LA-UR-03-2424

Approved for public release;
distribution is unlimited.

Title: Untangling of 2D Meshes in ALE Simulations

Author(s): Pavel Vachal
Czech Technical University at Prague, Czech Republic

Rao V. Garimella, Mikhail J. Shashkov
MS B284, T-7, Los Alamos National Laboratory, Los Alamos, NM USA

Submitted to: Journal of Computational Physics

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (8/00)

Untangling of 2D Meshes in ALE Simulations (LA-UR-03-2424)

Pavel Vachal

Czech Technical University in Prague, Czech Republic

Rao V. Garimella

MS B284, T-7, Los Alamos National Lab., Los Alamos, NM, USA

Mikhail J. Shashkov

MS B284, T-7, Los Alamos National Lab., Los Alamos, NM, USA

Abstract

A procedure is presented to untangle unstructured 2D meshes containing inverted elements by node repositioning. The inverted elements may result from node movement in ALE (Arbitrary Lagrangian Eulerian) simulations of continuum mechanics problems with large shear deformation such as fluid flow and metal forming. Meshes with inverted elements may also be created due to the limitations of mesh generation algorithms particularly for non-simplicial mesh generation. The untangling procedure uses a combination of direct node placement based on geometric computation of the feasible set and node repositioning driven by numerical optimization of an objective function that achieves its minimum on a valid mesh. It is shown that a combination of the feasible set based method and the optimization method achieves the best results in untangling complex 2D meshes. Preliminary results are also presented for untangling of 3D unstructured meshes by the same approach.

Key words:
mesh untangling, ALE (Arbitrary Lagrangian Eulerian) simulations, Rayleigh-Taylor simulation, element validity, triangles, quadrilaterals

Email addresses: vachal@galileo.fjfi.cvut.cz (Pavel Vachal), rao@lanl.gov (Rao V. Garimella), shashkov@lanl.gov (Mikhail J. Shashkov).

1 Introduction

Arbitrary Lagrangian-Eulerian or ALE methods are a popular class of methods for simulating continuum mechanics problems with large shear deformation such as fluid flow and metal forming [1,2]. ALE methods consist of a Lagrangian step in which the mesh nodes move according to the flow of the material, a rezone step in which the mesh is modified to improve its quality and the remapping step in which the solution is transferred from the old mesh to the new, improved mesh. In [3–5], methods were described to improve the quality of the mesh while keeping it close to the original mesh. However, in order to improve the meshes by the methods described in [3–5], all elements of the starting mesh must be valid or non-inverted. Therefore, if the Lagrangian step of an ALE simulation causes the mesh to become tangled (i.e., it has some elements that become inverted), the mesh must be untangled before the mesh improvement procedures are applied to it.

The need for untangling meshes also exists when a mesh generation procedure is unable to create all valid elements in a mesh. This situation may be encountered in the generation of all hexahedral meshes and general polyhedral meshes where there is no guaranteed method of directly generating a valid mesh [6]. It may also be encountered in advancing front based mesh generation of tetrahedral and hexahedral meshes [7–9] where it is possible to generate small cavities that cannot be filled with all positive volume elements. In these cases, it is very useful to have a tool that can untangle the mesh after the initial generation assuming that the right mesh connectivity has been generated.

Several researchers have recently begun focusing on the problem of untangling unstructured meshes by node repositioning [10–13]. Freitag and Plassmann [11,14] untangle meshes by optimization of a local function based on maximizing the minimum element area at each mesh vertex. Knupp [12] performs a global optimization of the difference between the absolute and signed values of element volumes in order to untangle the mesh. Kovalev et.al. [13] visit each vertex connected to at least one invalid element and reposition the vertex directly to a point in its feasible set (or "kernel") to make all connected elements valid. They define the feasible set of a vertex to be the set of all locations of the vertex for which all elements connected to the vertex will be valid. The new location of the vertex in the feasible set is found by a clever use of the simplex method to find three corners of the feasible set and taking their mean. Most of these procedures are capable of successfully untangling many tangled mesh configurations. However, there are conditions under which they may fail to untangle the mesh or they may not be able to produce elements with sufficient positive volume so as to provide a good starting point for mesh improvement procedures.

In this paper, a method is presented for untangling unstructured 2D meshes containing triangles and quadrilaterals. The method uses a multi-step approach based on a combination of relocating vertices to positions in their respective feasible sets and on minimizing a function to untangle the mesh elements. The untangling procedures only reposition nodes to untangle the mesh and do not perform any topological modifications since solution transfer between meshes with different connectivity is complex and expensive. Also in the interest of accurate solution transfer, the procedures endeavor to keep the mesh as close as possible to the original mesh.

The methods described here focus only on ensuring that all elements of the mesh meet at least bare minimum validity criteria as defined by the simulation procedures that will use them. However, the methods can easily be combined with separate procedures for improving the quality of the untangled mesh like the ones described in [4,5].

The rest of the paper is organized as follows. The next section discusses the definition of element validity and the definition of the feasible set for a vertex. Section 3 describes the untangling of a mesh by the feasible set method and untangling by optimization of an objective function is discussed in Section 4. Section 5 describes the multi-step untangling procedure that has achieved a greater success in untangling meshes than each of the two untangling methods by themselves. Finally, several simple and complex examples of mesh untangling and mesh optimization [4,5] are presented in Section 6 followed by conclusions.

2 Definitions

In this study, an element is considered valid if each corner of the element is considered valid. An element corner is valid if the Jacobian determinant of its mapping to a right angle corner with unit sides is positive. This requirement is more conservative than that required by some numerical methods [15] in which the element or its "subcells" must only have a decomposition into any one set of simplices (triangles and tetrahedra) with all positive volumes (See Figure 1). However, it does not explicitly account for the requirements of other methods such as having positive Jacobian determinants everywhere or at quadrature points.

