Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona, Spain, July 9-13, 2018

ICCFD10-170

High-Order Mesh Generation using RBFs Based on Geometrical Similarity

Zi-Hao Zhao*, Wen-Ping SONG *, Ke SONG *
Corresponding author: wpsong@nwpu.edu.cn

* National Key Laboratory of Science and Technology on Aerodynamic Design and Research, Northwestern Polytechnical University, Xi'an, 710072, China

Abstract: This paper proposes a method to generate high-order curvilinear mesh using compact radial basis function (RBF) interpolation with optimized support radius. We start with an initial high-order linear mesh, and relocate the boundary nodes to their true positons on the true curved geometry. Then the rest mesh nodes are moved using RBF interpolation based on boundary nodes displacement. However, if one assigns identical support radius to all control points, the interpolation might result in poor-quality or even invalid mesh elements. In this article, we present an optimization process to determine a proper set of support radius. Additionally, a novel and fast curved mesh quality estimation based on geometrical similarity between linear and curvilinear mesh element is used to speed up the optimization process. A number of examples including highly stretched boundary layer meshes of complex geometries are tested, the results illustrate that our method is able to produce high quality curved meshes with low computational cost.

Keywords: Curved Mesh Generation, Mesh Quality Control, Geometrical Similarity, Radial Basis Function.

1 Introduction

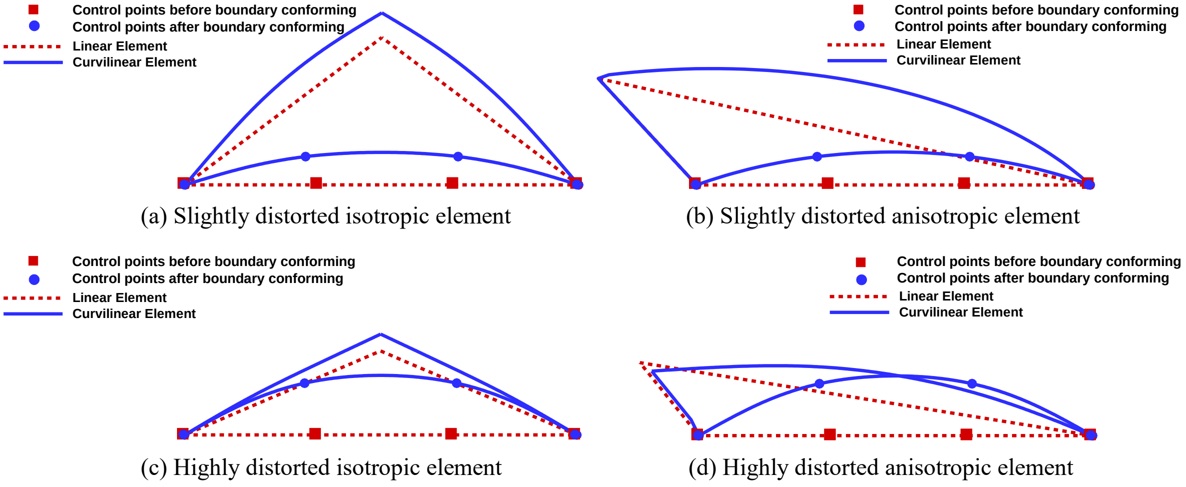
High-order CFD methods such as Discontinuous Galerkin (DG) methods [1], Spectral Difference (SD) methods [2, 3] and Flux Reconstruction/Correction Procedure via Reconstruction (FR/CPR) methods [4] have been developed for many years. Both academia and industry have paid considerable attention to these methods for their favorable numerical features and inherent parallelism. For high-order methods, linear mesh elements become not adequate, more than one paper [5, 6] showed clearly that straight-sided mesh can severely reduce the accuracy of these methods. Therefore, high-order space discretization is essential for high-order methods. In 2012, Wang et al. [7] pointed out that one of the required pacing items in high-order community was curvilinear mesh generation. However, there is still no commercial mesh generator is able to produce high-order unstructured mesh.

Research in this field has mainly focus on *posteriori* approach which means a linear-to-curvilinear transformation. The process is as follow: coarse straight-sided mesh is generated first, then boundary nodes are relocated according to the true curvature of boundary, and the final step is to regularize all interior nodes according to boundary displacements. The present paper also follows this idea. Challenge around *posteriori* approaches is to find a robust and efficient method through which the displacements of boundary mesh nodes can propagate to the interior nodes properly. Multiple research groups have developed different methods to accomplish this task. Their works can be broadly divided into two groups. The first group treats the mesh as a solid body and their theme is to find a best solid mechanics model which can lead to optimal mesh quality. Until now, the physical models proposed by different researchers include linear elasticity [8, 9], non-linear elasticity [10, 11], a thermo-elastic analogy [12] and Winslow equation [13]. The basic thought of second group is to take use of an optimization process, where the objective function is set to be a functional related to mesh quality. The functional include Jacobian [14], shape distortion metric [15, 16], and spring analogies for surface deformation [17]. Other than above two groups, very few studies tried *interpolation* technique to perform mesh untangling [18, 19].

Actually, most of above methods have already been applied to mesh deformation problem [20] in computational aeroelasticity and aerodynamic shape optimization communities for years. Actually, the regularization of curvilinear mesh can also be treated as a mesh deformation problem: the upgrade of high order curved boundary surface mesh from linear surface mesh can be viewed as the movement of mesh boundary and then the untangling process of mesh elements can be regarded as a regular mesh deformation procedure. The initial thought of present study comes from this idea, our goal is to use sophisticated algorithms in mesh deformation community for reference to accomplish high-quality curvilinear mesh generation.

Nowadays, one of the most popular algorithms in mesh deformation community is radial basis function (RBF) interpolation [21]. In 2007, Boer .et al [22] first introduced it into mesh deformation field. During the last decade, researchers developed many works [23-25] to make it more efficient and robust. Recent years, very few studies introduced it into curvilinear mesh generation. Sasrty [26] first adopted RBF interpolation with global support using thin-plate-spline to perform interior mesh nodes regularization in 2015. Zala [27] improved Sasrty's method by replacing the kernel and introducing a smoothing algorithm in 2018. They both adopted a global support as interpolation function. However, in mesh deformation community, people prefer compact support since it has been proven a good balance between the quality of resultant mesh and computational efficiency [23]. In 2016, Kashi [28] tried a RBF interpolation with compact support to produce curvilinear mesh. His results showed that RBF interpolation is much better than solid-mechanics based methods in both mesh quality and computing cost.

When we use RBF interpolation with compact support, a parameter "*support radius*" should be assigned first. Usually, people choose to adopt identical support radius for all control points. However, it might become unsuitable when encountering highly distorted [29] mesh elements (Figure 1), which could be a common situation in CFD simulation with high order methods.


■ Control points before boundary conforming
● Control points after boundary conforming
Linear Element
Curvilinear Element

(a) Slightly distorted isotropic element

(b) Slightly distorted anisotropic element

(c) Highly distorted isotropic element

(d) Highly distorted anisotropic element

Figure 1: Examples of curvilinear elements with different distortion. RBF interpolation has a good performance with slightly distorted element as shown in case (a) and (b). However, it might generate poor-quality or even invalid element when the distortion is large as shown in (c) and (d).

