges. For higher order discretization, the curved quadratic mesh was used as input and further discretized accordingly. The density solutions for these simulations are shown in Fig. 8. As it can be seen, the linear solution is very coarse and does not capture the density pattern well. For the high order cases, the density solutions have better resolution and smoothness as the order increases.

Lastly, we present two more 3D examples in Fig. 9 and 10. Fig. 9 shows high order meshes of a 3D turbine blade and compares the regular quadratic mesh, evaluated IGA mesh, and the IGA fitted quadratic and cubic meshes. Fig. 10 shows high order meshes of a 3D mechanical part and also compares the regular quadratic mesh, evaluated IGA mesh, and the IGA fitted quadratic and cubic meshes. From these figures, we can still see the boundary smoothness of these IGA high order meshes are better than the regular high order meshes.

We also created high order meshes of these two 3D examples with the elastic analogy approach. However, the results are not displayed here as the differences from the IGA generated meshes are hard to discern due to high mesh density. In Table 1, we compare CPU time required to generate these meshes. The CPU time for IGA mesh generation includes two portions, one for fitting that involves solving a linear system of equations for least square minimization of Eq. 8 and the other for the evaluation of the physical positions of the parametric sampling points. The CPU time for elastic analogy does not include the time for computing the displacement of boundary nodes from the IGA fitting procedure. For the 3D blade case, IGA mesh generation took around 1.43 seconds while elastic analogy took around 47.77 seconds, more than 33 times of the IGA time. For the 3D mechanical part case, IGA mesh generation took around 21.42 seconds while elastic analogy took around 336.33 seconds, over 15 times of the IGA time. Evidently, the IGA based approach was faster than the elastic analogy method by a large margin.

	Elements	IGA		Elastic Analogy
		Fitting	Evaluation	
3D Blade	7,448	1.16	0.27	47.77
3D Mechanical	50,752	19.40	2.02	336.33

Table 1: Computational (CPU) time (in seconds) for quadratic mesh generation. The 3D blade case is shown in Fig. 9 and the 3D mechanical case is shown in Fig. 10. The HYPRE linear solver was used for both fitting and elastic analogy with the BoomerAMG preconditioner and preconditioned conjugate gradient (PCG) iterative solver [47] and the convergence tolerance was set to $1e-6$. The computation was performed on a desktop computer with an Intel Xeon W-2145 3.7GHz processor.

The examples presented above show several highlights of this new posterior, curved high order mesh

generation method. It requires only an initial linear mesh without the need of the CAD geometry and guarantees validity and adequate smoothness of the generated mesh. Our method notably outperforms the elastic analogy method in terms of both computational cost and mesh smoothness. On the flip side, there are also several limitations of the current method. First, the geometry approximation accuracy of the generated high order meshes is quadratic (since we use quadratic splines). The curved higher order (cubic or fourth order) meshes generated from the quadratic splines present here are used to demonstrate the capability of generating meshes of arbitrary orders that cater to the requirement of some academia or commercial high order analysis code. For example, in the flow around the NACA 0012 airfoil case, we show that better solution accuracy can be obtained with higher order meshes and discretization schemes. To further improve the geometric accuracy, higher order spline bases are desired. As stated in [30], the poly-spline based method utilized here focuses on construction of quadratic splines, but it can be extended to cubic polynomials if needed. But further degree elevation or p-refinement of the constructed spline bases are not as straightforward for unstructured meshes. Nonetheless, as our focus here is high order mesh generation of arbitrary order from the initial linear meshes, we have presented the possibility to use meshes of increasing order to perform p-refinement analysis. Second, the accuracy of the high order mesh also dependents on the linear mesh. If the linear mesh is too coarse, the current method may not recover the initial curved geometry well despite smooth boundary approximation. But if we assume the linear mesh is dense or an accurate representation of the initial geometry, our spline approximation can recover the curved initial geometry in good fidelity. Third, by the nature of the least square fitting, especially with the introduction of middle nodes and frame rotation in the objective function, we cannot guarantee the fitted smooth boundary interpolates the boundary nodes in the input linear mesh. In our approach, by assigning relatively high weights for these boundary nodes, we can obtain very good fitting for the curved boundary with smooth placement of middle nodes. As a result, we can often move the high order nodes back to their initial positions to ensure interpolation, as long as this movement will not cause degenerated or highly distorted elements. For the alternative method with additional element degeneration energy, we can control the fitting accuracy better with tolerance settings for the optimization, and, if needed, we can even enforce interpolation during each iteration while ensuring element validity.

The poly-spline method [30] utilized in this work for basis construction yields C^1 continuity for regular grid-like regions, where standard IGA can be applied, and C^0 continuity across interfaces with spline-incompatible elements, where standard Q_2 bases are constrained to ensure C^0 continuity. In the work by Wei et al. [48, 49], an alternative approach was proposed to build spline bases on unstructured hex-meshes and the continuity of constructed bases was studied around extraordinary nodes. Their method addresses the same problem as the poly-spline method: how to construct at least C^0 bases around singular or extraordinary points/edges/faces. They suggest adding additional control points around the singularities, thus the bases are splines everywhere. However, at the interfaces around the singularities, they propose to constrain nodes

and thus reduce smoothness. Since cubic spline bases are used, we believe their method may achieve C^1 continuity in some places around the singularities. For curved high order mesh generation applications, our approach is not tied to a specific basis construction [30] and could be extended to alternative constructions [48, 49]. We opted for using the poly-spline method [30] since the code is publicly available as an open-source project and provides an integrated meshing and IGA analysis framework.