Given the definition of element validity, the ***feasible set*** *for a vertex can be defined as the set of all positions of the vertex for which each connected element is valid at the corners affected by the position of the vertex.* In other words, if a vertex is in its feasible set, then each connected element is valid at the vertex and at the adjacent (edge-connected) neighbors of the vertex.

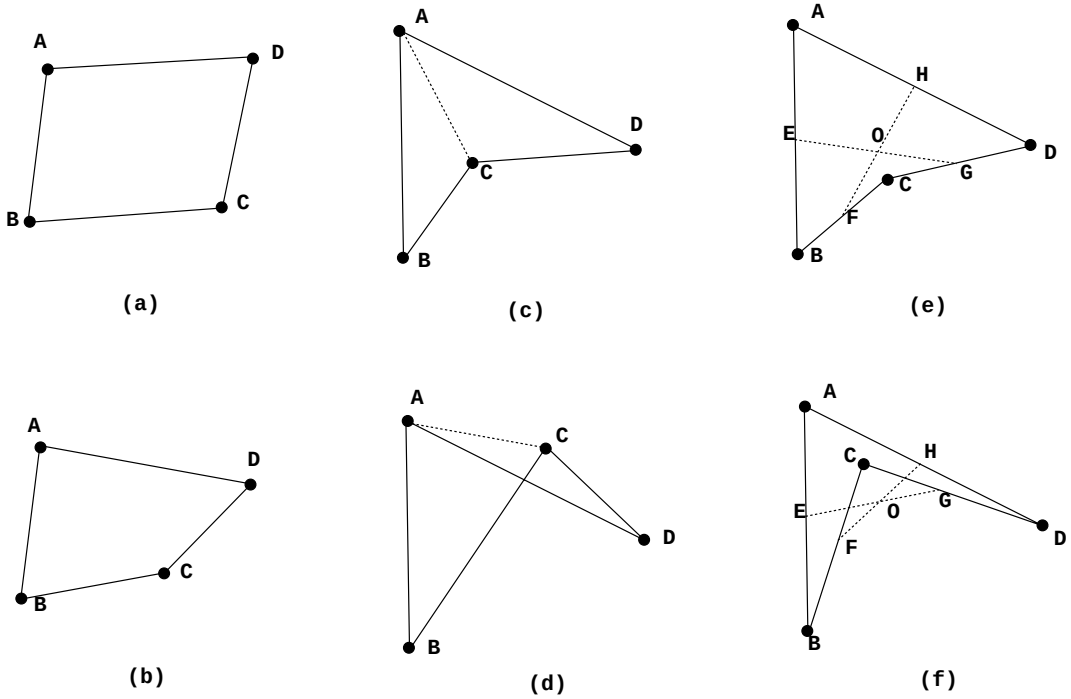


Fig. 1. Valid and Invalid Quadrilateral Elements: (a),(b) Valid quads with positive Jacobian everywhere (c) Quad with valid decomposition into positive area triangles ABC and ACD but negative Jacobian at reentrant corner, C (d) Quad with no valid decomposition into positive area triangles (e) Quad decomposition into valid subcells AEOH, BFOE, FCGO, GDHO but negative Jacobian at C (f) Quad with invalid subcells

The feasible set for a vertex connected to a single triangle is a half-space as shown in the 2D example in Figure 2a assuming a ccw ordering of vertices of the triangle. For a vertex connected to a single quadrilateral or general polygonal element, the feasible set of a vertex is defined by the intersection of three half-spaces as illustrated in Figure 2b. This is because the position of the vertex influences the Jacobian at the vertex V_1 and its two edge connected neighbors, V_2 and V_3 in the element. In the figure, if vertex V_1 moves past line AA' in the opposite direction of the arrows, the Jacobian at V_1 and the area of triangle $V_1V_2V_3$ become negative. If it moves past line BB', the Jacobian at V_2 and the area of triangle $V_2V_4V_1$ become negative and if it moves past line CC', the Jacobian at V_3 and area of triangle $V_3V_1V_5$ become negative. Figure 2c shows the feasible set for a vertex interior to a patch of elements. From the definition of the feasible set, it is clear that the feasible set is a convex polygon in 2D while it is a convex polyhedron in 3D. Note that the feasible set for a vertex can be empty depending on the geometry and configuration of the elements connected to the vertex. This implies that given the positions of the boundary vertices of the patch, there is no location for the vertex which will simultaneously make all the elements valid. Such an example is shown in Figure 3 along with configurations where the feasible set is a line and a point.