For these elements, the quality of regularized curvilinear elements are highly related to the support radius of control points. If one set support radius without careful consideration, the interpolation algorithm may produce invalid elements. To overcome the problem, we will introduce an optimization process to find a proper set of support radius for all control points. We also propose a novel mesh quality indicator based on geometrical similarity between linear mesh element and curved element as the objective function. It's computational more efficient than some previous optimization-based works' objective function. Moreover, this indicator is also capable to reflect the similarity between linear and curvilinear mesh.

This paper is organized as follow. In section 2, a brief introduction of RBF interpolation with compact support is described. The novel mesh quality indicator will be presented in section 3. Section 4 states the detailed optimization procedure of support radius. Then, we illustrate the performance of our method through three examples including meshes with highly stretched anisotropic boundary layer elements and coarse isotropic elements for complex geometry in section 5. Conclusions are placed in section 6.

2 RBF Interpolation with compact support

Radial basis functions are set of radially-symmetric functions. They are used as a basis for the displacement field in the interior mesh nodes regularization. The form of an RBF interpolation is

$$\boldsymbol{s}(\boldsymbol{x}) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \varphi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_i\right\|\right) \tag{1}$$

where $\boldsymbol{s}$ is the function vector to be evaluated at mesh node $\boldsymbol{x}$ and will define its displacements in different coordinates, the subscript i indicates the i-th control point, while $\boldsymbol{x}_i$ identifies its location. Here the control points are the mesh nodes located on curved boundaries of CAD models. The coefficient vector $\boldsymbol{\alpha}_i$ can be found by exact recovery of the original function at all the control points. φ is the kernel of interpolation, and can be divided into three different categories: global, local and compact [23] . Compact functions have been proven a good balance between quality of the mesh motion and the conditioning of the linear system for finding the set of α_i. The form of compact functions are expressed as follow

$$\varphi(\xi) = \begin{cases} \varphi(\xi) & 0 \le \xi \le 1 \\ 0 & \xi > 1 \end{cases} \tag{2}$$

where $\xi = \left(\left\|\boldsymbol{x} - \boldsymbol{x}_i\right\|\right) / r_i$ and r_i is the support radius of i-th control point.

There are multiple choices of compact functions, in our study, Wendland's C2 kernel is selected as basis function:

$$\varphi(\xi) = (1-\xi)^4 (4\xi + 1) \tag{3}$$

Then α_i can be solved using

$$\boldsymbol{S}_i = \sum_{j=1}^{N} \boldsymbol{\alpha}_i \phi\left(\left\|\boldsymbol{x}_j - \boldsymbol{x}_i\right\|\right) \tag{4}$$

where $\boldsymbol{S}_i$ is a displacement vector of i-th control point. This leads to a linear system of N equations in N unknowns at each coordinate direction.

RBF interpolation is able to propagate the displacements of boundary nodes to interior of the mesh. Moreover, the motions of interior mesh nodes will follow the same motion pattern of boundary nodes. This property will be used to construct the new mesh quality indicator in section 3.

As we mentioned in section 1, for mesh with highly distorted elements, using RBF interpolation with same support radius for all the control points might lead to poor-quality or even invalid elements. Hence, in the following sections, the detail of optimization procedure for determining a suitable set of support radius distribution will be proposed.

3 Quality indicator based on geometrical similarity

Before defining the quality indicators, there is an essential principle must be declared first. For previous literatures adopting optimization-based methods [14], the mesh quality is obtained by evaluating all the curvilinear elements. However, as we stated in section 2, since RBF interpolation has a nature property of propagating the displacements of boundary nodes to interior mesh nodes, an assumption to simplify the mesh quality estimation can be introduced: the quality of curvilinear mesh will be estimated only at the elements attached to the curved boundaries. Once these elements are untangled, according to the propagating property, we believe all the interior elements can be untangled. This assumption can reduce a considerable computing cost of optimization process especially for fine meshes.

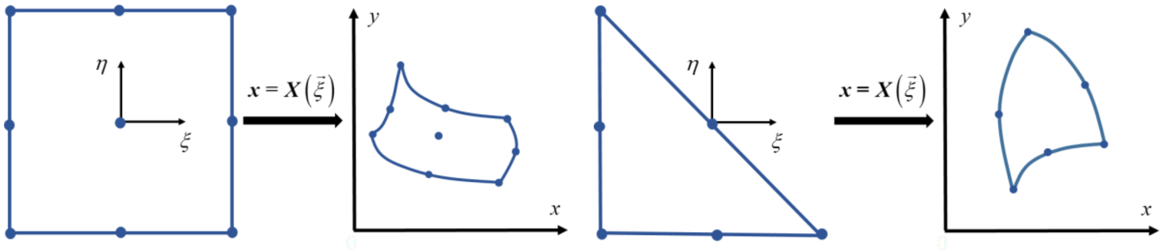
Then, in order to find an acceptable set of support radius, a proper method for quantifying the mesh quality is required. As the mesh quality is to be set as the objective function of optimization, it might be computed for a huge amount of elements and iterations. Hence, the computational efficiency of the evaluation algorithm is a crucial point. Previous works mainly adopted a Jacobian-related functional [15] as the mesh quality indicator. There is no doubt that they are one of the most accurate quality indicators, however, their computational cost could be relatively expensive.

In this section, the most popular curvilinear mesh quality indicator scaled Jacobian [30] will be introduced first. We discuss the corresponding evaluation strategy, and show that it’s costly as an objective function. Therefore, based on the mathematical property of RBF interpolation, we propose a novel mesh quality indicator, which is computational inexpensive, to be our objective function.

3.1 Scaled Jacobian

Since most high order methods involve quadrature rule, a transformation between the computational and the physical space is needed (Figure 2). The determinant of the Jacobian matrix of this mapping $\boldsymbol{x} = \boldsymbol{X}\left(\vec{\xi}\right)$ can be used to evaluate the quality of a curvilinear element. The scaled Jacobian I is defined as follow

$$I_i = \frac{\min_{\vec{\xi}\in\Omega_i} J\left(\vec{\xi}\right)}{\max_{\vec{\xi}\in\Omega_i} J\left(\vec{\xi}\right)} \tag{5}$$


η
ξ
$\boldsymbol{x} = \boldsymbol{X}\left(\vec{\xi}\right)$
y
x
η
ξ
$\boldsymbol{x} = \boldsymbol{X}\left(\vec{\xi}\right)$
y
x

(a) mapping of a quadrilateral element

(b) mapping of a triangular element

Figure 2: Notation for mapping between reference elements and curvilinear elements in 2D domain.

The subscript i indicates i-th mesh element. In practice, the Jacobian is evaluated at all quadrature nodes. I_i is always a real number less or equal to 1, which makes it a practical quality indicator for all kinds of curvilinear mesh. A negative or very small value of I_i indicates that the i-th element is invalid or close to degenerate.

