Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable (IMR25)

An Equal-Space Algorithm for Block-Mesh Improvement

Jin Yao, Douglas Stillman

Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94551, USA

Abstract

A simple smoothing algorithm is proposed for general block-structured meshes. The basic method converts a multi-dimensional problem of mesh-smoothing to a set of one-dimensional problems of length-measurement (or similar geometrical operations). The method is robust, easy to implement, and provides nearly uniform spacing between mesh surfaces. Variations with special features to the basic algorithm are also briefly described. A successive-over-relaxation (SOR) operation can be applied to some of the variations and achieve a convergence rate several times higher than traditional methods.

Peer-review under responsibility of organizing committee of the 25th International Meshing Roundtable (IMR25).

Keywords: Block-structure; Mesh-size; Equal-space; Mid-point; Directional line-sweep; Successive-over-relaxation (SOR)

Nomenclature

• stencil	spatial entity carrying geometry of elements that share a given node
• basic-stencil	logically one-dimensional stencil in physical space
• equal-space-point	spatial point that has equal space to the boundary of stencil
• mid-point	equal-space-point on basic-stencil
• mid-line	basic-stencil defined by equal-space-points (as improved mesh-line)

1. Introduction

Mesh smoothing algorithms have been utilized for many years successfully in the generation of meshes for structural mechanics applications. They have proven reasonably effective for mesh sizes involving as many as hundreds of thousands of elements in three dimensions, although they are somewhat costly. In certain fluid mechanics applications, however, we routinely face meshes on the order of one hundred million elements, with a billion elements

* Jin Yao. Tel.: +1-925-424-4657
E-mail address: yao2@llnl.gov

Peer-review under responsibility of organizing committee of the 25th International Meshing Roundtable (IMR25).

on the horizon. These meshes are typically generated with direct node location algorithms, but it would be highly advantageous to have smoothing algorithms that work across the entire mesh domain to provide a globally optimized grid. Thus, we seek an algorithm that is relatively inexpensive to run on up to one billion elements or more and has convergence improved over existing iterative mesh smoothers.

One of the most popular smoothing methods for blocked structured meshes is equi-potential relaxation. It can be derived from a variational principle based on the theory of differential geometry ([2], [6]). An equi-potential method has the advantages of producing smooth meshes, robustly with no mesh-folding, and is easy to code. However, there are issues associated with the original equi-potential relaxation method ([1]) such as slow convergence, grid attraction with curvilinear meshes, and poor mesh quality near a concave boundary that may cause numerical problems for a simulation.

Efforts to modify the original equi-potential method have been made, and certain improvements are achieved such as grid attraction prevention, however with the side effect of slowing down the convergence ([7], [9]). A penalty function can be applied on a concave boundary to improve mesh quality with an equi-potential method but usually is not robust ([6]). It is fair to say the equi-potential mesh-relaxation, while being very successful (particularly with a Cartesian mesh), still has unresolved issues.

Another effective mesh smoothing method is angle-based ([3]) employed in the the original INGRID ([4]) mesh-generation package. This is simple to code and works for an unstructured mesh as well. It shares many nice features with an equi-potential method when applied to a block-structured mesh. However, it also has issues such as shrinking mesh-size near a reduced connectivity point (and expanding mesh-size with an enhanced connectivity point).

The element-metric based methods with MESQUITE ([5]) generally provides a good mesh-quality for a unstructured mesh. However it does not utilize the regular topology of a block-structured mesh and could have a slow convergence with a global optimization for a mesh of a great many elements.

In this paper a new smoothing approach is proposed for a block-structured mesh. The idea is to evenly space mesh-lines/surfaces ([11], [12]). With the proposed method, the coordinate of an updated node is not a direct combination of nodal coordinates as with an equi-potential method. Instead, for updating a given node a *equal-space-point* is computed with a given geometrical rule in a stencil consisting of the elements directly linked to the node.

An *equal-space-point* keeps an even spacing from opposite walls of a stencil in all (logical) directions, thus to ensure the mesh surfaces are separated with an equal space in each direction. Various ways to estimate the *equal-space-point* are investigated in this paper. The most basic one employs only measurement of arc-length on a mesh-line and shall be explained in detail throughout the text. Other choices are briefly described before the numerical examples are shown.

In general, a uniform mesh size provides better numerical accuracy for simulations. The new method is aimed at improving the mesh quality near concave boundaries and irregular connectivity points by evenly spacing the mesh surfaces globally. Similar to the equi-potential relaxation and the angle-based method, the equal-space relaxation produces smooth mesh-lines, and is robust with no mesh folding. In addition, the proposed method also naturally prevents grid attraction effects, and produces good mesh quality on a concave boundary or at an irregular connectivity point.

Furthermore, the proposed method allows point redistribution to be separately done on each mesh-line. A sweep can be performed in a single logical direction at a time, thus simplifying the coding effort.

In situations where convergence speed becomes important, a one-dimensional SOR (*successive over relaxation*) operation can be applied on the arc-length of a mesh-line, and to achieve a convergence rate an order of magnitude better than the equi-potential method.

In this article, we explain how the proposed mesh improvement method works with a unified operation on a triplet of nodes on a mesh line, for a block-structured mesh, in one-dimension, two-dimensions, and three-dimensions. The reader will find a multi-dimensional problem of mesh improvement can be converted to a length measurement on logically one-dimensional stencils with the proposed basic method. This is a feature that other mesh improvement methods probably do not share.

We first introduce the idea of 'equal-space-point' and a logically one-dimensional 'basic-stencil'. Then we explain how a two-dimensional equal-space point is computed with a logically two-dimensional stencil based on the (logically) one-dimensional equal-space points on the walls of the 2D stencil with figures. We then describe how to

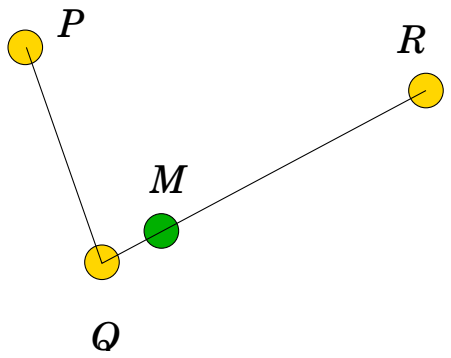

Fig. 1. A triplet of ordered points (P, Q, R) defines a basic-stencil. PQ, and QR are straight-line-segments. A basic-stencil can be thought of as a triplet of nodes on a multi-dimensional mesh-line. Point M evenly divides the length (**PQ** + **QR**) so is called a *equal-space-point* (or a *mid-point*).

compute a three-dimensional equal-space point with a logically 3D stencil based on the (logically) two-dimensional equal-space-points on the walls of the three-dimensional stencil.