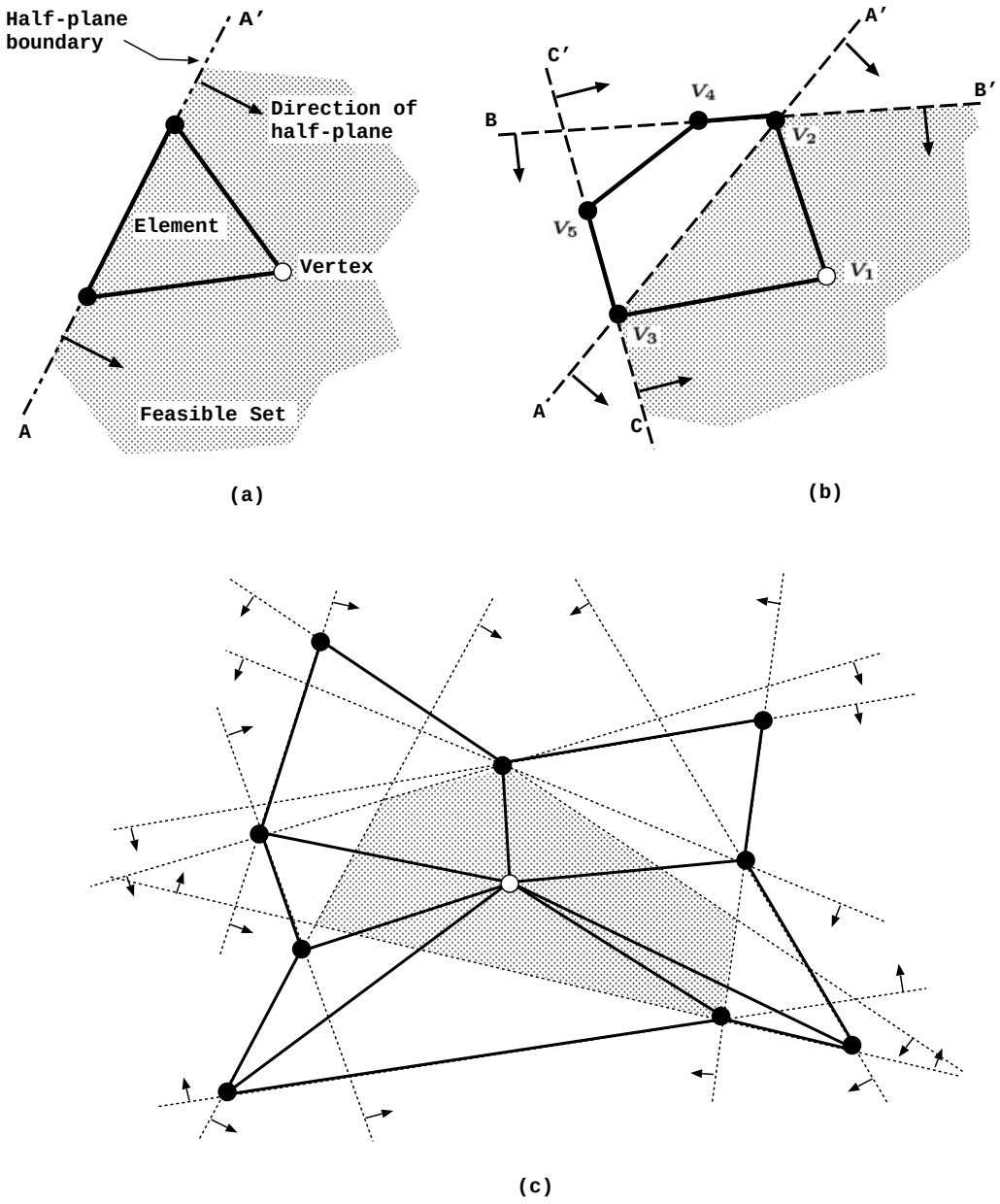


Fig. 2. Illustration of Feasible Sets (a) Feasible set for vertex connected to single element (b) Feasible set for vertex connected to a pentagon (c) Polygonal feasible set for vertex connected to patch of triangular and quadrilateral elements.

3 Untangling by Finding the Feasible Set

The definition of a feasible set given above leads to a natural method of untangling meshes in which the feasible set of a vertex is determined and the vertex is positioned inside it. This is referred to here as the *feasible set method*. In this approach each vertex of the mesh is visited and its connected elements examined to see if any of them are invalid. If an invalid element exists among

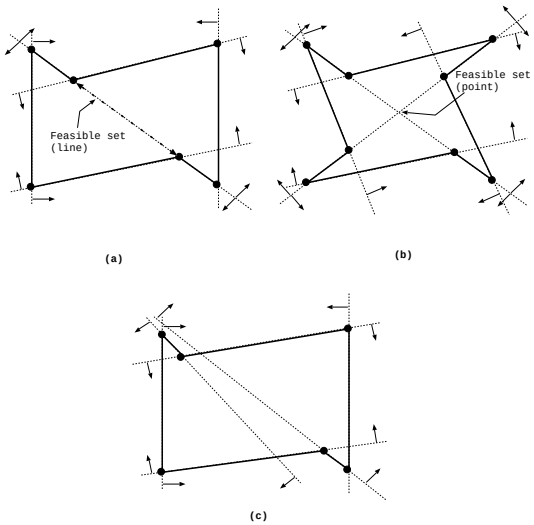


Fig. 3. Degenerate and Null Feasible Sets: (a) Feasible set is a line (b) Feasible set is a point (c) Null or empty feasible set.

the elements connected to the vertex, the feasible set of the vertex is computed and the vertex is placed in center of the feasible set. The method loops over the mesh until all elements are valid or no invalid element can be fixed.

The feasible set of a vertex can be found by computing the intersection of half-spaces representing the feasible region of the vertex with respect to each element. In 2D, this is accomplished by the intersection of pairs of lines demarcating the feasibility half-planes. In this work, the feasible set is found by the chasing algorithm described by O'Rourke [16] and has been found to work well on a number of test cases. In Figure 4, an example of mesh untangling by the intersection based feasible set method is illustrated step by step. In the figure, the invalid quadrilaterals are shown shaded and nodes connected to at least one invalid quadrilateral are shown in black.