However, the calculation of scaled Jacobian could be costly. For a nodal high order element determined by n nodes, we consider the basis $\{N_i\}_{i=1,\ldots,n}$ of nodal shape function with Lagrange interpolation. Then, the mapping can be expressed as:

$$\boldsymbol{x} = \boldsymbol{X}\left(\vec{\xi}\right) = \sum_{i=1}^{n} \boldsymbol{x}_i N_i\left(\vec{\xi}\right) \tag{6}$$

and the Jacobian of mapping (6) is:

$$J\left(\vec{\xi}\right) = \begin{vmatrix} \sum_{i=1}^{n} x_i \frac{\partial N_i}{\partial \xi}\left(\vec{\xi}\right) & \sum_{i=1}^{n} x_i \frac{\partial N_i}{\partial \eta}\left(\vec{\xi}\right) \\ \sum_{i=1}^{n} y_i \frac{\partial N_i}{\partial \xi}\left(\vec{\xi}\right) & \sum_{i=1}^{n} y_i \frac{\partial N_i}{\partial \eta}\left(\vec{\xi}\right) \end{vmatrix} \tag{7}$$

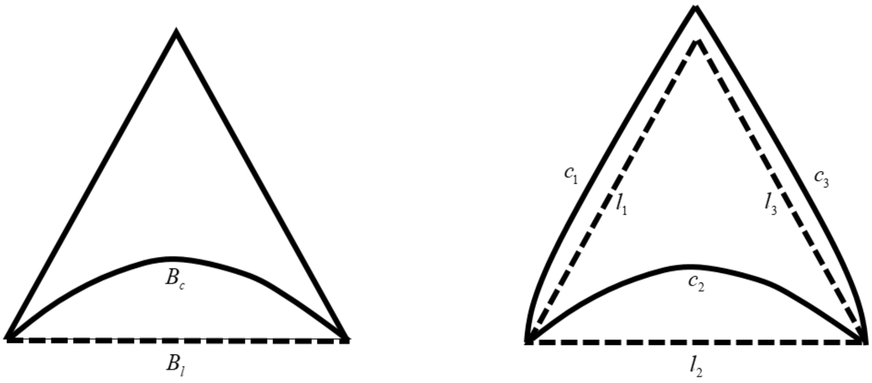
Clearly, the evaluation of scaled Jacobian involves multiple interpolation processes and as the CPU time costed given in Table 3.1 suggest, adopting scale Jacobian as an objective function will be less efficient than our new quality indicator which will be introduced in the coming section.

3.2 Geometrical similarity of a curvilinear element

To improve the efficiency, a fast estimation of mesh quality is needed to replace scaled Jacobian. Moreover, as we mentioned before, most methods adopt *posteriori* approaches to produce curvilinear mesh, which means a linear mesh is already generated in the first place. Here we assume the quality of linear mesh is good and all the elements are well-shaped and distributed reasonably. Hence, the geometrical patterns and distribution of curvilinear elements produced by *posteriori* approaches are best to be similar to these properties in linear mesh. For example, in a boundary layer mesh, the boundary mesh intervals and orthogonality near wall should not be highly changed after a *posteriori* approach.

Although scaled Jacobian is able to indicate the mesh validity clearly, it cannot reflect the similarity between linear and curvilinear mesh. Also, its calculation is costly. To indicate the similarity of meshes and improve the efficiency, a fast estimation of similarity is needed to replace scaled Jacobian as an objective function. Giving the propagating property of RBF interpolation, we found that if it works properly, the resultant curvilinear mesh element will share the same geometrical pattern with its corresponding linear element. Here we define it as the *geometrical similarity* between linear and curvilinear element. And a similarity ratio s of a certain element attached to curved boundary is defined as the ratio of the arc length between the segment attached to curved boundary and its corresponding linear segment (Figure 3(a)).

$$s{=}B_c\ /\ B_l \tag{8}$$



(a) Arc lengths of segment attached to boundary in linear and curvilinear element

(b) Arc lengths of linear and curvilinear segment

Figure 3: Illustration of geometrical similarity and mesh quality indicator Q .

Based on this parameter, we define a quality indicator Q of a certain curvilinear element attached to curved boundary as:

$$Q = A \cdot \left(\frac{\prod_{n=1}^{N} c_n / l_n}{s^2} - 1 \right) + 1 \tag{9}$$

c_n and l_n means the arc lengths of n-th curvilinear segment and its corresponding linear segment respectively (Figure 3(b)). Since the ratio between products of arc lengths are very close to one, an amplification' factor A is introduced to make the indictor more intuitive. We have used A=10^5 in present work. Q is always a real number large or equal to 1, and smaller Q indicates that curvilinear element is geometrically more similar to its corresponding linear element.

3.3 Simple examples

In order to illustrate the efficient of different estimations, we consider the simple cases of both quadrilateral and triangular elements around a unit circle.

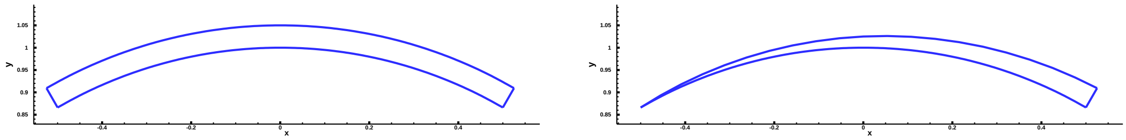


Figure 4: Quadrilateral (left) and triangular (right) first layer mesh element of a unit circle.

The algorithm described in section 3.1 and 3.2 is used to calculate the scaled Jacobian I and geometrical similarity Q from 2^{nd} order to 4^{th} order element. Table 3.1 summarizes the CPU time for all estimations and Table 3.2 summarizes the estimated values. In this work, all CPU time were counted with sequential computations on a desktop PC equipped with Intel Xeon E5-1620 v3 at 3.5GHz.

Table 3.1 CPU time ($10^{-6} s$) of two different quality indicators from 2^{nd} order to 4^{th} order element.

Quality Indicator	Quadrilateral element			Triangular element		
	Order 2	Order 3	Order 4	Order 2	Order 3	Order 4
I	71.00	130.00	185.00	37.50	92.10	102.00
Q	9.00	14.00	20.00	5.60	9.80	11.60

Table 3.2 Estimated values of two different quality indicators from 2^{nd} order to 4^{th} order element.

Quality Indicator	Quadrilateral element			Triangular element		
	Order 2	Order 3	Order 4	Order 2	Order 3	Order 4
I	0.84	0.95	0.95	0.79	0.92	0.92
Q	1.00	1.00	1.00	2.88	2.94	2.92

Giving the much better computational efficiency and acceptable accuracy, it’s clearly that our mesh quality indicator Q is more adequate as an objective function for support radius optimization which will be discussed in section 4.

4 Support radius optimization

4.1 Optimization procedure

In this part, we are going to discuss an important part of present work: how to use optimization technique to determine a proper set of support radius. To reduce the cost of this process, the control points are firstly divided into several groups in which the control points share identical support radius. Then, a genetic algorithm (GA) is used to perform an unconstrained optimization on the sorted support radius.

4.1.1 Grouping on elemental boundary displacement ratio

The computational cost of optimization highly depends on the number of variables for most optimization methods. Since every control point has its own support radius, theoretically, the number of variables will be equivalent to the number of boundary mesh nodes in our optimization problem. However, even for a simple 2D airfoil, this number can easily goes up to 100, which could lead to a very inefficient optimization. Hence, a variable-reduction process must be proposed. In this study, we decided to sort the control points into few groups and each group share identical support radius.

A simple grouping strategy based on elemental boundary displacement ratio D is employed to sort the control points. And this parameter is defined as the ratio between the areas (2D space) of deformed region A_d and linear element A_l (Figure 5).