Next we describe how to implement the basic algorithm as a one-dimensional-sweep method. The treatment of irregular connectivity in 2D and 3D follows. After that we discuss the behavior of the basic method on a concave boundary and a treatment for possible unsmoothness caused by a unsmooth fixed boundary mesh. Following that, some other variations of algorithms to compute an equal-space point are briefly described in addition to the proposed basic equal-length-dividing algorithm.

Numerical examples are then provided to support the conclusion.

2. Mesh improvement with equal-space-points

Poor mesh quality usually occurs where mesh-lines/surfaces are not evenly spaced. With a block-structured mesh, it is trivial to equally space the mesh-lines in the logical space. However, it is not so easy to map the node position from logical space to physical space. Even when such a map can be performed, it does not necessarily provide even spacing of mesh-lines/surfaces in the physical space because the Jacobian of the mapping can vary.

The proposed method is aimed at evenly spacing surfaces directly in the physical space for a specified region. For this purpose, an *equal-distance-point* is computed for some stencil associated with a node and the mesh surfaces are updated with the *equal-distance-points* by a geometrical rule.

2.1. A basic-stencil and its equal-space-point (mid-point)

A triplet of points linked by two *straight-line-segments* as points (P, Q, R) shown in fig. 1 is defined as a *basic-stencil*. The proposed method is based on a geometrical operation that locates an *equal-space-point* on a basic-stencil on a mesh-line.

A natural way to select an equal-space-point is to choose the point that equally divides the total length $|PQ| + |QR|$. In most of this paper we use the above equal-length-dividing point, and often call such an equal-space-point a 'mid-point'.

Updating a point directly to the equal-distance point on a basic-stencil is a one-dimensional mesh smoothing algorithm. It would distribute points on a spatial curve with an equal arc-length between neighbors when converged.

2.2. A 2D regular stencil and its equal-space-point

In fig. 2, a two-dimensional regular stencil in the physical space is shown with the left figure. We are going to move the node '0' at center to a better position for mesh improvement. Counting from left to right, there are three 'vertical'

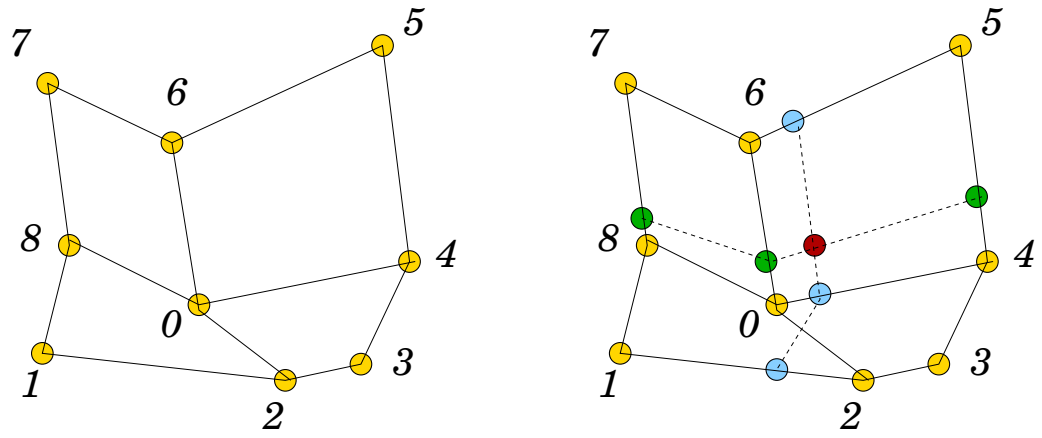

Fig. 2. The left figure shows a regular stencil about node 0 at center, the green points are mid-points on 'vertical' basic stencils; the blue points are mid-points on 'horizontal' basic stencils. Each triplet of mid-points defines a mid-line. The geometrical average of the mid-points of the pair of mid-lines defines a two-dimensional equal-space-point. In this drawing the two final mid-points happen to meet at the intersection of the two mid-lines.

basic-stencils defined by the node triplets (1, 8, 7), (2, 0, 6), and (3, 4, 5). Likewise there are three 'horizontal' basic-stencils defined by the node triplets (1, 2, 3), (8, 0, 4), and (7, 6, 5), counting from bottom up.

We observe in the right figure of fig. 2 the triplet of green points defines its own mid-point, and the triplet of blue points defines its own mid-point as well. We call the basic-stencil defined by the blue (or green) points *'mid-lines'* for convenience. A mid-line can be thought as locally improved position of a mesh-line. By taking the geometrical average of the mid-points on this pair of mid-lines, we have a simple choice of a 2D equal-space-point.

Then, computing a two-dimensional equal-space-point becomes a task of dealing with mid-lines. This simplicity is true in three-dimensions as well. The extension for computing a 2D (logically) equal-space-point in three-dimensions is straightforward. The only difference is that the points are three-dimensional.

2.3. A 3D regular stencil and its equal-space-point

A regular 3D stencil would own 27 basic-stencils, and *nine* 2D (logically) stencils in space. In each logical direction, there are three 2D stencils. Each of them owns a spatial 2D equal-space-point. This triplet of 2D equal-space-points forms a mid-line in the corresponding logical direction and is associated with a mid-point. The geometrical average of the three mid-points (each in a logical direction) defines a three-dimensional equal-space-point.

Again, for computing a three-dimensional equal-space-point, the only operation is finding mid-points on basic-stencils (logically 1D) and taking a geometrical average at the end.

We have taken the geometrical average of mid-point on mid-lines for computing an equal-space-point. One however could define an alternative 3D equal-space-point with intersecting spatial polygons defined by mid-points.

3. Converting a spatial problem to one-dimension

The proposed method depends solely on a geometrical operation that computes an *'equal-space-point'* (or *'mid-point'*). The mid-points can be computed separately before hand. Fig. 3 shows a three-dimensional node. The given node is associated with three mid-points, each one on a mesh-line crossing the node.