Analysis-suitability of basis functions is very important to guarantee a stable and convergent IGA solver [50]. As demonstrated in [30], the poly-spline FEM is an integrated meshing, spline-basis construction, and FEM pipeline. The standard spline bases for each element of an unstructured mesh satisfies the pointwise nonnegativity, partition of unity, and compact support requirements by construction and thus analysis-suitable [50]. With these properties and the use of standard spline bases in combination with regular Q_2 elements, the linear independence (i.e. the geometric blending functions constitute a basis) and affine invariance properties are satisfied as a result. In the work by Schneider et al. [30, 1], the analysis-suitability of this method was validated by automatically solving Poisson's equation and other partial differential equations on a large collection of models. For the current application of high order mesh generation, our approach produces valid meshes with regular high order elements, which are analysis-suitability and well suited for both structure and flow analysis. Global refinement of the poly-splines and high order mesh generation is relatively easy. For local refinement of our mesh generation process, it is straightforward to generate elements of different orders within a mesh since each element can be sampled independently while keeping a continuous (but not necessarily conformal) interface between elements. For high order analysis code that does not require conformal element interfaces, this is not an issue. Otherwise, the neighbors of these locally refined elements may need to be further decomposed to generate conformal element interfaces.

4. Conclusion

We have presented a new posteriori method for generating curved high order meshes from IGA fitting and sampling. The present method is based on the recently developed poly-spline FEM that can handle quadrilateral or hexahedral dominant meshes. The salient features of the current method include: 1) it requires no CAD geometry from which the initial linear or low order mesh is generated; 2) it can handle both structured and unstructured quadrilateral or hexahedral meshes; 3) it provides superior performance over the elastic analogy based methods in terms of overall mesh smoothness and computational cost. Our method can also be extended for global or local refinement due to its local smoothness and continuity across elements. Other future work could include elevation of the IGA shape functions to higher orders, extension of the IGA based method to generate unstructured high order tetrahedral meshes [51] or hybrid meshes with mixed element types [52].

5. Acknowledgement

We would like to thank Dr. H.Q. Yang for providing us the 2D and 3D turbine blade meshes. This work was partially supported by a funded project through US DOD Contract W911NF-20-P-0008. It was also partially supported by the NSF CAREER award 1652515, the NSF grants IIS-1320635, OAC-1835712, OIA-1937043, CHS-1908767, CHS-1901091, NSERC DGECR-2021-00461, a gift from Adobe Research, a gift from nTopology, and a gift from Advanced Micro Devices, Inc. In addition, this work is also supported by the NYU IT High Performance Computing resources, services, and staff expertise.

References

[1] T. Schneider, Y. Hu, X. Gao, J. Dumas, D. Zorin, D. Panozzo, A large scale comparison of tetrahedral and hexahedral elements for finite element analysis, CoRR abs/1903.09332 (2019). arXiv:1903.09332.

[2] P. E. Vos, S. J. Sherwin, R. M. Kirby, From h to p efficiently: Implementing finite and spectral/hp element methods to achieve optimal performance for low-and high-order discretisations, Journal of Computational Physics 229 (13) (2010) 5161–5181.

[3] X. Zhou, J. Lu, Nurbs-based galerkin method and application to skeletal muscle modeling, in: Proceedings of the 2005 ACM symposium on Solid and physical modeling, 2005, pp. 71–78.

[4] J. A. Cottrell, T. J. Hughes, Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, 2009.

[5] H. Huynh, Z. J. Wang, P. E. Vincent, High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids, Computers & fluids 98 (2014) 209–220.

[6] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, et al., High-order cfd methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (8) (2013) 811–845.

[7] A. Jameson, Advances in bringing high-order methods to practical applications in computational fluid dynamics, in: 20th AIAA Computational Fluid Dynamics Conference, 2011, p. 3226.

[8] P. Ledger, K. Morgan, O. Hassan, Frequency and time domain electromagnetic scattering simulations employing higher order edge elements, Computer methods in applied mechanics and engineering 194 (2-5) (2005) 105–125.

[9] B. Cockburn, G. E. Karniadakis, C.-W. Shu, Discontinuous Galerkin methods: theory, computation and applications, Vol. 11, Springer Science & Business Media, 2012.

[10] H. Yang, Z. Chen, A. Przekwas, J. Dudley, A high-order cfd method using successive differentiation, Journal of Computational Physics 281 (2015) 690–707.

[11] P. E. Vincent, A. Jameson, Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists, Mathematical Modelling of Natural Phenomena 6 (3) (2011) 97–140.

[12] S. Dey, R. M. O'bara, M. S. Shephard, Curvilinear mesh generation in 3d., in: IMR, Citeseer, 1999, pp. 407–417.

[13] S. Dey, R. M. O'Bara, M. S. Shephard, Towards curvilinear meshing in 3d: the case of quadratic simplices, Computer-Aided Design 33 (3) (2001) 199–209.

[14] F. Hindenlang, T. Bolemann, C.-D. Munz, Mesh curving techniques for high order discontinuous galerkin simulations, in: IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach, Springer, 2015, pp. 133–152.

[15] J. Ims, Z. Duan, Z. J. Wang, meshcurve: an automated low-order to high-order mesh generator, in: 22nd AIAA computational fluid dynamics conference, 2015, p. 2293.

[16] R. Poya, R. Sevilla, A. J. Gil, A unified approach for a posteriori high-order curved mesh generation using solid mechanics, Computational Mechanics 58 (3) (2016) 457–490.

[17] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities, International journal for numerical methods in engineering 79 (11) (2009) 1309–1331.

[18] T. Toulorge, C. Geuzaine, J.-F. Remacle, J. Lambrechts, Robust untangling of curvilinear meshes, Journal of Computational Physics 254 (2013) 8–26.

[19] H. Q. Yang, X. Zhou, R. E. Harris, S. Yang, An open source, geometry kernel based high-order element mesh generation tool, in: AIAA Scitech 2019 Forum, 2019, p. 1719.

[20] X. Luo, M. S. Shephard, J.-F. Remacle, The influence of geometric approximation on the accuracy of high order methods, Rensselaer SCOREC report 1 (2001).

[21] S. Dey, M. S. Shephard, J. E. Flaherty, Geometry representation issues associated with p-version finite element computations, Computer methods in applied mechanics and engineering 150 (1-4) (1997) 39–55.