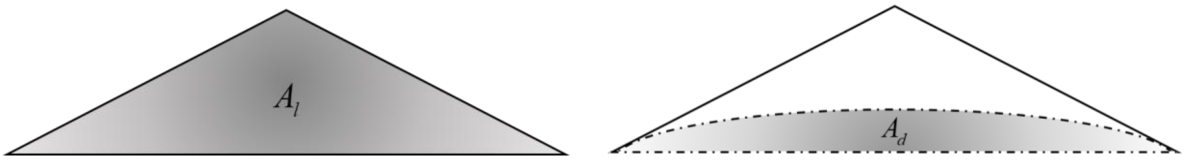


Figure 5: Illustration of elemental boundary displacement ratio which is defined as $D = A_d / A_l$.

Elements with similar D will be divided into same group, and in each group, control points with non-zero displacements will share same support radius. As for a certain zero-displacement control point, its support radius is set as the average value of all the adjacent control points' support radius. In Figure 6, a simple example is used to illustrate our strategy.

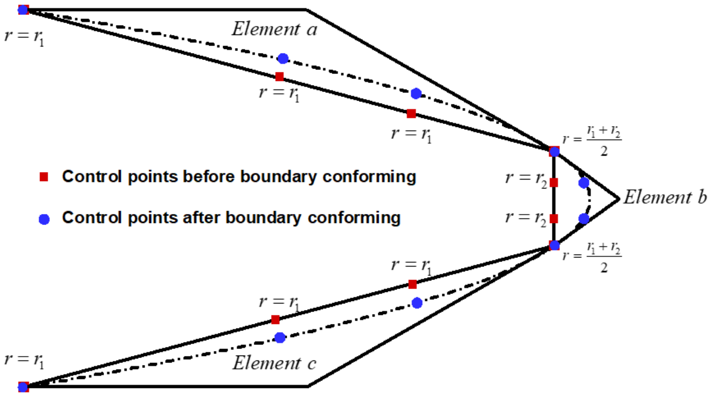

$r = r_1$
Element a
$r = r_1$
$r = r_1$
$r = \frac{r_1 + r_2}{2}$
■ Control points before boundary conforming
● Control points after boundary conforming
$r = r_2$
$r = r_2$
Element b
$r = \frac{r_1 + r_2}{2}$
$r = r_1$
$r = r_1$
$r = r_1$
Element c

Figure 6: Illustration of support radius grouping and determination. 3 Elements are divided into 2 groups: element a, c and element b according their boundary displacement ratios.

4.1.2 Optimization using genetic algorithm

Assume there are n mesh elements attached to the curved boundary, and $\boldsymbol{r}$ is the sorted support radius vector. Our objective function here is the geometrical similarity of the worst element. Then, the optimization problem can be defined as:

$$\begin{gathered} \min f(\boldsymbol{r}) \\ f(\boldsymbol{r}) = \max(Q_i) \quad i = 1, \ldots, n \end{gathered} \tag{10}$$

A genetic algorithm (GA) is adopted to solve this problem. Genetic algorithm is a global optimization method belongs to the larger class of evolutionary algorithms (EA), and it comes from the process of natural selection.

In our study, the genetic representation is done by Binary Gary Code, and $f(\boldsymbol{r})$ is used as the fitness function. To breed a new generation, a tournament selection is applied. As for the genetic operators, we choose two-point crossover and binary mutation. The whole optimization algorithm is very conventional, however, there is one trick we use to accelerate the convergence. We did few unconstrained optimization to determine the support radius. In all test cases, a common situation is observed: the support radius of group with larger elemental boundary displacement ratio is larger. Hence, after a new population generated, we reset the support radius to an ascending array and make sure that larger support radius is assigned to group with larger boundary displacement ratio. Nevertheless, most optimization algorithm, such as gradient based method, is not suitable for this accelerating technique and that is one of the important reason we select GA as our optimization method.

4.1.3 Stopping criterion

We now present a stopping criterion for the optimization algorithm. To determine a good stopping point for the procedure, we define two requirements:

(1). All the first layer mesh elements are valid.

(2). The optimization process converges.

In practice, we will simply check the scaled Jacobian of all first layer elements and the gradient of objective function for latest 4 iterations. When both of them are satisfied, the procedure halts.

4.2 Method Validation

Before applying our method to complex geometry in practical aerodynamic analysis, a simple mesh of a 16% relative thickness ellipse (Figure 7) is tested as a validation case to examine the validity of our method.

We first apply RBF interpolation with identical support radius ($r = 1$) for every control point (Figure 8). It's easy to find that the quality of resultant curvilinear mesh is very poor and clearly, the geometrical pattern of mesh elements is very different from initial linear mesh. Obviously, this mesh does not satisfy the principle of *posteriori* approach mentioned in section 3.2.

Next, we are going to try our improved method. There are totally 14 elements attached to the ellipse and we will divide them into 3 groups according to their elemental boundary displacement ratios (Figure 9). Then, genetic algorithm start to optimize the sorted support radius. Since genetic algorithm is an algorithm with randomness, we repeat this test case for 10 times. In this validation case, the average cost of determining a proper set of support radius and performing interpolation to interior mesh nodes is 0.843 second. Note that we will also use same strategy to count computational cost in all examples at section 5.

Then, we use a worst set of support radius to generate curvilinear mesh, and the resultant mesh (Figure 10) is still much better than identical support radius. To show the difference between meshes more clearly, we decide to perform a comparison of the curvilinear element with worst geometrical similarity (Figure 11). It's easy to find that after RBF interpolation with identical support radius ($r = 1$), the element become more distorted and that's the reason scaled Jacobian and geometrical similarity are getting worse simultaneously. However, if one adopt RBF interpolation with optimized support radius, the element become valid and geometrically much more similar to the linear element.

After analyzing the quality of certain mesh element, we calculate the scaled Jacobian of whole meshes (Figure 12). In the histograms we notice that using identical support radius to perform RBF interpolation is not able to produce valid mesh. Also, besides the invalid elements, the mesh contains some poor-quality elements. In contrast, interpolation with optimized support radius can generate valid mesh and even the worst element has scaled Jacobian of 0.42.

This validation case shows that support radius optimization is necessary for some curvilinear meshes and our method is able to deal with these meshes within acceptable computational cost.

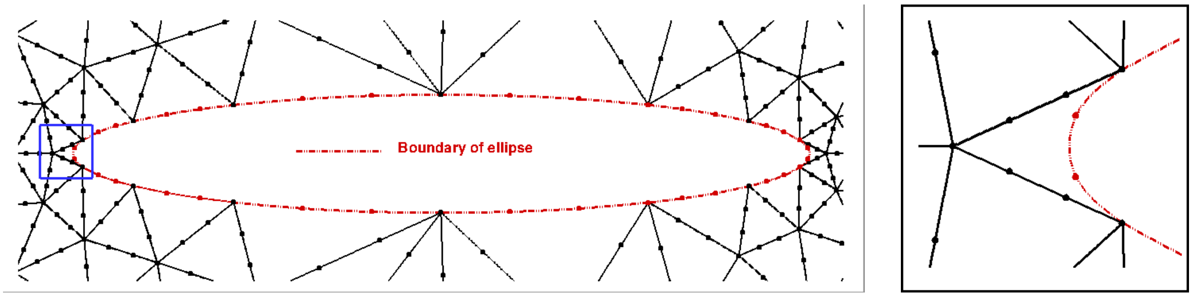


Figure 7: Third order conform-boundary-only mesh of a 16% relative thickness ellipse (left) and the detail of large curvature part (right).