With a one-dimensional mesh, each interior node corresponds to a single mid-point. The updated position of the node is this mid-point itself.

In two dimensions, each interior node carries two one-dimensional mid-points.

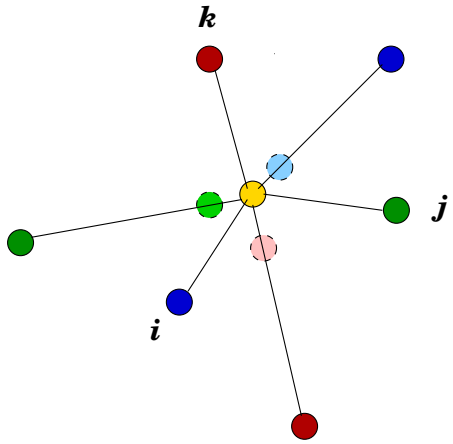

Fig. 3. In three-dimensions, the *yellow* node at the center has three mesh-lines passing through it. The *blue* nodes are the pair of neighbors on the $i-$mesh-line, *green* nodes are the pair of neighbors on the $j-$mesh-line, and the pair of *red* nodes are the neighbors on the $k-$mesh-line. There are three one-dimensional *basic stencils* associated with the yellow node at center. Therefore, three mid-points can be defined, each in a logical direction.

A given 2D regular stencil carries two sets of basic-stencils, each in a logical direction. Each set of basic-stencils with their three mid-points define a mid-line thus a final mid-point, see fig. 2. A 2D equal-space-point is simply the geometrical average of the two final mid-points (or alternately the intersection of the two mid-lines).

In three dimensions, each interior node carries three one-dimensional mid-points as already mentioned above.

A given 3D regular stencil carries *three* sets of 2D (logically) stencils. Each set consists of *three* stencils regarding a logical direction and defines three 2D equal-distance points. Therefore, a triplet of mid-points exists that defines a mid-line in a given logical direction. All together, there are *three* mid-lines, each in a logical direction. A 3D equal-space-point is taken as the geometrical average of the mid-points of the mid-lines.

Therefore, in a given iteration of the proposed method, the approach to update a node depends on mid-points on logically one-dimensional stencils only. In other words, the proposed method converts a multi-dimensional mesh improvement problem to one-dimension.

3.1. A directional sweeping scheme

As shown above, each $k-$dimensional regular node is associated with k 'mid-points', one from each logical direction. Clearly, by employing a one-dimensional algorithm to redistribute points in the arc-length of a mesh-line, the proposed algorithm can be implemented in a fashion of one-dimensional sweeping thus simplifying the coding for better efficiency.

3.2. At an irregular connectivity

A node at an irregular connectivity (for example, a reduced connectivity point) is not associated with a regular stencil, and needs to be dealt with differently. However, the concept of mid-points can still be used, and the treatment is even simpler. We demonstrate the treatment with a reduced connectivity. The treatment is similar for an enhanced connectivity.

At a reduced connectivity point in two-dimensions, we take the geometrical center of the triangle formed by the neighbor nodes directly connected to the given node by mesh-lines. Another choice is taking the center of the inscribed-circle of the above triangle. The latter seems to give a even wider mesh-size.

In three dimensions, two configurations are possible. In the first case, a node is at the joint of *four* hexahedron elements. We take the *geometrical center*, or the center of the *inscribed-sphere* of the tetrahedron formed by the neighbor nodes directly linked to the given node by mesh-lines. In the second case, a given node is shared by three

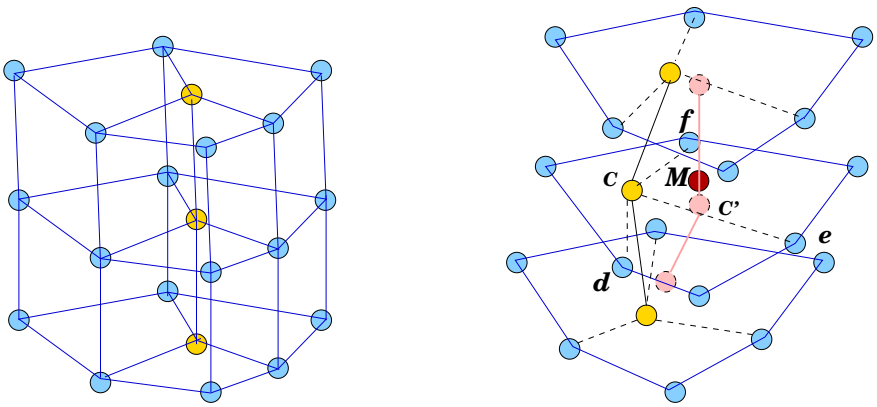

Fig. 4. Left figure shows a stencil for a node at a reduced-connectivity point of the second type. In the right figure, node C is at a spatial, logically 2D reduced connectivity on a patch consist of three faces. d, e, f are neighbor nodes of C connected with mesh-lines on the patch (dash-lines). The geometrical center of *triangle def*, C', is the equal-space-point (marked by a dash-circle) for the patch in middle . Likewise the other two dash-circles mark equal-space-points defined on the other two patches. M is the mid-point on the mid-line (colored *pink*) defined by these three points.

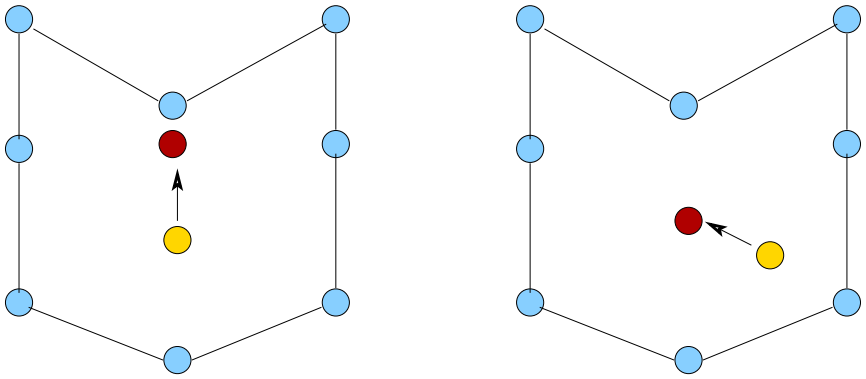

Fig. 5. In the left figure, an equi-potential method can move the yellow node to the position of the red node close to element boundary and reduces the space between neighbor nodes. In the right figure, the proposed method moves a misplaced node to a location with equal space to element boundary.

elements on one-side and three on the other side (fig. 4, left). There are *three* nodes (including the given node) at logically 2D reduced connectivity, each on a patch of three spatial faces (fig. 4). This triplet are on a mesh-line that passes and logically orthogonal to the triple-face patches. Each triple-face patch determines a logically 2D reduced-connectivity point. These three points form a mid-line, and we take its mid-point to update the center node.