While the intersection based computation of the feasible set works well in 2D, finding the feasible set polyhedron by intersection of half-spaces is impractically complex in 3D. Therefore, an alternate method for positioning the vertex inside the feasible set is implemented based on the simplex method as described in [13]. In [13] the idea is proposed that the boundary lines of the half-planes forming the feasible set in 2D can be interpreted as inequality constraints on the minimization of an arbitrary function. If the function is a linear function then its minimum must occur at the intersection of two of these inequality constraints or in other words the corner of the feasible set polygon. Therefore, the corners of the feasible set polygon can be found by minimizing different linear functions along with the appropriate inequality constraints using the simplex method [17,18].

This idea is further simplified by recognizing that for untangling, it is sufficient

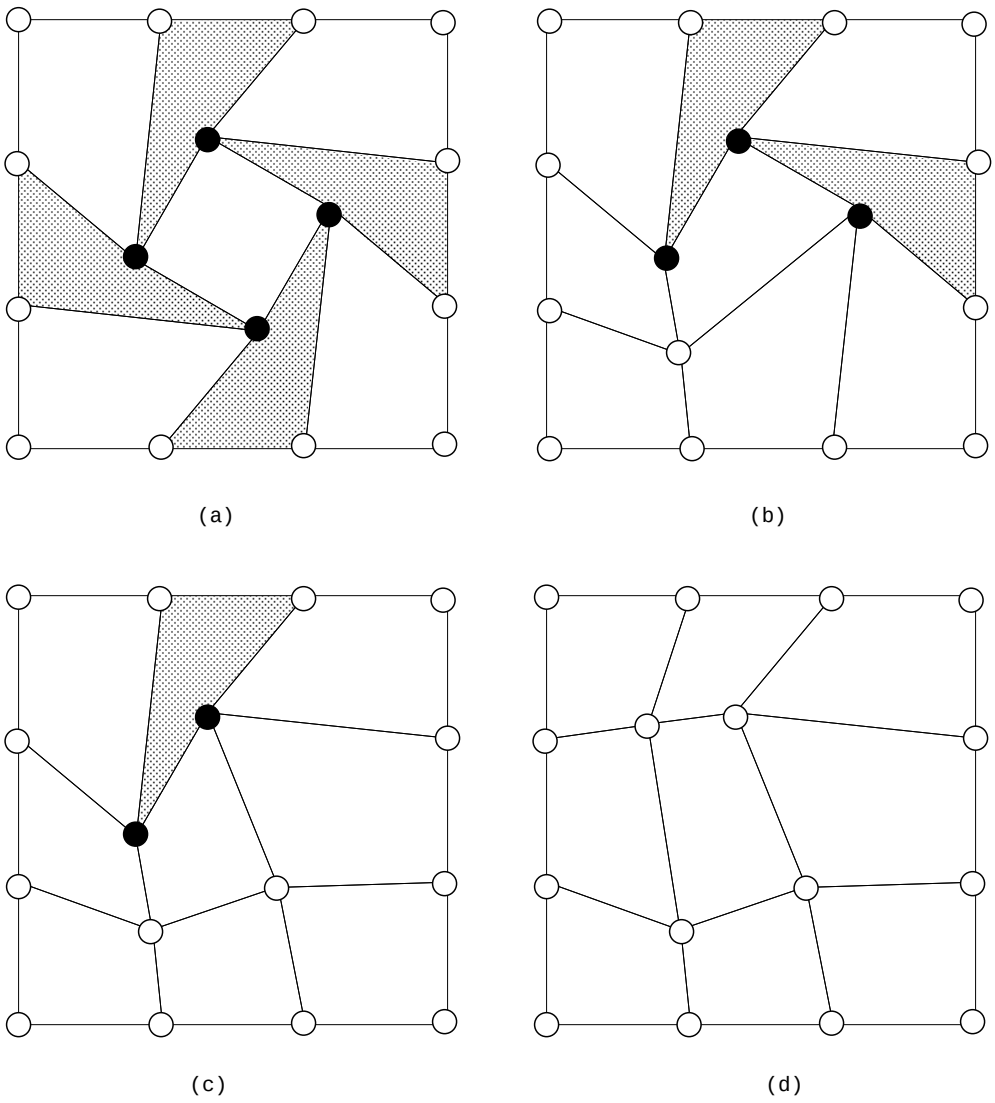


Fig. 4. Untangling of mesh by feasible set method with node placement at the center of the feasible set (a) Initial mesh (b),(c) Intermediate meshes (d) Final mesh. The shaded quadrilaterals indicate invalid elements and vertices represented in black are connected to at least one invalid quadrilateral.

to find any one position for the vertex inside its feasible set to make all connected elements valid. This implies that it is unnecessary to find all corners of the feasible set polygon. Therefore, for untangling a patch of elements in 2D, it can be inferred that it is sufficient to find three distinct corners of the feasible set polygon and reposition the vertex to the center of the triangle formed by these corners. Therefore, the procedure minimizes simple linear functions $f(x, y) = x, -x, y, -y$ to find three distinct corners of the feasible set polygon. The vertex is then repositioned to the center of the triangle formed by these corners. If only two distinct corners can be found, then the feasible set is a line connecting the two and the vertex is repositioned to the mid-point of the line. If only one distinct corner can be found, the feasible set is a point and

the vertex is repositioned to this point. Note that in the case when the feasible set is a line or a point, one or more elements connected to the vertex will be of zero area.

The feasible set approach to untangling is very useful because it is a direct way of fixing inverted elements affecting only those nodes connected to invalid elements. This characteristic can be useful for accurate solution transfer in ALE methods from the old mesh to the new, untangled mesh. Also, if the feasible set is not degenerate, fixing the mesh by this method results in elements that have substantial volume rather than near zero volume. However, the shortcoming of this approach is that some elements cannot be fixed because the feasible set associated with each of their vertices is empty.