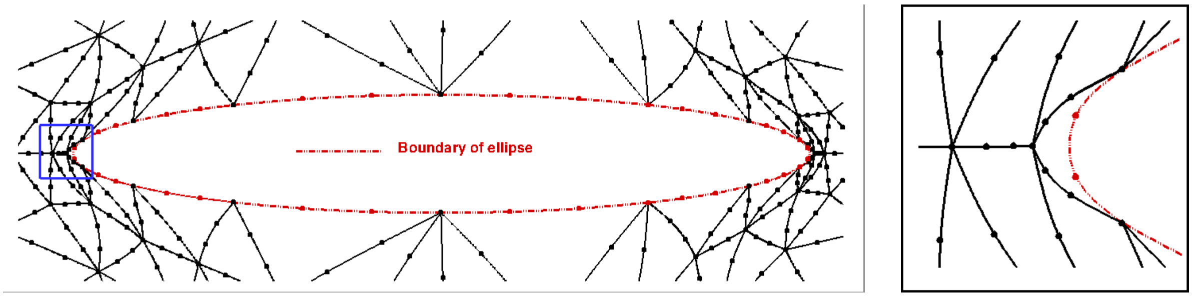


Figure 8: Curvilinear mesh generated by identical support radius.

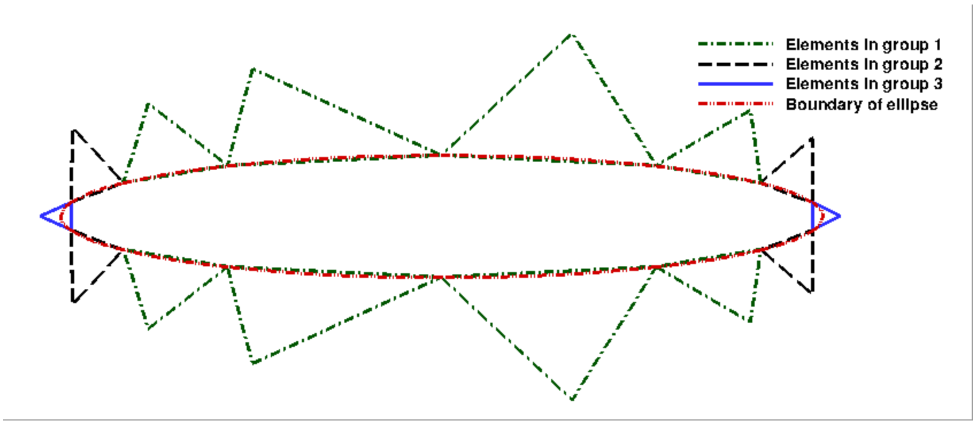


Figure 9: Grouping on mesh elements attached to ellipse.

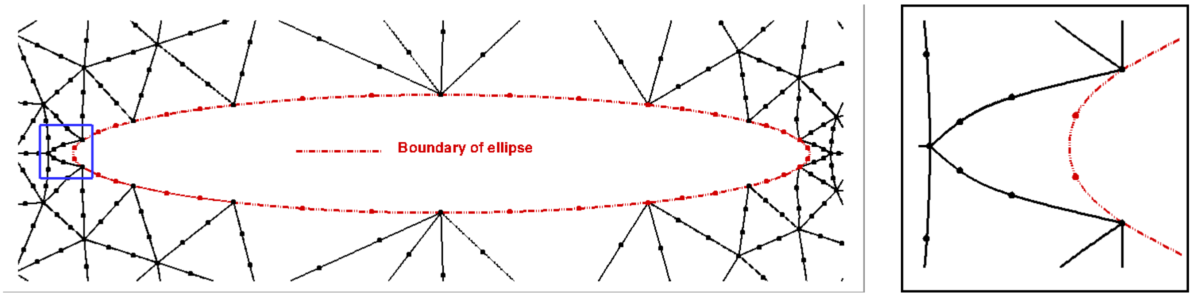


Figure 10: Curvilinear mesh generated by optimized support radius.

5 Examples of application

In this section, we are going to apply our method to three meshes with different kind of curvilinear elements: highly stretched boundary layer mesh with triangular and quadrilateral elements and a very coarse isotropic triangular mesh. Both boundary layer mesh have a polynomial approximation of degree $p = 3$, as for the isotropic mesh, the polynomial order is 4.

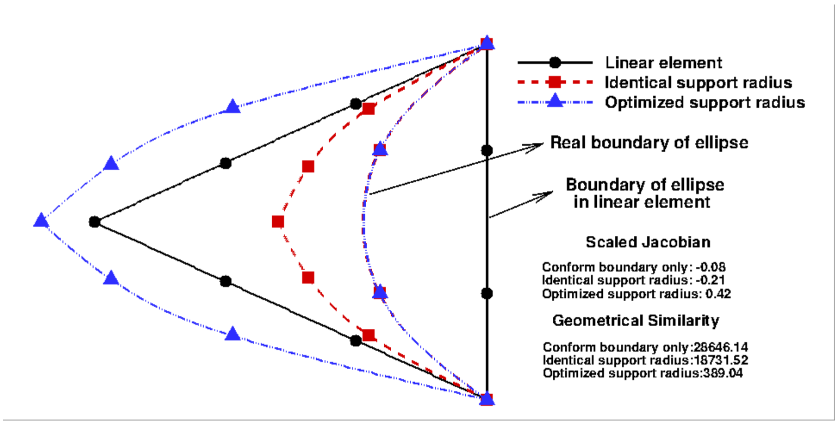
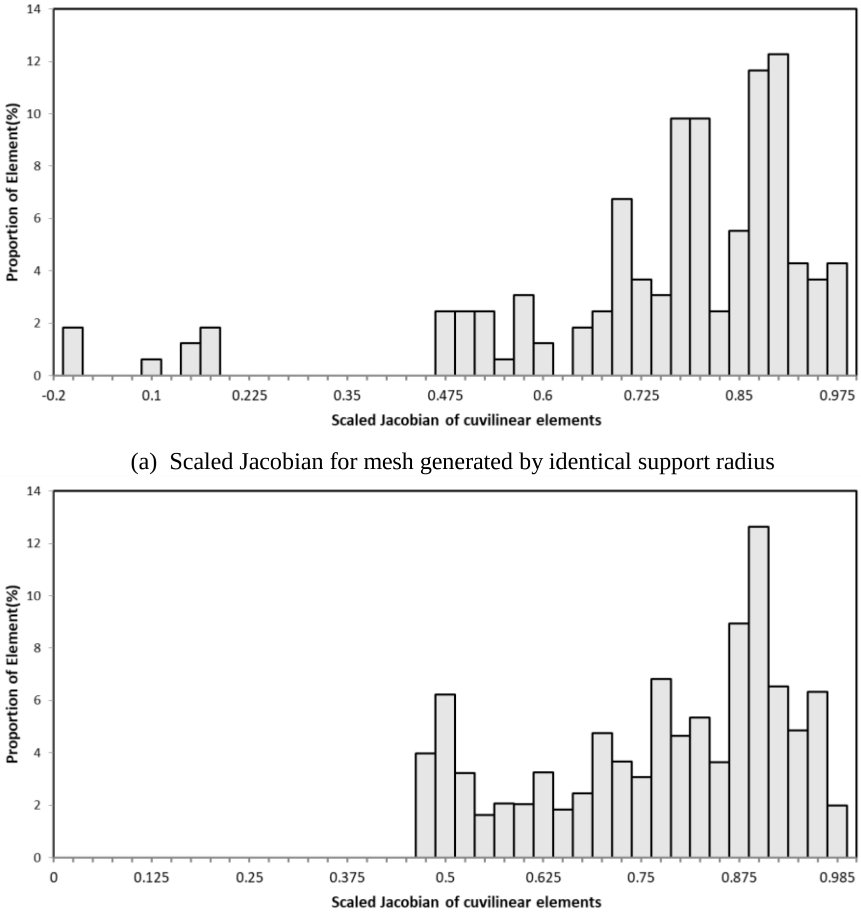