A mesh-line defined by a set of nodes with a second kind of reduced-connectivity (as shown in fig. 4) will meet either a first kind of reduced-connectivity point, or the boundary. A reduced-connectivity point is always on boundary or at the joining lines between blocks.

3.3. Near a concave boundary

A conventional mesh-improvement method usually has an issue with the elements close to a concave boundary. Elements tend to squeeze together and lay on the concave portion of a boundary. This behavior introduces thin elements.

The proposed equal-space method with the choice of equal-length-dividing mid-point or equal-distance points behaves differently. For example, an equi-potential method often moves a node originally at an ideal position to a

location near a concave boundary (fig. 5, left). The proposed method does the opposite by computing an updated point which has nearly equal space (or distance) between a pair of opposite boundaries (fig. 5, right), thus producing elements of more equal sizes.

3.4. A treatment for face-smoothness

The equal-length-dividing works well in 3D in general, However, if one of the spatial 2D stencils that define a wall of a 3D regular stencil has fixed nodes that form sharp-angles on a mesh-line, this unsmoothness can be carried inward and this is not desired.

To ensure smoothness, we project each 2D equal-space-point obtained on a spatial logically 2D stencil to a fitting plane defined by the nodes of the 2D stencil in order to flatten the stencil. As the result, the smoothness of interior mesh-lines are improved. One needs to apply this operation only to the near boundary mesh. However, we perform it with all the interior nodes. The mesh sizes near a reduced-connectivity point would be reduced by a little compared to the case of no face-smoothing, but still quite acceptable, and the overall mesh smoothness is improved.

4. Other choices for computing an 'equal-space' point

First of all, the solution by averaging the mid-points on mid-lines in each logical direction can be replaced by intersecting mid-lines (as locally improved mesh lines) in 2D, and intersecting spatial polygons (as locally improved mesh-surfaces) defined by mid-points in 3D. Although the implementation is not as easy, using intersections seems to converge faster consistently.

Not only that, the way to select a mid-point is also flexible. In our numerical practices, the following variations have been evaluated.

4.1. Mid-face line-intersection

With this option, one picks the face center points of a regular 2D stencil, in total 12 of them (fig. 6, left). They form 4 mid-lines, 2 in each logical direction. Then in each direction there is a pair of mid-points that define a line-segment. The equal-space point is the intersection of these two line-segments. In the case that they do not intersect, one picks the concave corner of a quadrilateral formed by the four mid-points.

Our 2D examples of fast convergence in the next section is performed with this choice, utilizing the successive-over-relaxation (SOR) scheme.

4.2. The equal-distance point

This algorithm seeks in a regular stencil (2D or 3D) an *equal-distance-point* that has equal-distances to a pair of opposite boundaries in each logical direction (fig. 6, right). In locating the equal-distance point, a system of nonlinear equations needs to be solved (we use a Newton's method for the root-finding).

In the three-dimensional implementation of locating an equal-distance-point, we use a quadratic Bernstein shape function to interpolate each wall of a regular stencil for computing the distance. For nodes near a reduced connectivity point, we break each element face into 4-triangles.

This choice with equal-distance-point can diverge if a stencil is badly distorted initially. However, it gives probably the most satisfying spacing between mesh-surfaces among our choices, and its solution is very smooth. Employing the equal-distance-points can help to refine a mesh that has no badly twisted stencils.

4.3. A variable dividing ratio

We have taken a ratio of $1/2$ to divide a given length. If required (for example in the case of a stencil with fixed points that are not evenly spaced) this ratio can be adjusted locally.

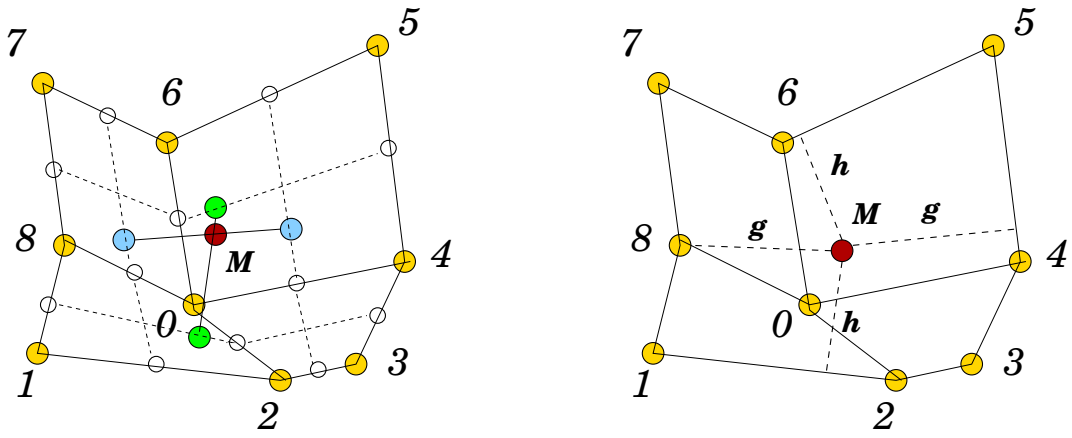

Fig. 6. The left figure shows the choice of mid-face line-intersection. The 12 center-points of faces form 4 mid-lines, with a pair in each direction that define a line-segment. The intersection of the two line-segments gives the equal-space-point M. In the right figure, a point M exists with the same distance g to linked-line-segments $(1, 8, 7)$ and $(3, 4, 5)$, and the same distance h to linked-line-segments $(7, 6, 5)$ and $(1, 2, 3)$. We call M the equal-distance-point defined by the given stencil. It is boundary determined, has no dependence on the position of node 0, and serves as the updated position of the center node 0.

It is evident that the proposed equal-space method (with whatever algorithmic choices described above) define an ideal local distribution of nodes in the case that the mesh-lines are (locally) flat in each logical direction. This statement is also true with an orthogonal curve-linear mesh.