4 Optimization Approach to Untangling

Several researchers have proposed the use of numerical optimization for untangling meshes. In this approach an appropriate objective function is minimized so as to make all the invalid elements valid. Freitag and Plassmann [11] attempt to untangle meshes by minimizing a local function at each mesh vertex. The local function at each vertex is taken to be the minimum volume of all connected elements. Knupp [12] proposed optimization of a global objective function based on the difference between signed volumes of elements and their corresponding absolute values. Thus, the objective function has a contribution from an element only if it has a negative area. If the area of the i'th element in a mesh is α_i, then the function to be minimized is

$$f(x) = \sum_{i}^{n} (\mid \alpha_i \mid -\alpha_i) \tag{1}$$

Minimization of this function can only bring elements to a zero area (volume) state which is still considered unusable in numerical simulations. Therefore, Knupp suggested the addition of a user controlled parameter β to the function, modifying it to be

$$f(x) = \sum_{i}^{n} (\mid \alpha_i - \beta \mid -(\alpha_i - \beta)) \tag{2}$$

The role of β is to force the function to reach a minimum when the elements have a small positive volume instead of zero volume. However, as discussed later, the proper selection of β can be tricky. The behavior of the objective

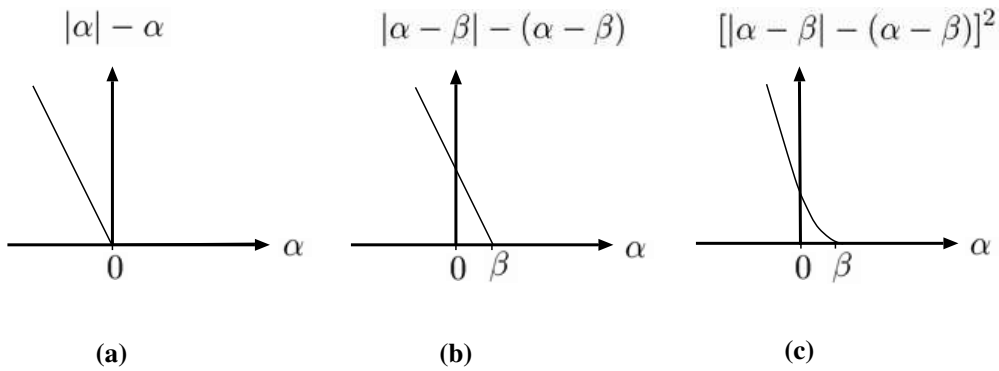


Fig. 5. Objective function behavior with respect to area/volume α of one element (a) Linear function without β (b) Linear function with β (c) Quadratic function with β

function for one element without and with β is shown in Figures 5a and b respectively.

The objective function in Eq. 2 is non-smooth due to the presence of an absolute value term in a linear form and minimization of such functions requires special and expensive methods of non-linear optimization. If a gradient based method is used, it must be modified to account for the fact that the gradient of the objective function is discontinuous. This can be done by interrupting a line search if valid elements become invalid or vice versa thereby causing a jump in the gradient. At this point the gradient can be reevaluated and the optimization continued.

To overcome the above optimization issues, the objective function proposed by Knupp in [12] has been modified in this work so that it is quadratic and smooth as shown for one element in Figure 5c. Since the objective function for one element is smooth and convex, it is expected that the global objective function formed by combining such functions for all elements of the mesh (shown in Eq. 3) is also smooth and convex.

$$f(x) = \sum_{i}^{n} \left(\mid \alpha_i - \beta \mid - (\alpha_i - \beta) \right)^2 \tag{3}$$

This smooth function can then be minimized using a standard numerical optimization method for smooth functions such as the conjugate gradient method [17,18]. It has been found in practice that minimization of the quadratic form of the objective function untangles the mesh more reliably than the linear form.

To illustrate the untangling of meshes by numerical optimization, some 1D cases of mesh untangling are discussed. In the first example consider a 1D

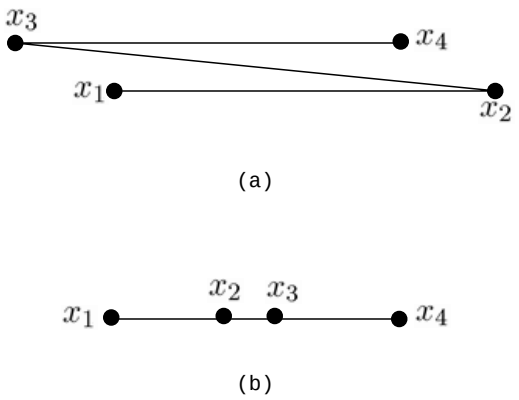


Fig. 6. (a) Tangled 1D mesh (b) One possible untangled mesh configuration

mesh with 4 points, x_1, x_2, x_3 and x_4, of which x_1 and x_4 are fixed boundary points, and x_2 and x_3 are interior points as shown in Figure 6. To make visualization of the mesh tangling easier, valid elements are shown with their nodes on a single line or level while an invalid element is shown with one of its nodes on the next higher level. Going from left to right, element 1 is valid ($x_1 < x_2$) and therefore, points x_1 and x_2 are shown on the same level. Element 2 is invalid and therefore, it is displayed as going from x_2 to x_3 at the next upper level. Element 3 is valid and so its vertices x_3 and x_4 are at the upper level. Note that the mesh shown in the figure cannot be fixed by the feasible set approach described earlier, since $x_3 < x_1$ making the feasible set of x_2 empty and the $x_2 > x_4$ making the feasible set of x_3 empty.