Linear element
Identical support radius
Optimized support radius
Real boundary of ellipse
Boundary of ellipse in linear element
Scaled Jacobian
Conform boundary only: -0.08
Identical support radius: -0.21
Optimized support radius: 0.42
Geometrical Similarity
Conform boundary only:28646.14
Identical support radius:18731.52
Optimized support radius:389.04

Figure 11: Comparison of a certain curvilinear element between different meshes. Clearly, optimized support radius is able to produce better element in both aspects of geometrical similarity and scaled Jacobian.



(a) Scaled Jacobian for mesh generated by identical support radius

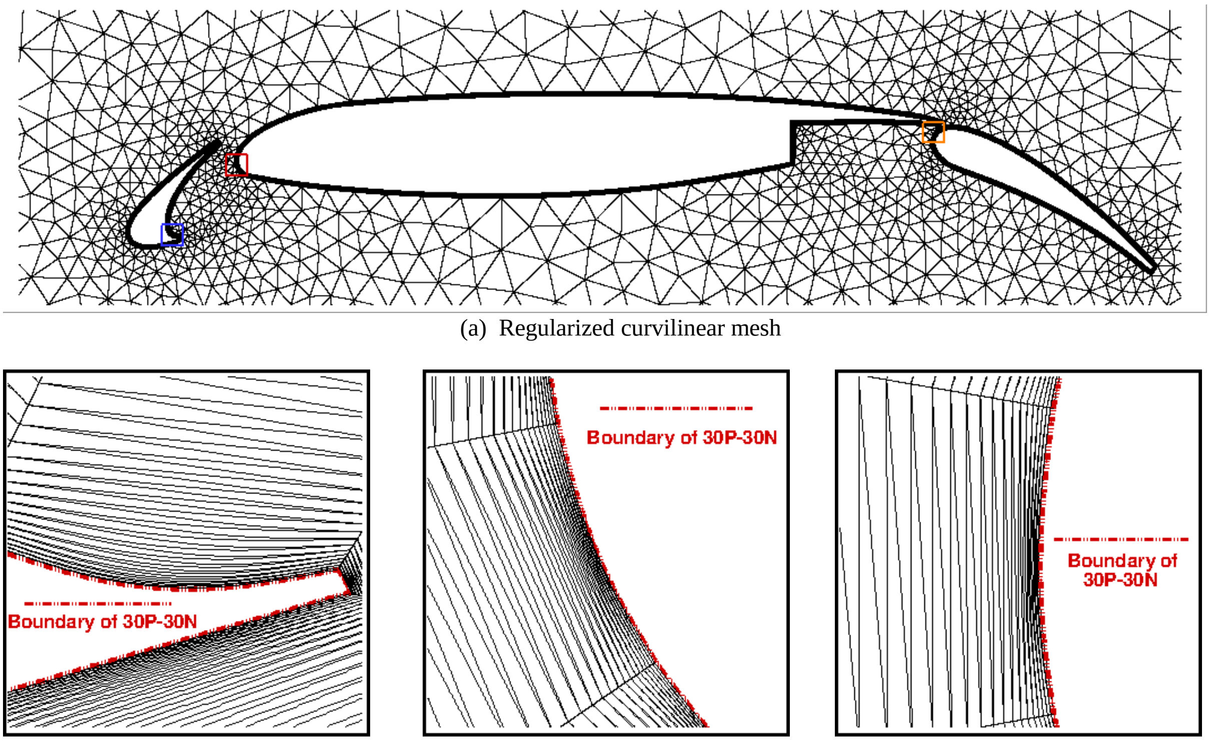
(b) Scaled Jacobian for mesh generated by optimized support radius

Figure 12: Comparison of mesh quality between two curvilinear meshes. Obviously, the quality of mesh generated by optimized support radius is much better than identical one.

Also, in order to show that our method is practical for meshes used in aerodynamic analysis, we consider two classical geometries as the test cases of boundary layer meshes. For these two meshes, if one perform a CFD simulation with polynomial order $p \geq 4$, the y^+ (estimated at the first Gauss point near wall) of first layer mesh elements will be less than 1.

5.1 Anisotropic triangular boundary layer mesh

We start our examples with highly stretched anisotropic triangular elements. A classical high-lift geometry of the 30P-30N is considered. This airfoil was a benchmark case from the first *International Workshop on High Order CFD Methods* (HiOCFD). The mesh in radial direction is much finer than circumferential direction which is a typical situation for high Reynolds number flow simulation and will result in a highly distorted curvilinear mesh. There are totally 142 elements attached to the curved geometry. We first divide them into 4 groups according to their elemental boundary displacement ratios. Then, the build-in optimizer begins to search a proper set of support radius. Once the optimizer find acceptable values, mesh untangling is performed by a RBF interpolation. The whole procedure costs 20.42 seconds and the regularized mesh is placed in Figure 13. The orthogonality near wall is well-remained and we see that even for the parts where the boundary layer is highly distorted, our method is capable to generate well-shaped elements. The scaled Jacobian is plotted in Figure 14, it's clear that all the elements are valid.



(a) Regularized curvilinear mesh

(b) Detail of slat cusp (c) Detail of leading edge in main element (d) Detail of leading edge in flap

Figure 13: Curvilinear triangular boundary layer mesh of 30P-30N. Figure (b), (c) and (d) show the details of large curvature parts from slat, main element and flap, separately.

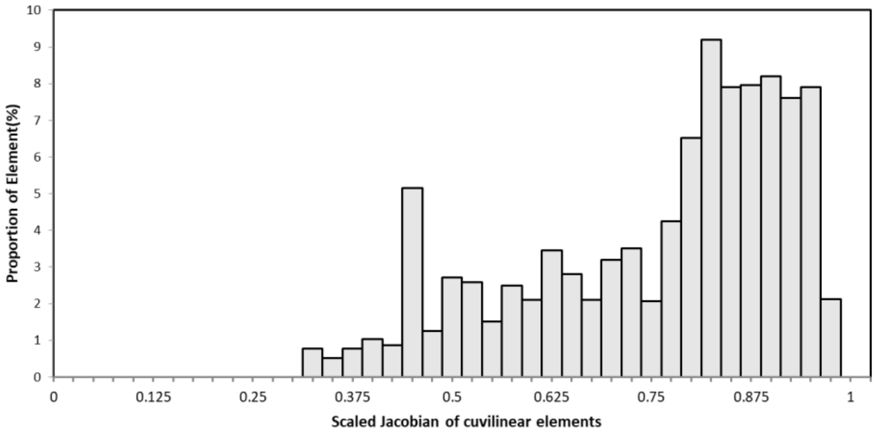


Figure 14: Scaled Jacobian of Curvilinear 30P-30N mesh.