5. Numerical examples

5.1. A case of fast convergence with successive-over-relaxation (SOR)

In the case that the desirable point distribution on a mesh-line is an equal length between neighbor points, a successive-over-relaxation (SOR) method can give a super-linear convergence rate. A SOR method does not update a point directly at the equal-length dividing point between the two neighbors. Rather, it over-relaxes the given point with a factor ω between 0 and 2, such that

$$\ell_n^{i+1} = (1-\omega)\ell_n^i + \frac{\omega}{2}(\ell_{n-1}^i + \ell_{n+1}^i). \tag{1}$$

Where ℓ is the length measurement, n is the index of a node on a given mesh-line, and i is the count of iterations

We take a two dimensional mesh with zigzag mesh-lines similar to the Kershaw mesh [7], [9], [10]. The mesh is smoothed by the original equi-potential method, also by mid-face line-intersection method described in the last section. The two-dimensional equal-space-point is computed with intersecting the two line-segments defined by a pair of mid-points computed with an SOR factor of 1.995 in mid-face-lines in each direction. This equal-space-point immediately updates the node location with a natural double loop over all interior points (boundary points are fixed, outer-loop is with i and the inner-loop with j). The boundary of the mesh is convex in this case, which allows the operation of intersection to perform well.

In this case the proposed method converges over an order of magnitude faster (fig. 7) than the original equi-potential method in early iterations and $four$ times faster in later iterations (fig. 8). The L_2 errors mentioned in the description of figures is defined as the square-root of the average of physical distance squared between a node and its ideal position.

5.2. With a polar geometry

Fig. 9 shows how the proposed mesh-improvement method works with an orthogonal curve-linear mesh. The equal-space nature of the proposed method ensures an ideal spacing of mesh-lines.

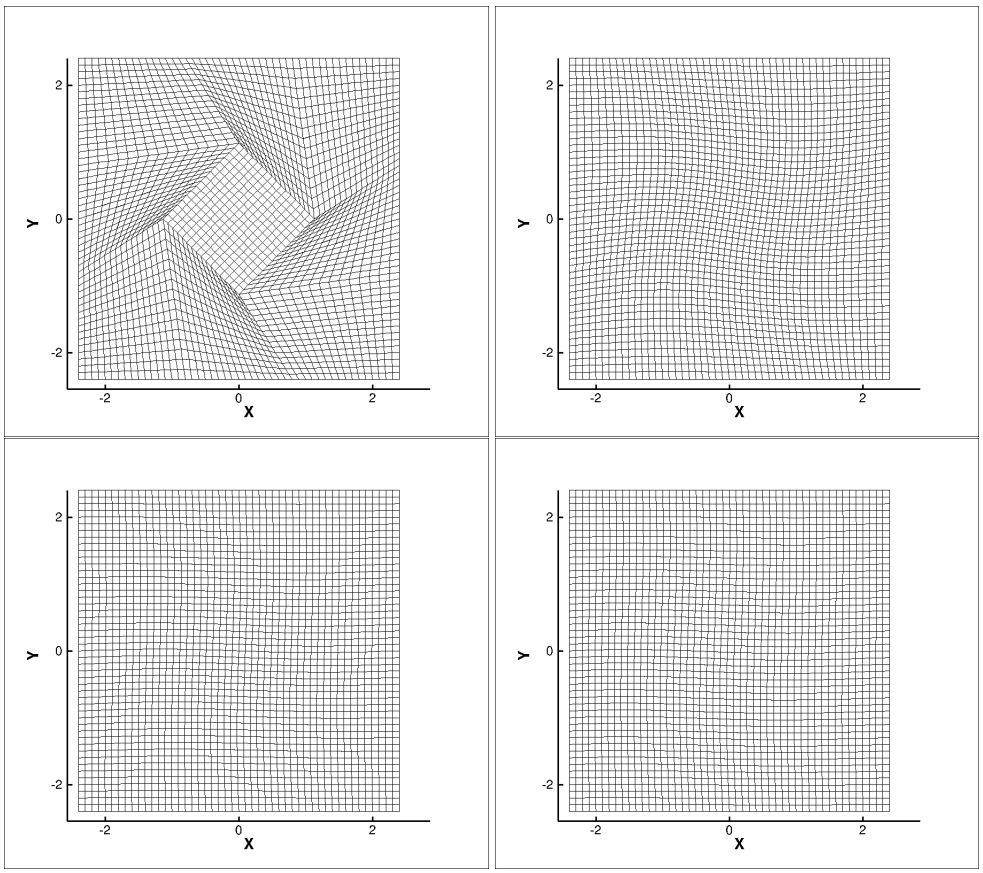

Fig. 7. A mesh with zigzag mesh-lines is shown on the upper-left. It is smoothed by a Winslow-Crowley method to the one on upper-right after 250 iterations (with a L_2 error of 0.092681), to the lower-left one with 500 iterations (with a L_2 error of 0.023690). An even better mesh-quality can be obtained with the mid-face line-intersection method using a SOR factor of 1.995 by only 67 iterations (lower-right, with a L_2 error of 0.023288).

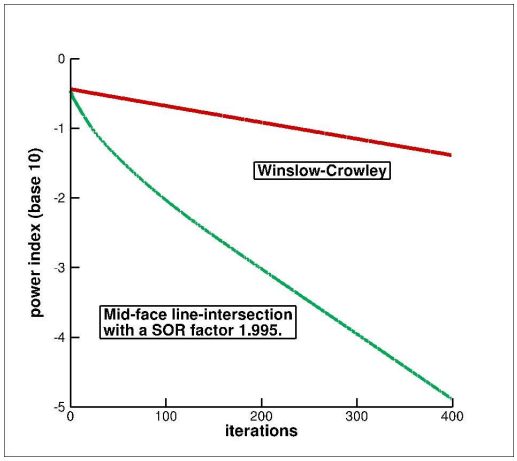

Fig. 8. The L_2 error vs. iteration numbers for the above example. The vertical axis is the power index $\log_{10}(err_{L2})$.