[22] M. Fortunato, P.-O. Persson, High-order unstructured curved mesh generation using the winslow equations, Journal of Computational Physics 307 (2016) 1–14.

[23] X. Jiao, D. Wang, Reconstructing high-order surfaces for meshing, Engineering with Computers 28 (4) (2012) 361–373.

[24] X. Jiao, N. R. Bayyana, Identification of c1 and c2 discontinuities for surface meshes in cad, Computer-Aided Design 40 (2) (2008) 160–175.

[25] S. L. Karman, J. T. Erwin, R. S. Glasby, D. Stefanski, High-order mesh curving using wcn mesh optimization, in: 46th AIAA Fluid Dynamics Conference, 2016, p. 3178.

[26] P.-O. Persson, J. Peraire, Curved mesh generation and mesh refinement using lagrangian solid mechanics, in: 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, 2009, p. 949.

[27] N. Dyn, D. Levin, J. A. Gregory, A 4-point interpolatory subdivision scheme for curve design, Computer aided geometric design 4 (4) (1987) 257–268.

[28] D. Zorin, P. Schröder, W. Sweldens, Interpolating subdivision for meshes with arbitrary topology, in: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996, pp. 189–192.

[29] T. DeRose, M. Kass, T. Truong, Subdivision surfaces in character animation, in: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, 1998, pp. 85–94.

[30] T. Schneider, J. Dumas, X. Gao, M. Botsch, D. Panozzo, D. Zorin, Poly-spline finite-element method, ACM Transactions on Graphics (TOG) 38 (3) (2019) 1–16.

[31] T. J. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement, Computer methods in applied mechanics and engineering 194 (39-41) (2005) 4135–4195.

[32] T. Martin, E. Cohen, Volumetric parameterization of complex objects by respecting multiple materials, Computers & Graphics 34 (3) (2010) 187–197.

[33] B. Li, X. Li, K. Wang, H. Qin, Surface mesh to volumetric spline conversion with generalized polycubes, IEEE Transactions on Visualization and Computer Graphics 19 (9) (2012) 1539–1551.

[34] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, T. W. Sederberg, Isogeometric analysis using t-splines, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 229–263.

[35] D. Schillinger, L. Dede, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, T. J. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of nurbs, immersed boundary methods, and t-spline cad surfaces, Computer Methods in Applied Mechanics and Engineering 249 (2012) 116–150.

[36] W. Wang, Y. Zhang, L. Liu, T. J. Hughes, Trivariate solid t-spline construction from boundary triangulations with arbitrary genus topology, Computer-Aided Design 45 (2) (2013) 351–360.

[37] X. Li, T. W. Sederberg, S-splines: A simple surface solution for iga and cad, Computer Methods in Applied Mechanics

and Engineering 350 (2019) 664–678.

[38] W. Wang, Y. Zhang, M. A. Scott, T. J. Hughes, Converting an unstructured quadrilateral mesh to a standard t-spline surface, Computational Mechanics 48 (4) (2011) 477–498.

[39] W. Wang, Y. Zhang, G. Xu, T. J. Hughes, Converting an unstructured quadrilateral/hexahedral mesh to a rational t-spline, Computational Mechanics 50 (1) (2012) 65–84.

[40] X. Gao, W. Jakob, M. Tarini, D. Panozzo, Robust hex-dominant mesh generation using field-guided polyhedral agglomeration, ACM Transactions on Graphics (TOG) 36 (4) (2017) 1–13.

[41] X. Gao, H. Shen, D. Panozzo, Feature preserving octree-based hexahedral meshing, in: Computer Graphics Forum, Vol. 38, Wiley Online Library, 2019, pp. 135–149.

[42] J. C. Simo, D. D. Fox, On a stress resultant geometrically exact shell model. i: Formulation and optimal parametrization, Computer Methods in Applied Mechanics and Engineering 72 (3) (1989) 267–304.

[43] X. Zhou, J. Lu, Inverse formulation for geometrically exact stress resultant shells, International journal for numerical methods in engineering 74 (8) (2008) 1278–1302.

[44] M. Rabinovich, R. Poranne, D. Panozzo, O. Sorkine-Hornung, Scalable locally injective mappings, ACM Trans. Graph. 36 (2) (Apr. 2017).

[45] N. Krais, A. Beck, T. Bolemann, H. Frank, D. Flad, G. Gassner, F. Hindenlang, M. Hoffmann, T. Kuhn, M. Sonntag, et al., Flexi: A high order discontinuous galerkin framework for hyperbolic–parabolic conservation laws, Computers & Mathematics with Applications (2020).

[46] F. Hindenlang, G. J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, C.-D. Munz, Explicit discontinuous galerkin methods for unsteady problems, Computers & Fluids 61 (2012) 86–93.

[47] R. D. Falgout, U. M. Yang, hypre: A library of high performance preconditioners, in: International Conference on Computational Science, Springer, 2002, pp. 632–641.

[48] X. Wei, Y. J. Zhang, T. J. Hughes, Truncated hierarchical tricubic c0 spline construction on unstructured hexahedral meshes for isogeometric analysis applications, Computers & Mathematics with Applications 74 (9) (2017) 2203–2220.

[49] X. Wei, Y. J. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J. A. Evans, T. J. Hughes, Blended b-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 341 (2018) 609–639.

[50] H. Casquero, X. Wei, D. Toshniwal, A. Li, T. J. Hughes, J. Kiendl, Y. J. Zhang, Seamless integration of design and kirchhoff–love shell analysis using analysis-suitable unstructured t-splines, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112765.

[51] S. Xia, X. Qian, Isogeometric analysis with bézier tetrahedra, Computer Methods in Applied Mechanics and Engineering 316 (2017) 782–816.

[52] L. Engvall, J. A. Evans, Isogeometric unstructured tetrahedral and mixed-element bernstein–bézier discretizations, Computer Methods in Applied Mechanics and Engineering 319 (2017) 83–123.