5.2 Quadrilateral boundary layer mesh

Some high order CFD solvers are more efficient if quadrilateral elements are employed rather than triangular elements. To demonstrate our method also works for quadrilateral elements, an airfoil RAE2822 is selected as the test case. There are 56 elements attached to the airfoil and they are divided into 4 groups. Our algorithm spends 4.43 seconds to produce the final curvilinear mesh (Figure 15). In the untangled curvilinear mesh, even the highly stretched and distorted elements in leading edge are well regularized. Also, the quality of whole mesh is good, only a few elements have scaled Jacobian below 0.5.

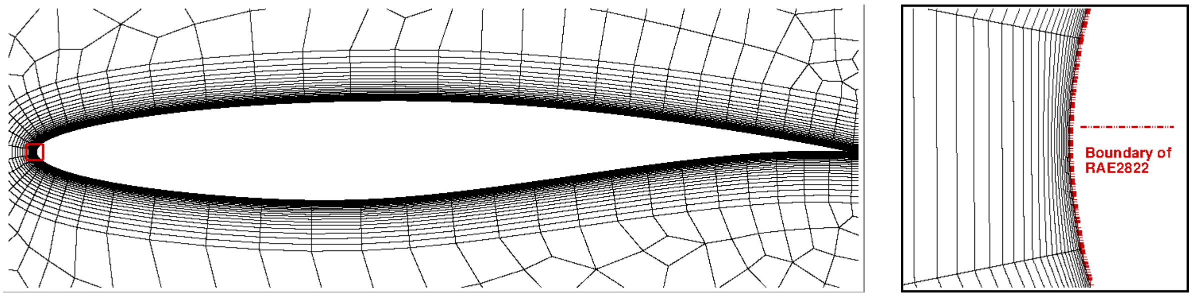


Figure 15: Curvilinear quadrilateral boundary layer mesh of RAE2822.

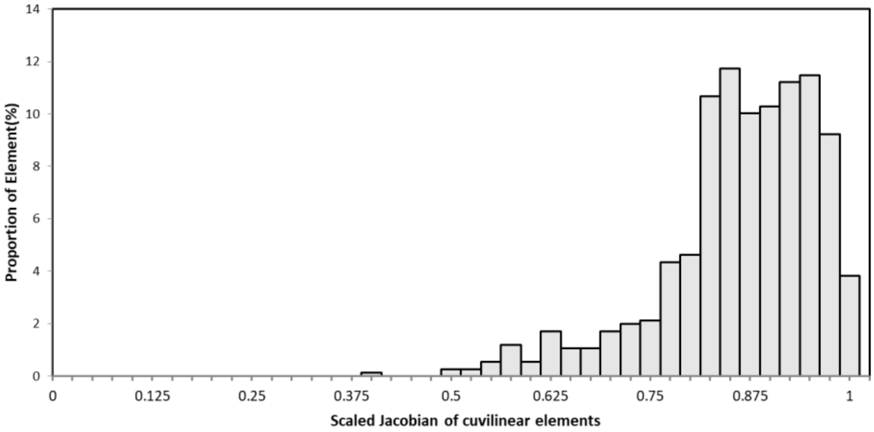


Figure 16: Scaled Jacobian of Curvilinear RAE2822 mesh.

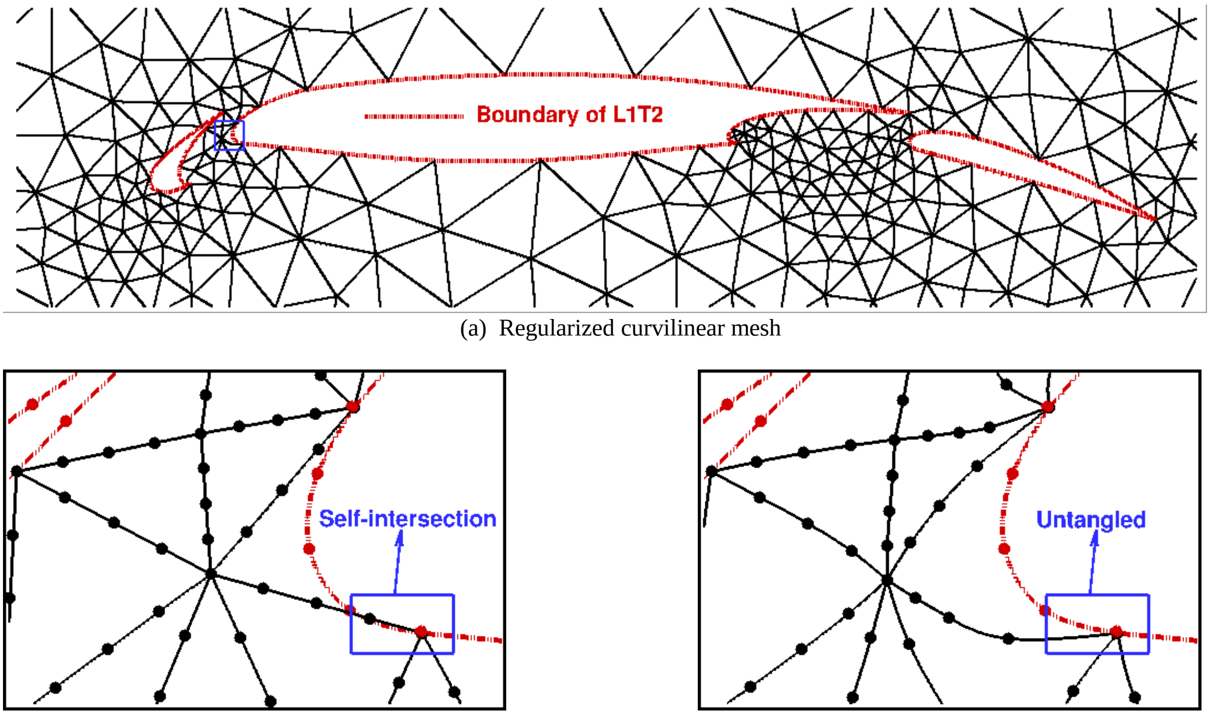
5.3 Isotropic triangular mesh

Coarse isotropic triangular mesh is also frequently adopted in high order community, in order to show that our method can deal with this kind of mesh, we apply the method to another high lift airfoil L1T2 with very coarse isotropic triangular elements (Figure 17(a)). In this case, only 47 elements are attached to the airfoil boundary and they are sorted into 4 groups. The conform-boundary-only mesh is invalid at leading edge of main element (Figure 17(b)). After RBF interpolation, we regularize the mesh successfully, and the invalid element become valid (Figure 17(c)). In this case, the whole untangling process lasts 10.52 seconds. Comparing to the last two examples, we note that our method is more efficient dealing with highly stretched mesh. However, for isotropic mesh, the computational cost is also acceptable.

6 Conclusion and Future Work

We improve the robustness of high order mesh generation using RBF interpolation with compact support by introducing an optimization process of support radius and propose a fast estimation of curvilinear mesh quality to make the algorithm more efficient. In all numerical experiments, we were able to generate valid mesh with well-shaped elements within acceptable computational cost.