The objective function formed from this mesh is a function of two variables and can be readily visualized. Figures 7a and b show the linear and quadratic objective functions respectively for this mesh. The interior of the central triangle represents the portion of the x_2, x_3 space in which the mesh is valid, i.e., $x_i < x_{i+1}$ for all $i = 1, 4$. The paths taken by the optimization from different starting points on the figure (i.e. different tangled mesh configurations) are also shown in the figures. The mesh configuration shown in Figure 6 is indicated as point P_1 in Figures 7a,b. The optimization is performed with a finite but small value for β. It can be seen that the final valid configuration of the mesh depends on the starting invalid configuration of the mesh since the solution is non-unique. Also, the optimization procedure gets stuck while using the linear objective function (Eq. 2) due to the presence of discontinuities but successfully untangles the mesh using the quadratic function (see point P_2).

The second example illustrates the possible non-local effect of the optimization approach on the mesh. The example chosen is a 1D example with 11 points out of which one point (x_6) is well outside the domain as shown in Figure 8b. The placement of this point makes the element between x_6 and x_7 invalid. The locations of the mesh vertices and the various configurations of the mesh as the optimization procedure untangles the mesh are shown in Figure 8a,b.

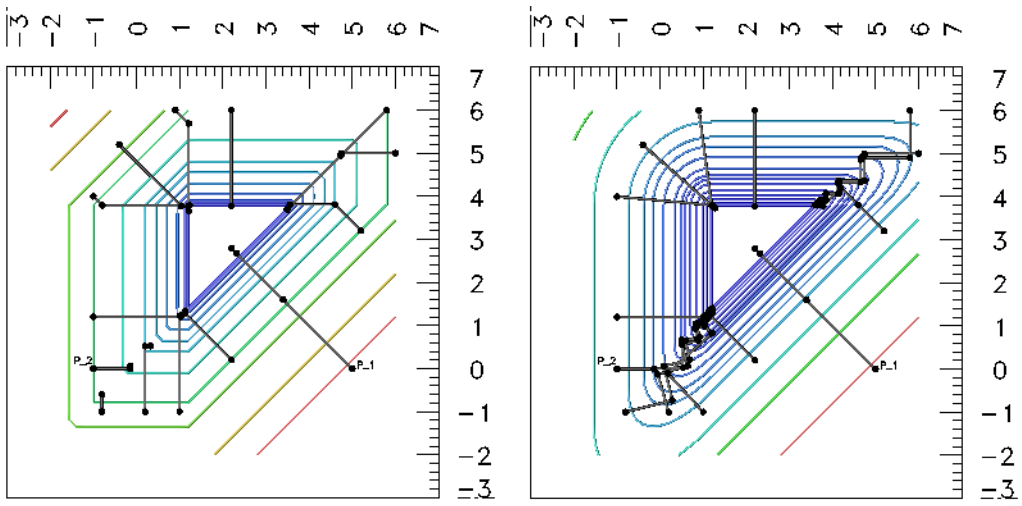
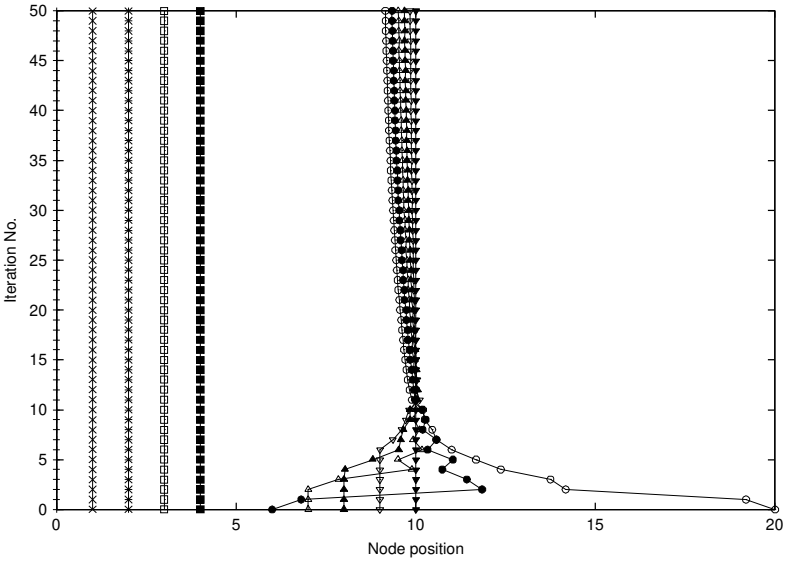


Fig. 7. Contour plots of objective functions for mesh untangling on which paths followed by the mesh points x_2 and x_3 from various starting configurations have been superimposed (a) Linear objective function $f(x) = \sum_i |\alpha_i - \beta| - (\alpha_i - \beta)$, $\beta = 0.1$ (b) Quadratic objective function $f(x) = \sum_i [|\alpha_i - \beta| - (\alpha_i - \beta)]^2$, $\beta = 0.1$.