5.3. With a concave boundary (and an enhanced connectivity point)

An example with an enhanced connectivity and a concave boundary is shown next. We take a perfectly symmetric 'star' mesh of *five* blocks with an enhanced connectivity at center. The intersection between the $x-$axis and the

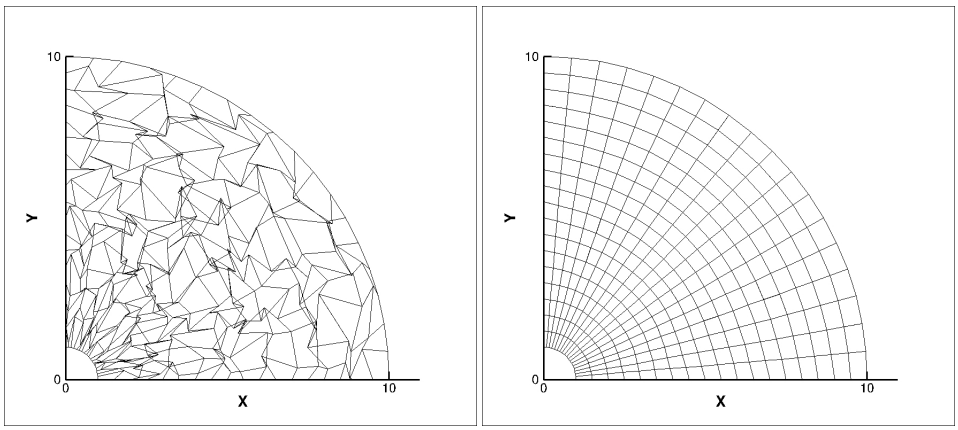

Fig. 9. An initially randomly twisted polar mesh is smoothed by the mid-face line-intersection method with 12 iterations. A successive-over-relaxation is applied with a SOR factor of 1.995.

boundary of the star are at (-10, 0) and (20, 0), with the center at the origin. Each interior node of this mesh is perturbed with a random displacement between -2 and 2 in both directions. As the result, a big portion of the elements are folded. We perform smoothing with both the Winlow-Crowley algorithm and the equal-length-dividing (the basic) algorithm. At the enhanced connectivity point, we simply take the geometrical average of its direct neighbor nodes linked by mesh-lines. This example demonstrates the robustness of the basic algorithm of equal-space-smoothing, and its ability to maintain uniform mesh spacing near a concave boundary and/or an irregular connectivity point over the original equi-potential method. The equal-distance method behaves similarly.

Results similar to the above are obtained for a curved surface mesh as well. For such a mesh, the smoothing operation is done in a surface fitting plane above a given node using orthogonal projections of the nodes in a 2D surface stencil. The updated position of the given node is projected back to the surface. The scheme is a planar 2D local smoothing operation, combined with two projection operations from and back to a curved surface.

5.4. A three-dimensional example

A three-dimensional comparison of the element-size effect near a reduced-connectivity with an angle-based method and the proposed equal-space method is performed. The geometry of a meshed region is a *half-sphere* of a nondimensionalized radius 9, with the upper quarter mesh fixed and the lower quarter mesh to relax.

Figure. 11 show the initially meshed half-sphere improved by three methods: the angle-based algorithm ([3]); the proposed equal-space method with equal-length-dividing mid-points; and with the choice of equal-distance points, each with 100 iterations. The 3D meshes are sliced by a plane defined with a point at (1, 1, 1) and a normal (6/7, 3/7, 2/7).

One can clearly see the angle-based method gives small mesh-sizes around a reduced-connectivity point. However, an equal-space method gives much more uniform mesh sizes throughout with smooth mesh-lines. The option of equal-length-dividing is simple, robust. The equal-distance option provides the largest mesh-size around a reduced-connectivity point. However, this option cannot be directly applied to a very twisted mesh because of possible non-convergence. It can be combined with a more robust smoothing algorithm (say, equal-space) for a more uniform mesh-size.

5.5. Mesh quality measurements

In figs. 12 we plotted the statistical distribution of element sizes with the angle-based-smoothing, the equal-space-smoothing (the proposed basic algorithm), and the equal-distance smoothing (by solving nonlinear equations about physical distances from a node to walls of a stencil) for the smoothing problem above. The element shapes all look normal with each algorithm, thus the mesh-size measurement becomes a proper indicator of mesh quality. The mesh-sizes in the figures correspond to a measure (defined as element volume divided by largest element face area) for