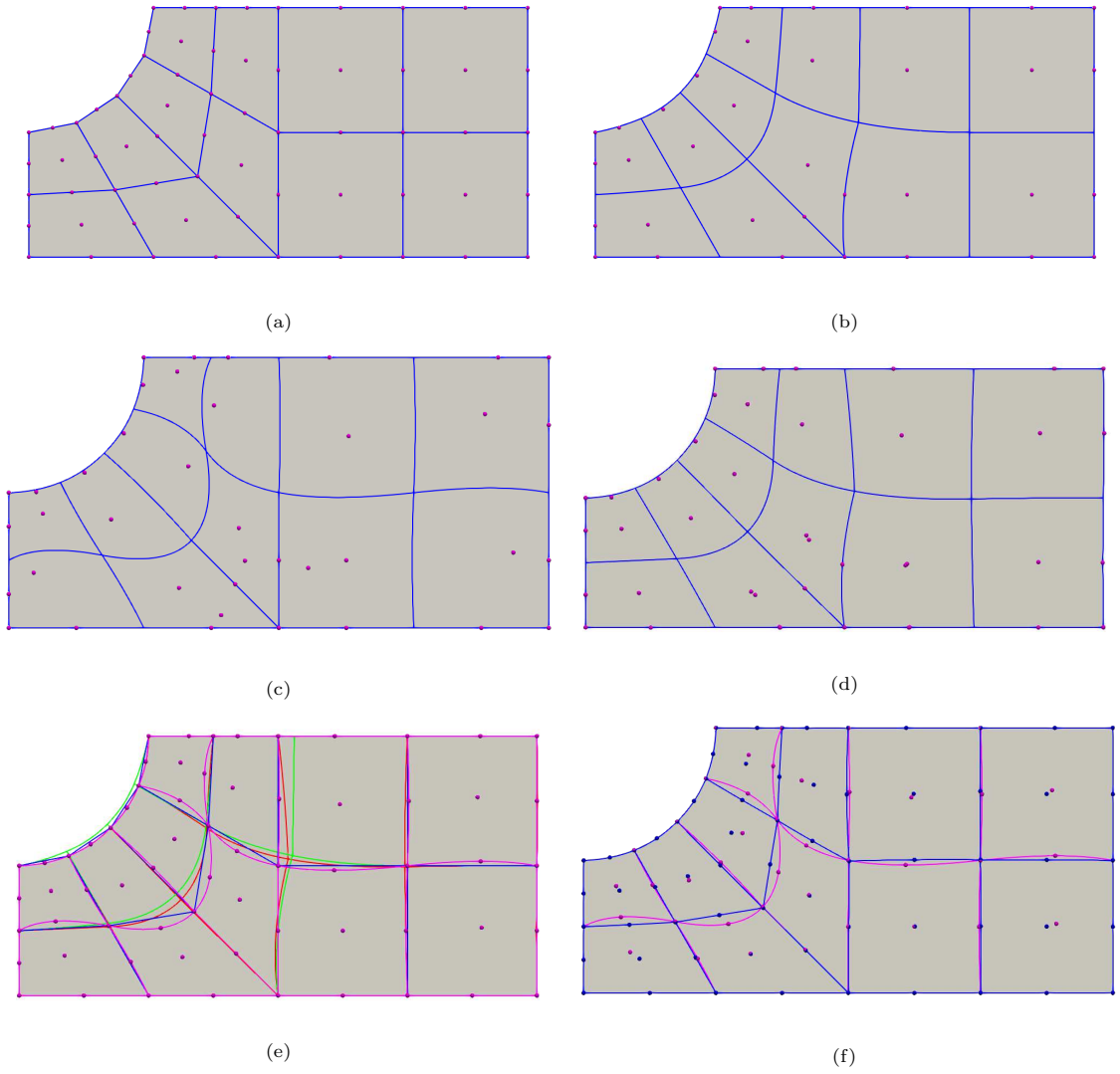
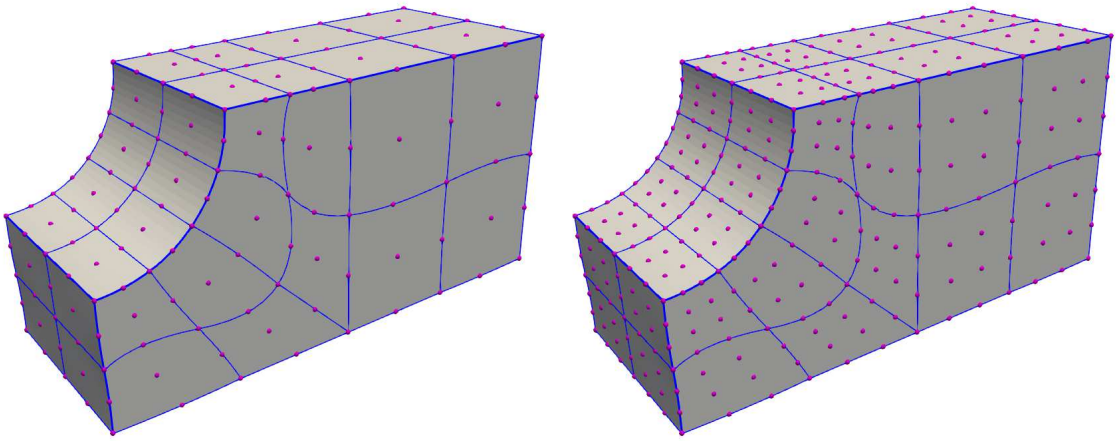
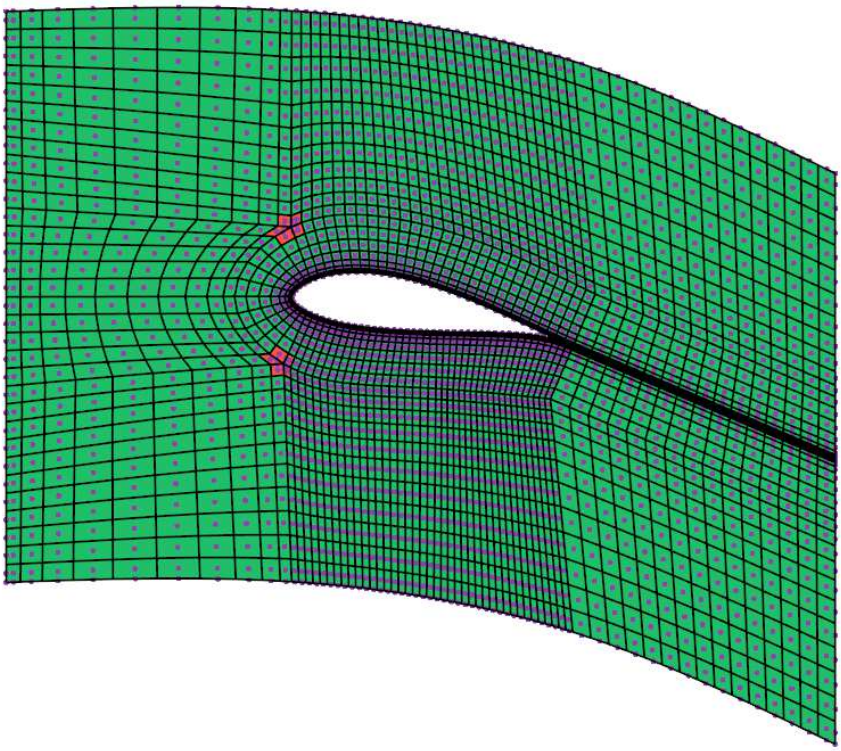