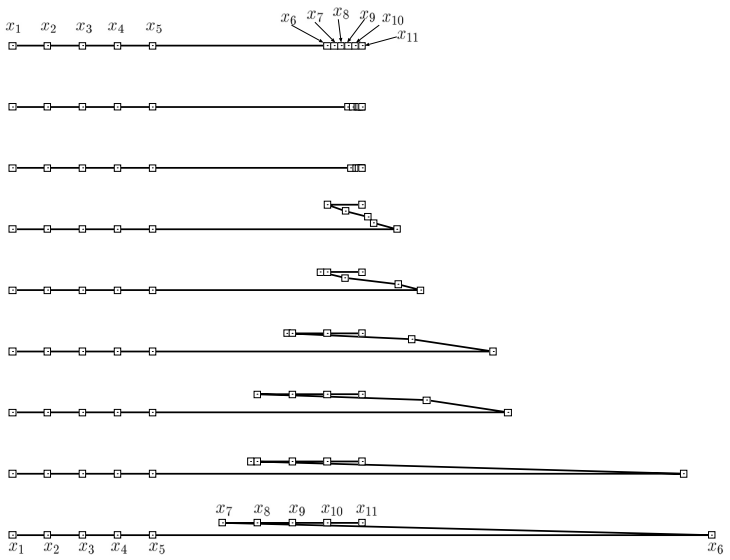
The optimization uses a β of 0.2 for the untangling. It can be seen that the optimization procedure moves nodes x_6, x_7, x_8, x_9 and x_{10} around to make the mesh untangled. In the process, several elements first become invalid and then return to a valid state. In contrast, the feasible set approach could have fixed this mesh by simply placing x_6 in between x_5 and x_7 in the initial mesh.

In practice, it is uncommon to encounter meshes entangled in such extreme ways and therefore, the optimization procedure usually only affects a small number of nodes in the local neighborhood. Still, the approach adopted here is to fix as much of the mesh as possible with the feasible set approach before using optimization to untangle meshes so that fewer nodes are perturbed in the process.

The optimization procedure for untangling 2D meshes is similar to the 1D meshes. For efficiency, the optimization procedure for 2D meshes is performed as a series of local iterations over the mesh vertices that minimize the global objective function instead of a simultaneous movement of all mesh nodes. It is important to note that at each local iteration at a vertex, all terms of the global function which are affected by the movement of the vertex are grouped into a local objective function which is minimized. For simplex elements, this just means that the area/volume of the elements connected to the vertex are involved in the local function. For more complex element types the local function includes the cross product (triple product in 3D) of the edge vectors in each connected element computed at the vertex itself and its edge connected neighbors. Computing the local objective function this way ensures that global objective function is still minimized at each local iteration.



(a)



(b)

Fig. 8. Optimization based untangling of 1D mesh with one node outside the domain. (a) Path followed by the mesh nodes as the optimization progresses (from bottom to top) (b) Mesh configuration at key stages in the optimization.

The optimization of the global function at interior vertices is preceded by a loop over the boundary vertices of the mesh with the purpose of untangling any invalid elements connected to the boundary. The objective function at any boundary vertex is evaluated as it is for an interior vertex. However, the optimization itself is performed with respect to a local parametrization of the mesh vertex with respect to the boundary mesh edges it is connected to [5]. This ensures that boundary mesh vertex remains on the original discrete boundary during its movement.

Figure 9 shows example of the mesh shown in Figure 4, untangled by the optimization procedure. Figure 9a shows the mesh optimized using the quadratic objective function without the use of β or $\beta = 0$. The procedure untangles the mesh but because β is zero, some of the valid elements are barely valid. The barely valid elements and the interior vertices connected to these elements are shaded in the figure. Figure 9b shows the same optimization but with a finite β, which is calculated as 10% of the problem size (diagonal of the bounding box) normalized by the number of elements in either the x or y directions and has the value of 0.106. As seen from the figure, the mesh is better in this case since all the elements are positively valid.

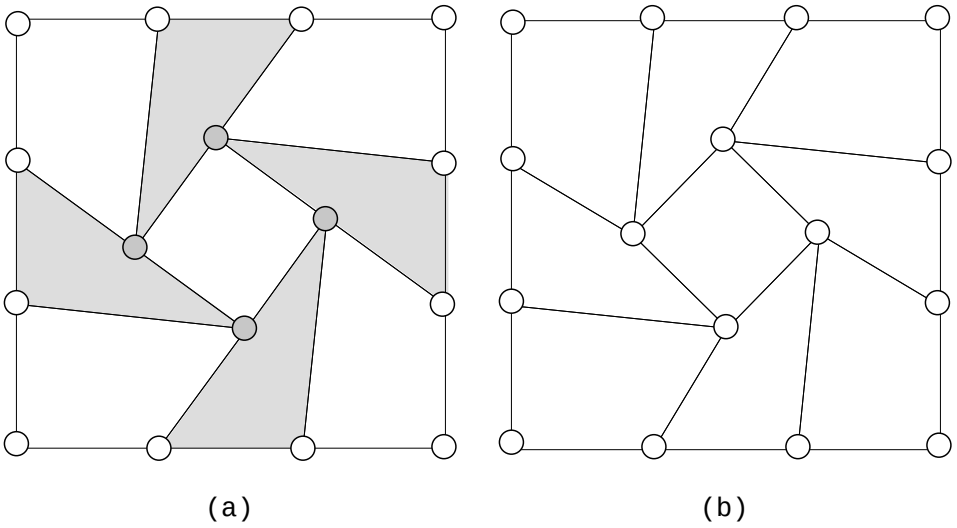


Fig. 9. Invalid mesh of Figure 4 optimized using optimization with quadratic objective function (a) Without β (barely valid elements shown shaded) (b) With $\beta = 0.106 = 10\%$ of bounding box diagonal normalized by number of elements in x or y directions

In using the optimization procedure for untangling meshes, the choice of β must be made carefully. Without β (i.e., $\beta = 0.0$), the optimization procedure can only make all the elements valid in the sense that the volume of every element is positive or at least zero. However, zero volume elements are unacceptable for many numerical procedures and improvement of meshes with zero volume elements can be difficult or impossible. Therefore, to force the untangled mesh to have all positive volume elements, the optimization procedure must use $\beta > 0$ in the objective function. On the other hand, increasing β indiscriminately can also be quite detrimental, since the objective function minimum can become non-zero and end up outside the feasible region for the patch. This lack of a solution implies that for the given boundary configuration it is not possible for all elements to achieve a volume equal to or greater than β. This undesirable effect of increasing β is illustrated in Figure 11 using the mesh patch shown in Figure 10. Therefore, the approach adopted here is to set β to the minimum acceptable area or volume of the elements with respect to the mesh optimization procedures that will subsequently improve