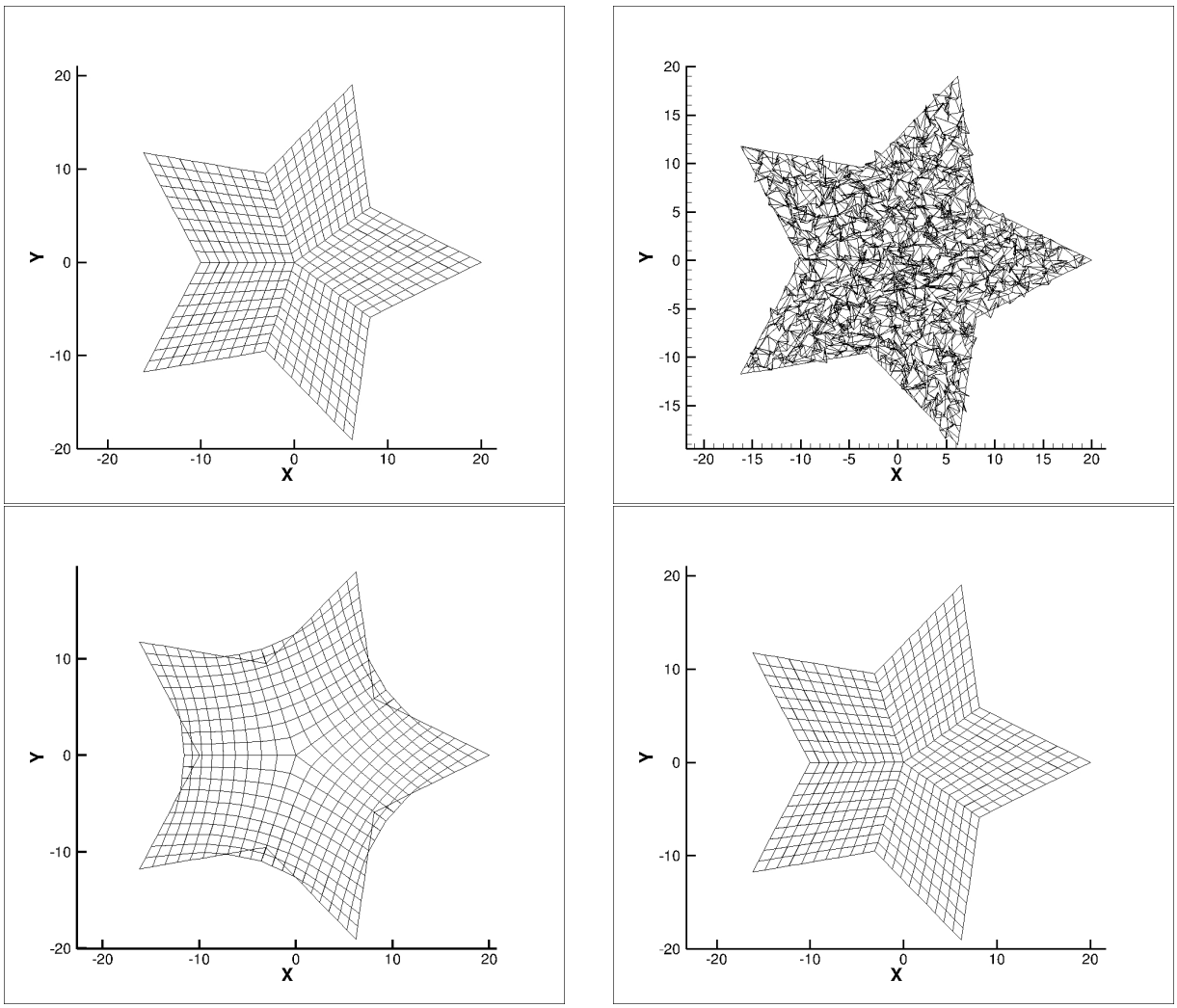

Fig. 10. The figure on upper-left shows the ideal star-mesh. The mesh on upper-right is perturbed (boundary nodes are fixed). The mesh on lower-left is obtained with the original equi-potential method. The mesh on lower-right is smoothed with the equal-length-dividing (the basic) method. 500 iterations are taken with each method to ensure convergence.

explicit time-steps (determined by the smallest mesh-sizes). By observing the smallest element-size, one finds the equal-space method provides a time-step at least twice as big as the angle-based method does. Even better, the equal-distance method gives a time-step at least three times as big (as visually shown near the reduced-connectivity point in figs. 11). In addition, the range of mesh-sizes with the angle-based method is reduced with the new methods, with the narrowest range obtained by finding the equal-distance points. The accuracy of a simulation is expected to be improved correspondingly.

5.6. The cost of an equal-space method

In a single iteration, an equal-space method with square-root computation for lengths costs a little more than an equi-potential method. Nevertheless, the mesh quality near irregular connectivity points and concave boundaries is improved. Moreover, with a SOR scheme, one can expect the cost to be reduced significantly with certain choice of the proposed method.

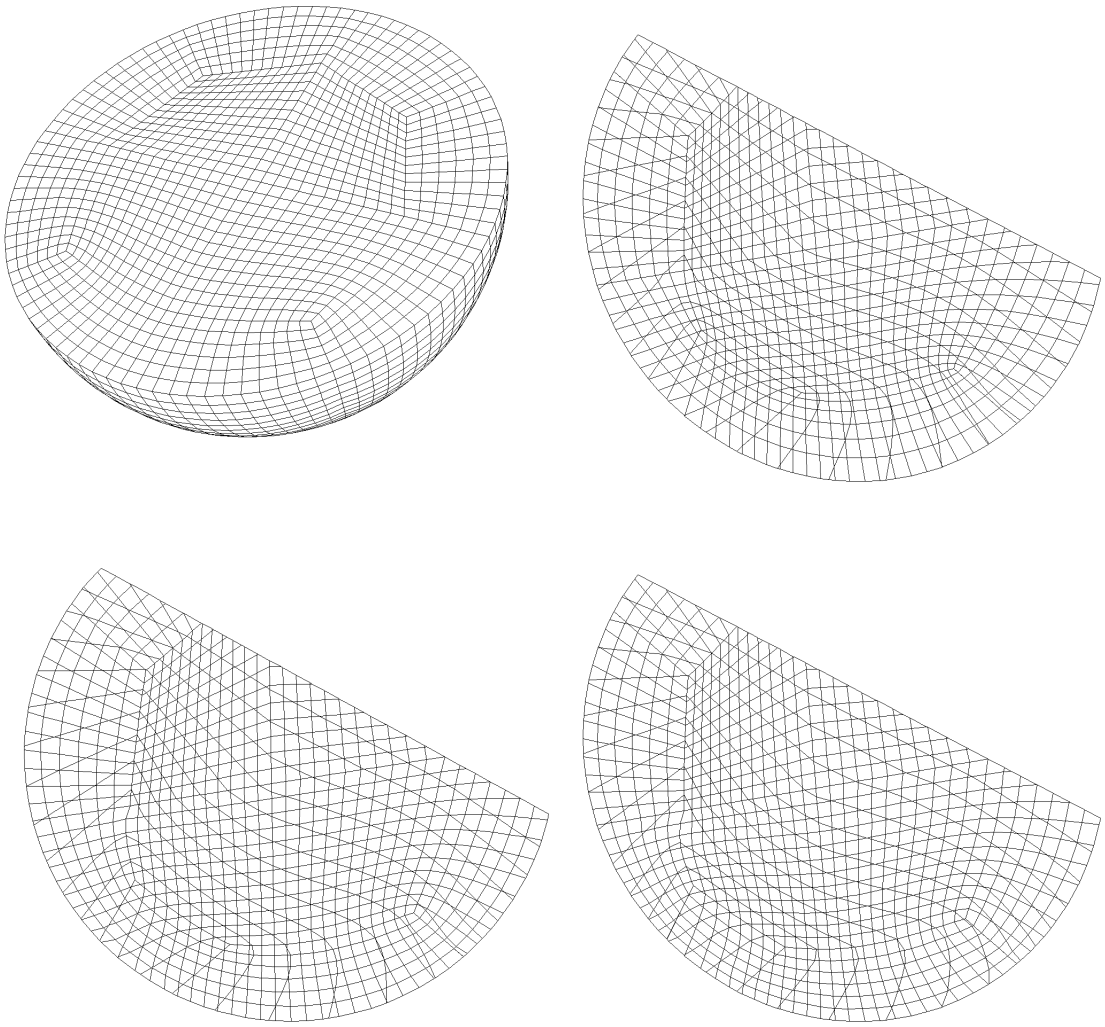

Fig. 11. Above figures show the meshed half-sphere described above sliced by a plane defined by point $(1, 1, 1)$ and a normal vector $(6/7, 3/7, 2/7)$. The upper-left is the initial 3D mesh (before slicing); upper-right for an angle-based smoothing method; lower-left for the proposed equal-space method with the option of equal-length-dividing with a face-smoothing, and lower-right with the option of multi-dimensional equal-distance point.