Figure 2: (a) The regular quadratic mesh of the quarter plate-hole geometry by simple refinement or subdivision of the linear mesh; (b) The evaluated IGA mesh elements based on the initial placement of the control points shown in Fig. 1b; (c) The position fitted IGA mesh from the new positions of control points; (d) The position and frame fitted IGA mesh from the new positions of control points; (e) Comparison of the regular and IGA meshes. The blue wire is the regular mesh, the green wire is the initial IGA mesh, the purple wire is the position fitted IGA mesh, and the red wire is the position and frame fitted IGA mesh. The displayed points are the generated quadratic mesh points from the first fitted IGA mesh. (f) Comparison of quadratic meshes generated with the first IGA fitting approach (purple) and the elastic analogy approach (blue).

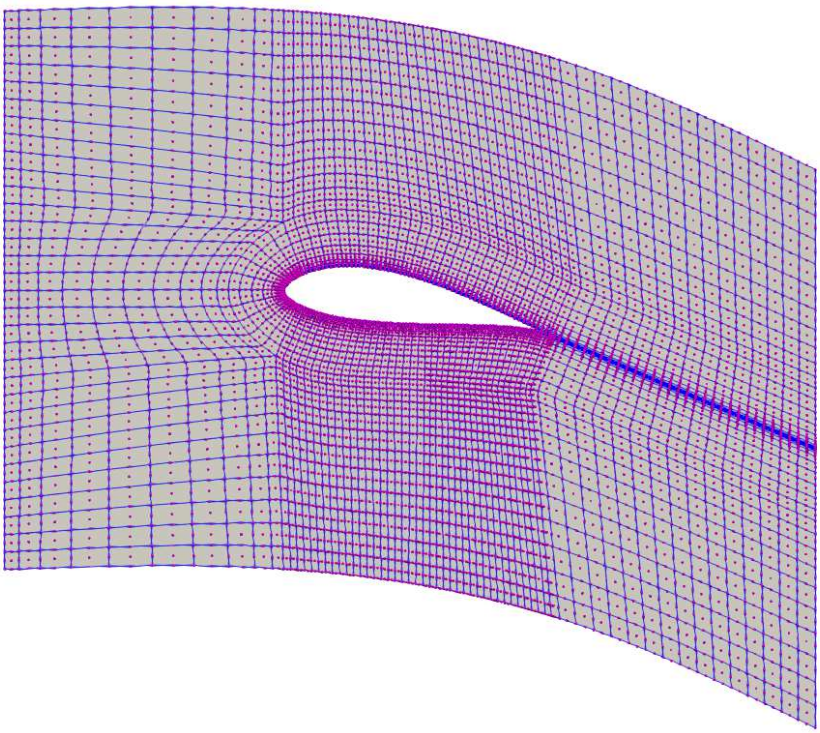


(a) (b)

Figure 3: 3D high order meshes of the quarter plate-hole geometry generated from IGA fitting and sampling.(a) Quadratic mesh; (b) Cubic mesh.



(a)



(b)

Figure 4: IGA quadratic mesh of a 2D turbine blade.(a) Initial placement of the IGA control points; (b)Fitted IGA quadratic mesh.