Ongoing work includes extending the mesh quality indicator Q and our improved method to highly distorted 3D complex unstructured curvilinear elements. As for the untangling method itself, the remained requirements are mainly about efficiency. To be honest, this issue has been considered from the first place. One of the most important reasons we adopt RBF interpolation to untangle mesh element and genetic algorithm to optimize support radius is that the communities have already developed many methods to improve their efficiency. For RBF interpolation, a number of data reduction strategies have showed their potential of speeding up the mesh deformation process. As for genetic algorithm, several parallel implementations have been developed for decades. We are going to adopt some sophisticated technique to make our method more efficient and useful and these would lead to a practical high order mesh generation program.



(a) Regularized curvilinear mesh

(b) Detail of leading edge in main element (conform-boundary-only mesh)

(c) Detail of leading edge in main element (untangled mesh)

Figure 17: Curvilinear isotropic triangular mesh of L1T2. In conform-boundary-only mesh, some mesh elements are self-intersected. After RBF interpolation, these elements are well untangled.

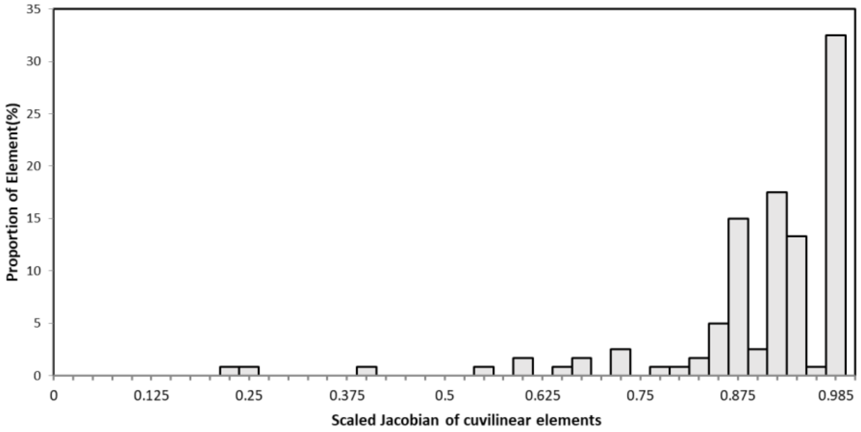


Figure 18: Scaled Jacobian of Curvilinear L1T2 mesh.

References

[1] Hesthaven J S, Warburton T. Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Springer Publishing Company, Incorporated, 2007.

[2] Liu Y, Vinokur M, Wang Z J. Spectral difference method for unstructured grids I: Basic formulation. Journal of Computational Physics, 2006, 216(2):780-801.

[3] Wang Z J, Liu Y, May G, et al. Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations. Journal of Scientific Computing, 2007, 32(1):45-71.

[4] Huynh H T. A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods. AIAA Journal, 2007.

[5] Toulorge T, Desmet W. Curved Boundary Treatments for the Discontinuous Galerkin Method Applied to Aeroacoustic Propagation. AIAA Journal, 2015, 48(2):479-489.

[6] Bernard, P. ‐ E, Remacle, J. ‐ F, Legat V. Boundary discretization for high ‐ order discontinuous Galerkin computations of tidal flows around shallow water islands. International Journal for Numerical Methods in Fluids, 2010, 59(5):535-557.

[7] Wang Z J, Fidkowski K, Abgrall R, et al. High ‐ order CFD methods: current status and perspective. International Journal for Numerical Methods in Fluids, 2013, 72(8):811-845.

[8] Xie Z Q, Sevilla R, Hassan O, et al. The generation of arbitrary order curved meshes for 3D finite element analysis. Computational Mechanics, 2013, 51(3):361-374.

[9] Hartmann R, Leicht T. Generation of unstructured curvilinear grids and high ‐ order discontinuous Galerkin discretization applied to a 3D high ‐ lift configuration. International Journal for Numerical Methods in Fluids, 2016, 82(6):316-333.

[10] Persson P-O, Peraire J. Curved Mesh Generation and Mesh Refinement using Lagrangian Solid Mechanics. 47th AIAA aeroapace sciences meeting and exhibit, Orlando (FL), USA, AIAA paper 2009-949.2009.

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

[12] D. Moxey, D. Ekelschot, Ü. Keskin. High-order curvilinear meshing using a thermo-elastic analogy . Computer-Aided Design, 2016, 72:130-139.

[13] Fortunato M, Persson P O. High-order unstructured curved mesh generation using the Winslow equations. Journal of Computational Physics, 2016, 307:1-14.

[14] Remacle J F, Toulorge T, Lambrechts J. Robust Untangling of Curvilinear Meshes. Journal of Computational Physics, 2013, 254(12):8-26.

[15] Gargallo-Peiró A, Roca X, Sarrate J. A surface mesh smoothing and untangling method independent of the CAD parameterization. Springer-Verlag New York, Inc. 2014.

[16] Gargallo-Peiró, Roca X, Peraire J, et al. Defining Quality Measures for Validation and Generation of High-Order Tetrahedral Meshes. Computer Animation & Virtual Worlds, 2013, 20(4):473-489.

[17] Sherwin S J, Peiró J. Mesh generation in curvilinear domains using high-order elements. International Journal for Numerical Methods in Engineering, 2002, 53(53):207-223.

[18]Ims J, Duan Z, Wang Z J. meshCurve: An Automated Low-Order to High-Order Mesh Generator. 22nd AIAA Computational Fluid Dynamics Conference,Dallas(TX), USA. 2015.

[19] Liu X, Qin N, Xia H. Fast dynamic grid deformation based on Delaunay graph mapping. Journal of Computational Physics, 2006, 211(2):405-423.

[20] Zhang W, Gao C, Zhengyin Y E, et al. Research Progress on Mesh Deformation Method in Computational Aeroelasticity. Acta Aeronautica Et Astronautica Sinica, 2014, 35(2):303-319.

[21]Buhmann M D. Radial Basis Functions: Theory and Implementations. Cambridge Monographs on Applied and Computational Mathematics. 2003.

[22] Boer A D, Schoot M S V D, Bijl H. Mesh deformation based on radial basis function interpolation. Computers & Structures, 2007, 85(11):784-795.

[23] Rendall T C S, Allen C B. Efficient mesh motion using radial basis functions with data reduction algorithms. Journal of Computational Physics, 2009, 228(17):6231-6249.

[24] Rendall T C S, Allen C B. Reduced surface point selection options for efficient mesh deformation using radial basis functions. Journal of Computational Physics, 2010, 229(8):2810-

2820.

[25] Gillebaart T, Blom D S, Zuijlen A H V, et al. Adaptive radial basis function mesh deformation using data reduction. Journal of Computational Physics, 2016, 321(C):997-1025.

[26] Sastry S P, Zala V, Kirby R M. Thin-plate-Spline Curvilinear Meshing on a Calculus-of-Variations Framework. Procedia Engineering. 2015, 124:135-147.

[27] Zala, V., Shankar, V., Sastry, S.P. et al. Curvilinear Mesh Adoptation using Radial Basis Function Interpolation and Smoothing. J Sci Comput. 2018.

[28] Kashi A, Luo H. Curved mesh generation using radial basis functions. 46th AIAA Fluid Dynamics Conference, Washington (D.C.), USA. 2016.

[29] Gargallo-Peiró A, Roca X, Peraire J, et al. Defining Quality Measures for Validation and Generation of High-Order Tetrahedral Meshes. Proceedings of the 22nd International Meshing Roundtable. Springer International Publishing, 2014:109-126.

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