6. Conclusion

We believe with a given smoothing algorithm, the solution of a mesh problem is boundary determined. A good mesh-smoothing algorithm should take an arbitrary initial configuration and quickly converge to the solution. We presented in this paper a simple equal-space mesh-smoothing method for a general block-structured mesh, with several algorithmic options. By using the equal-space mid-points, a multi-dimensional mesh-smoothing problem is converted to finding a set of mid-points, each on a logically one-dimensional stencil defined by a triplet of points on a mesh-line. A directional line-sweeping scheme can be applied to simplifying the coding and reduce computing cost.

The major benefit of the proposed method is that a nearly uniform mesh spacing can be achieved, especially near an irregular connectivity or a concave boundary. Not only that, with a successive-over-relaxation scheme the proposed method may achieve a much higher convergence rate compared to traditional smoothing methods in certain cases.

The proposed method has not only the usual positive features such as smooth mesh-lines, no mesh-folding, no mesh contraction with a curve-linear mesh, it also produces a mesh quality better than some conventional algorithms

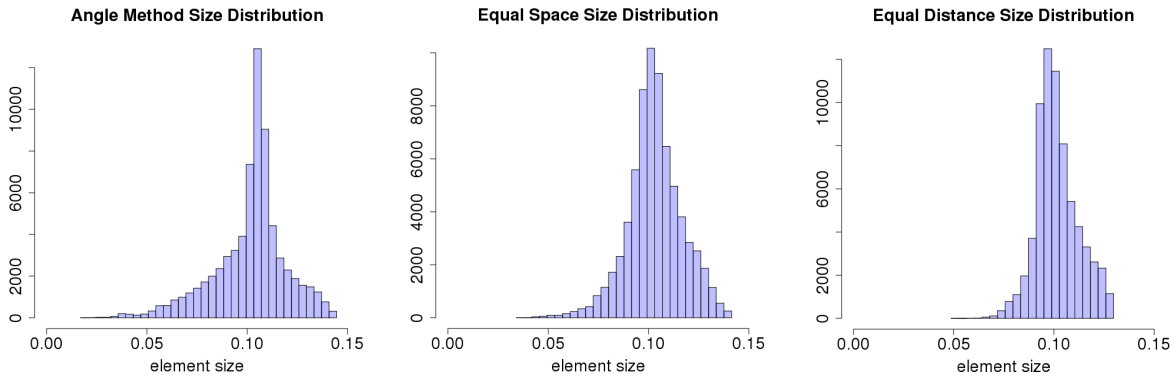

Fig. 12. Distribution of element sizes with different smoothing algorithms. The horizontal axis is the size of elements and with a uniform range. The vertical axis is the count of elements in each bin (ranges are not uniform). The left figure is for the angle-based method, the figure in middle for the basic equal-space method, and the right figure for the equal-distance method.

near a concave boundary / irregular-connectivity. The proposed method costs a little more than a conventional method in a single iteration but this draw-back is overwhelmed by a faster convergence. Furthermore, the simplicity of the proposed basic method with finding the equal-length-dividing points on mesh-lines makes it robust.

The concept of the equal-space-point may be extended to an unstructured mesh. There also can be various possible definitions of an equal-space-point. The proposed equal-space algorithm utilizes the topological regularity of a block-structured mesh and gives natural definition of an equal-space point. Because of its simplicity and robustness, we conclude the proposed equal-space algorithm (as well as the equal-distance algorithm) has the potential to become an effective numerical tool for improvement of general block-structured meshes.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] Winslow, A. M., “Numerical Solution of the Quasi-linear Poisson Equation in a Nonuniform Triangular Mesh”, Journal of Computational Physics, Vol. 1, pp. 149-172, 1967.
[2] Tipton, R. E., “Grid Optimization by Equipotential Relaxation”, Lawrence Livermore National Laboratory Report, July 15, 1992.
[3] Tian Zhou, and Kenji Shimad, “An Angle-based Approach to Two-dimensional Mesh Smoothing”, Proceedings, 9th International Meshing Roundtable, Sandia National Laboratories, pp.373-384, October 2000.
[4] Stillman, D. W., and Hallquist, J. O., ”INGRID: A Three-dimensional Mesh Generator for Modeling Nonlinear Systems”, Lawrence Livermore National Laboratory, UCID-20506, 1985.
[5] Brewer, M.L., Diachin, L. F., Knupp, P. P., and Leurent, T., ”The Mesquite Mesh Quality Improvement Toolkit”, International Mesh Roundtable, 2003.
[6] Patrick Knupp, and Stanly Steinberg, “Fundamentals of Grid Generation”, CRC Press, 1993.
[7] Jun, B.I. “A Modified Equipotential Method for Grid Relaxation”, Lawrence Livermore National Laboratory Report, UCRL-JC-138277. March 21, 2000.
[8] Tipton, R. E., “Mesh Relaxation in Elliptical Coordinates and Notch Points for ALE Hydro-Codes”, Lawrence Livermore National Laboratory Report, Nov.1, 2004.
[9] Matthew J. O’Brien, “Multigrid Methods for Mesh Relaxation”, Lawrence Livermore National Laboratory, UCRL-TR-222074, June 15, 2006.
[10] Kershaw, D. S. “Differencing of the Diffusion Equation in Lagrangian Hydrodynamic Codes”, Journal of Computational Physics, Vol. 39, pp. 375-395, 1981.
[11] Yao, Jin, “An Efficient Line-Sweep Mesh Relaxation Method with Locally Conservative Interface Reconstruction ”, Multimat Workshop, Acachon, France, 2011.
[12] Yao, Jin,“A Mesh Relaxation Study and Other Topics”, Lawrence Livermore National Laboratory Technical Report, LLNL-TR-637101, 2013.