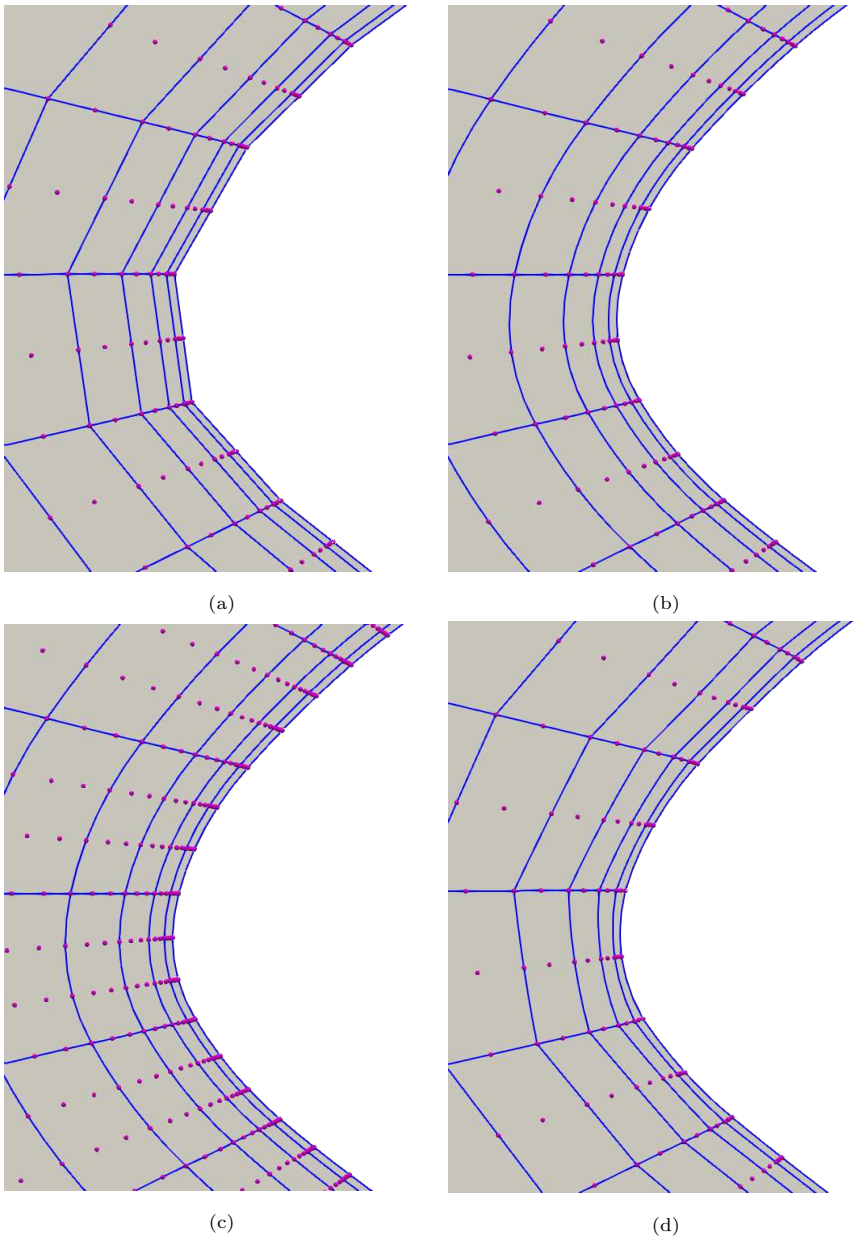
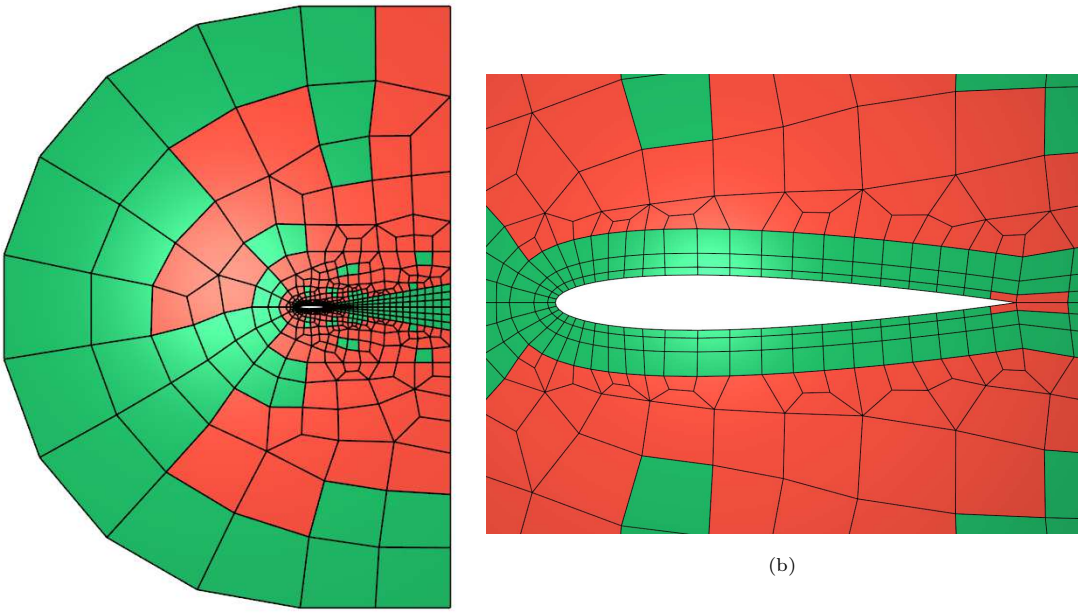


Figure 5: Zoom-in views of the regular and IGA meshes of the turbine blade. (a) Regular quadratic mesh; (b)IGA quadratic mesh; (c) IGA cubic mesh; (d) Quadratic mesh generated from the elastic analogy approach.



(a) (b)

Figure 6: Linear mesh of the NACA 0012 airfoil.(a) Full mesh with spline incompatible elements colored in red; (b) Zoom-in view of the mesh around the airfoil.

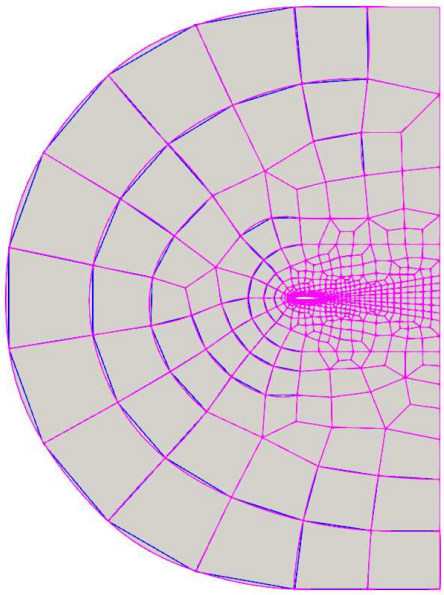


Figure 7: IGA fitted quadratic mesh of the NACA airfoil. The purple wires are element contours of the fitted mesh and the blue wires are the element contours of the initial linear mesh.

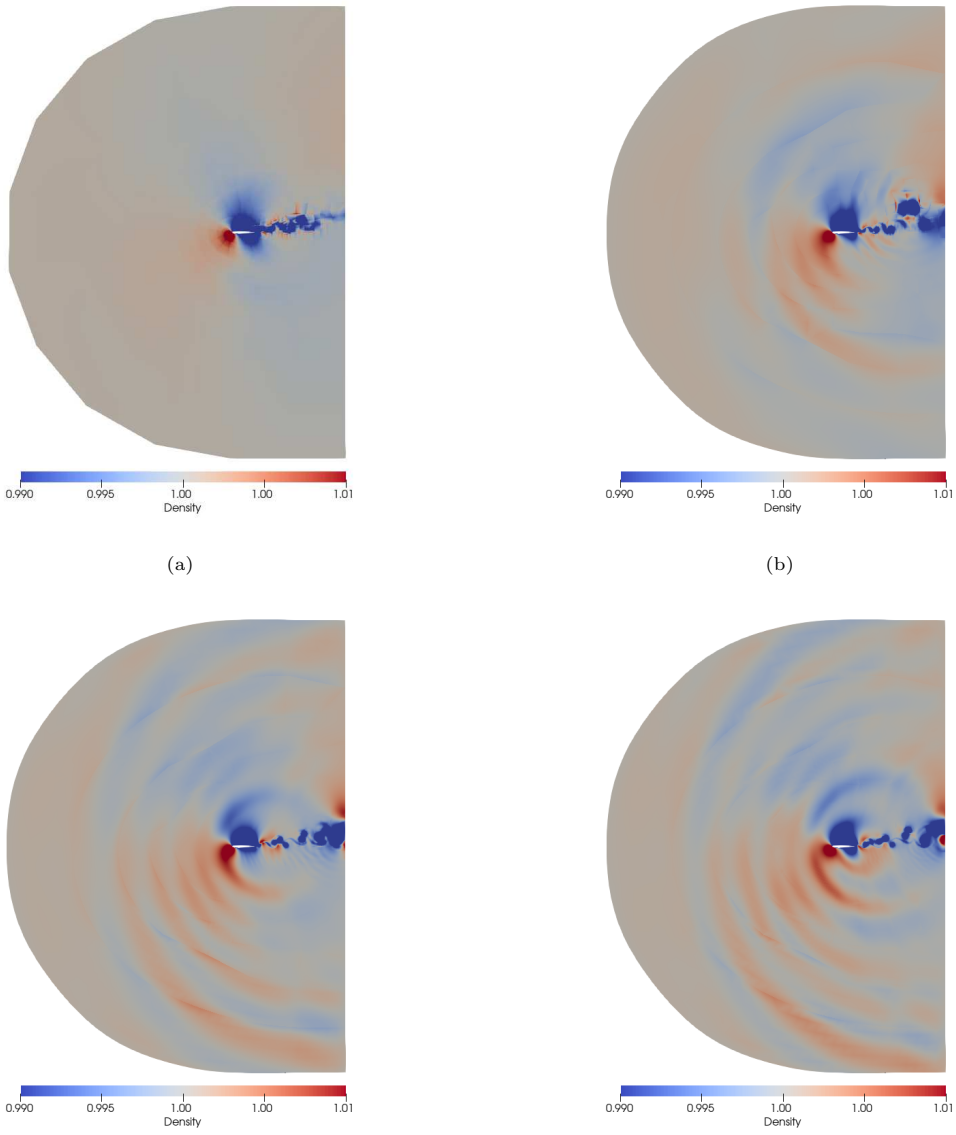
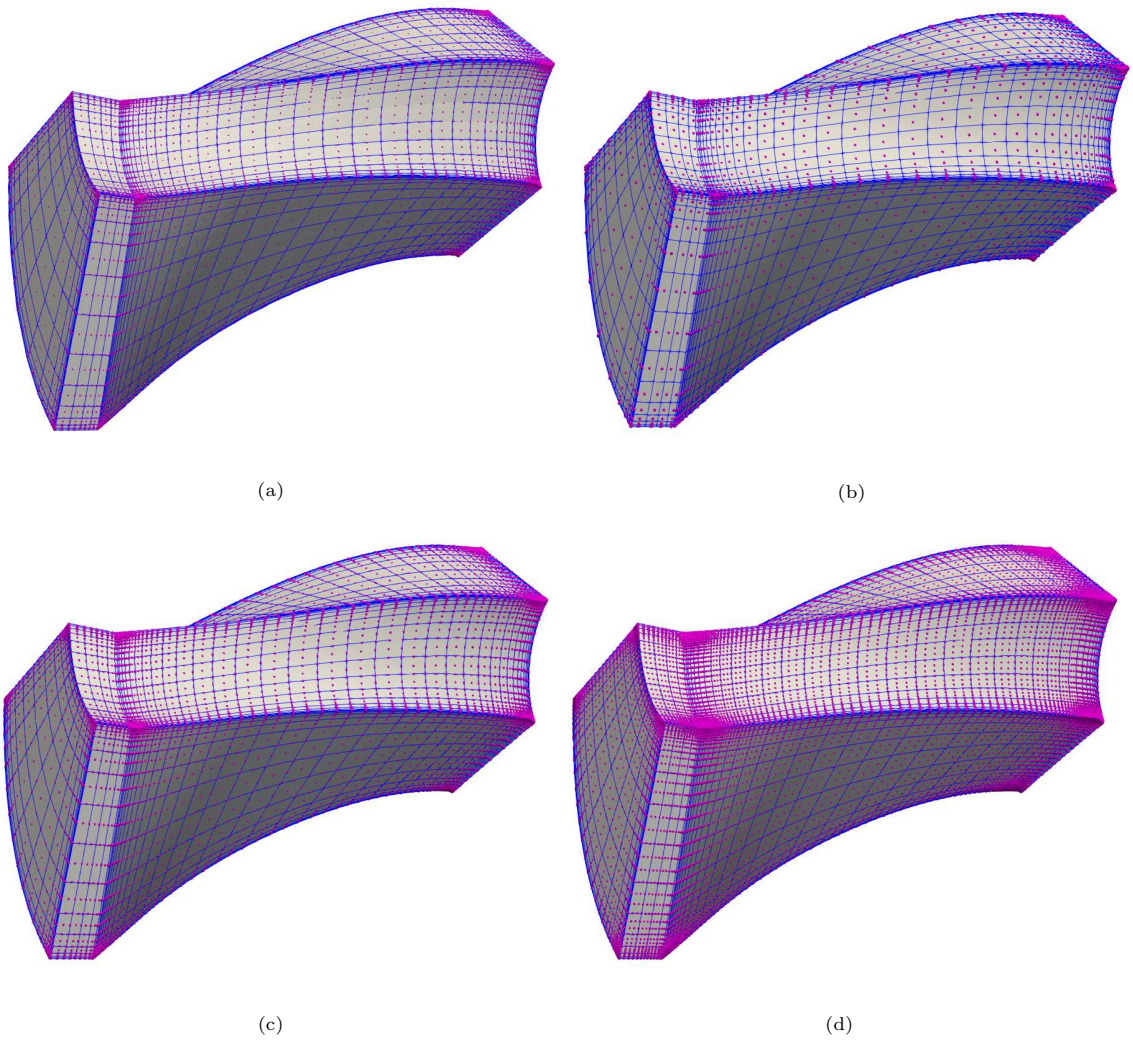


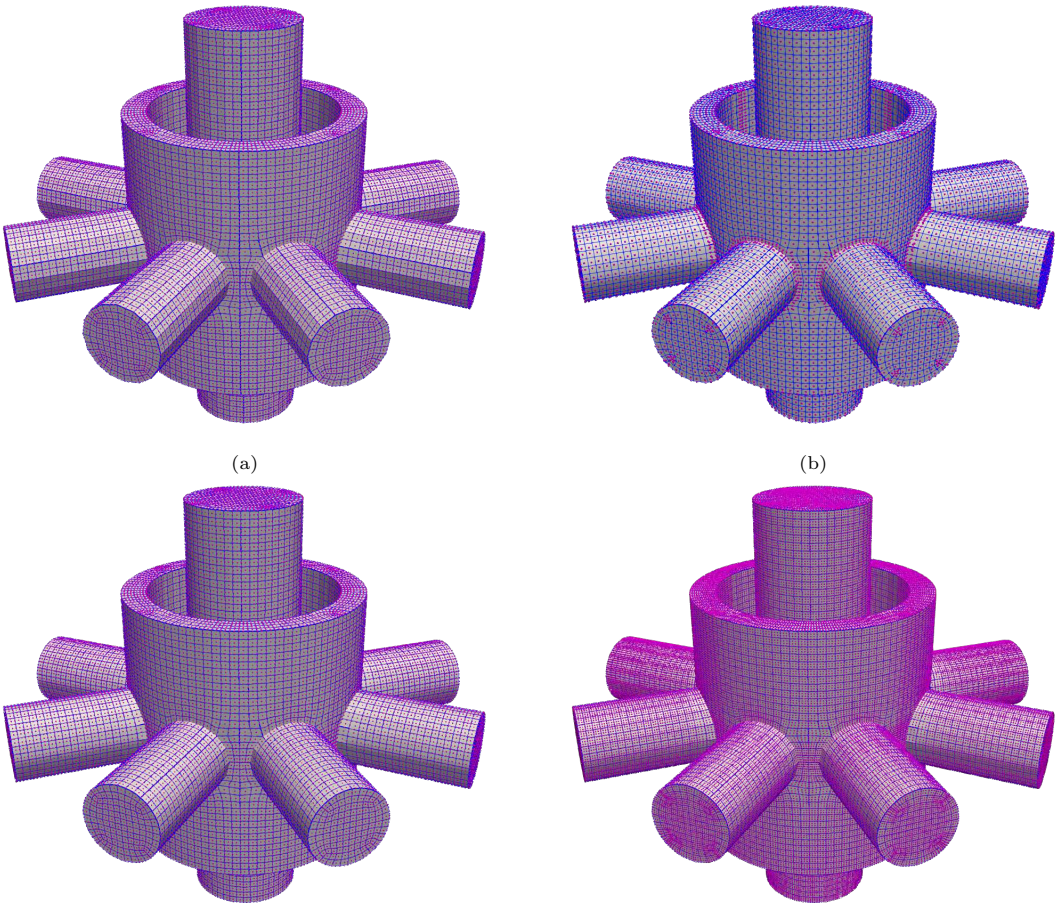
Figure 8: Density solutions of the flow around the NACA 0012 airfoil. (a) Linear, (b) quadratic, (c) cubic, and (d) fourth order solutions.



(a) (b)

(c) (d)

Figure 9: High order meshes of a 3d turbine blade. Its linear mesh has 7,448 elements and 8,775 nodes. (a) Regular quadratic mesh; (b) Evaluated IGA mesh with control points; (c) Quadratic mesh generated with IGA sampling; (d) Cubic mesh generated with IGA sampling. The evaluated IGA mesh has 10,240 nodes and the sampled quadratic and cubic meshes has 64,757 nodes and 212,635 nodes, respectively.



(a) (b)

(c) (d)

Figure 10: High order meshes of a 3D multi-axis mechanical part. Its linear mesh has 50,752 elements and 59,185 nodes. (a) Regular quadratic mesh; (b) Evaluated IGA mesh with control points; (c) Quadratic mesh generated with IGA sampling; (d) Cubic mesh generated with IGA sampling. The evaluated IGA mesh has 108,646 nodes and the sampled quadratic and cubic meshes has 43,9201 nodes and 1,444,561 nodes, respectively.