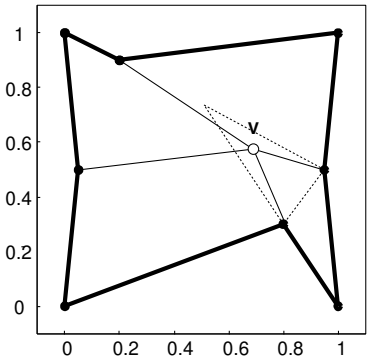


Fig. 10. Mesh patch used to illustrate the effect of increasing β on untangling by optimization. Dotted triangle shows feasible region for vertex V.

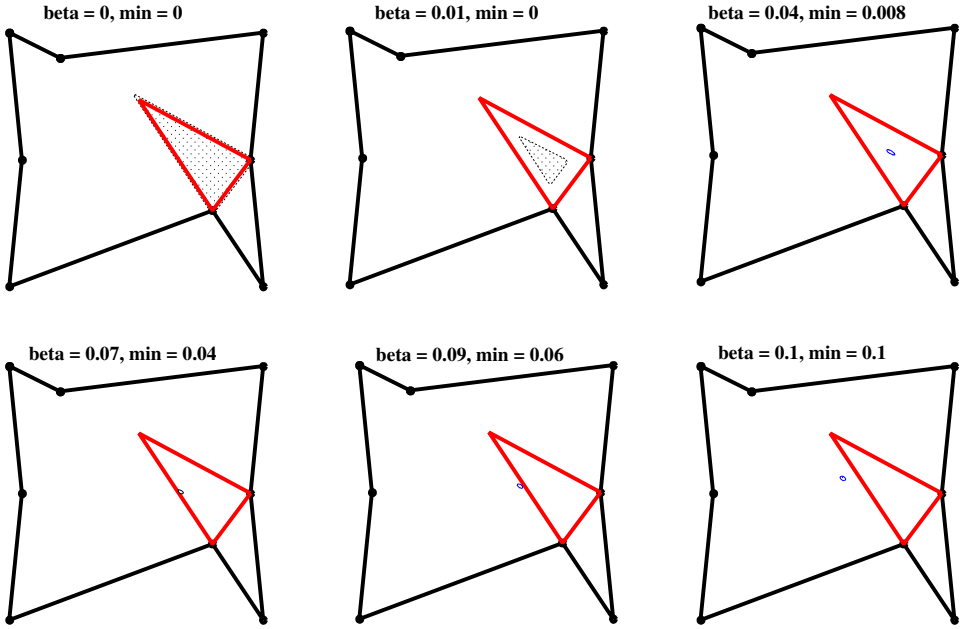


Fig. 11. Effect of increasing β on the minimum of the function $\sum_i[|\alpha_i-\beta|-(\alpha_i-\beta)]^2$ for a particular mesh patch. The solid triangle in the interior indicates the feasible set and the dotted region indicates the minimum of the function which can be outside the feasible set for a large β.

the mesh.

5 Three-step Procedure for Mesh Untangling

In the previous sections, it was seen that the feasible set approach had a local effect on the mesh but could not always fix the mesh. On the other hand, the optimization approach, usually fixed the mesh by making all elements non-negative but could affect a larger number of nodes and resulted in barely valid

elements. Therefore, the procedures have been combined here into a 3-step procedure for maximizing the possibility of untangling the mesh with minimal impact on the mesh.

The first step of the procedure performs untangling by the feasible set method so as to fix as many elements as possible with minimal impact to the valid part of the mesh.

The second step of the procedure performs a minimization of the quadratic objective function described earlier in order to fix any remaining invalid elements. The optimization procedure first performs a local or vertex-by-vertex optimization loop over the boundary vertices in order to try and fix as many elements as possible by their movement. The movement of the boundary vertices is constrained to the original discrete boundary by a local parameterization technique [5]. This is followed by local optimization loop over all interior vertices connected to invalid elements.

The third step of the procedure performs another round of untangling on barely valid elements by the feasible set approach to increase their volume. This step requires redefinition of the feasible set boundaries to account for the desired element volumes. Once this is done, the mesh is usually suitable for use as input to a mesh quality improvement procedure such as the one described in [4,5].

The effect of the 3-step procedure is illustrated first with a 1D example as shown in Figure 12. In the first step, the feasible set approach is able to fix element x_4x_5 by moving x_5 to the middle of x_4 and x_6. However, it cannot fix element x_8x_9 because the feasible set for both x_8 and x_9 is empty. In the second step, the optimization method fixes all the elements but nodes x_8,x_9 and x_{10} are too close to each other. In the third step, the feasible set method achieves a better distribution of the nodes by placing nodes connected to the barely valid elements, x_8x_9 and x_9x_{10} in the middle of their respective feasible sets.

6 Results

The first example shown in Figure 13 illustrates the untangling of a mesh arising from a Rayleigh-Taylor simulation. The mesh is made invalid during the Lagrangian step of an ALE simulation of the problem and must be fixed before the simulation can proceed. Figure 13a shows a part of the overall domain, Figure 13b shows the tangled portion of the mesh (with squares marking nodes connected to invalid elements). Figure 13c shows the untangled mesh corresponding to the tangled meshes shown in Figures 13a,b. The stages of

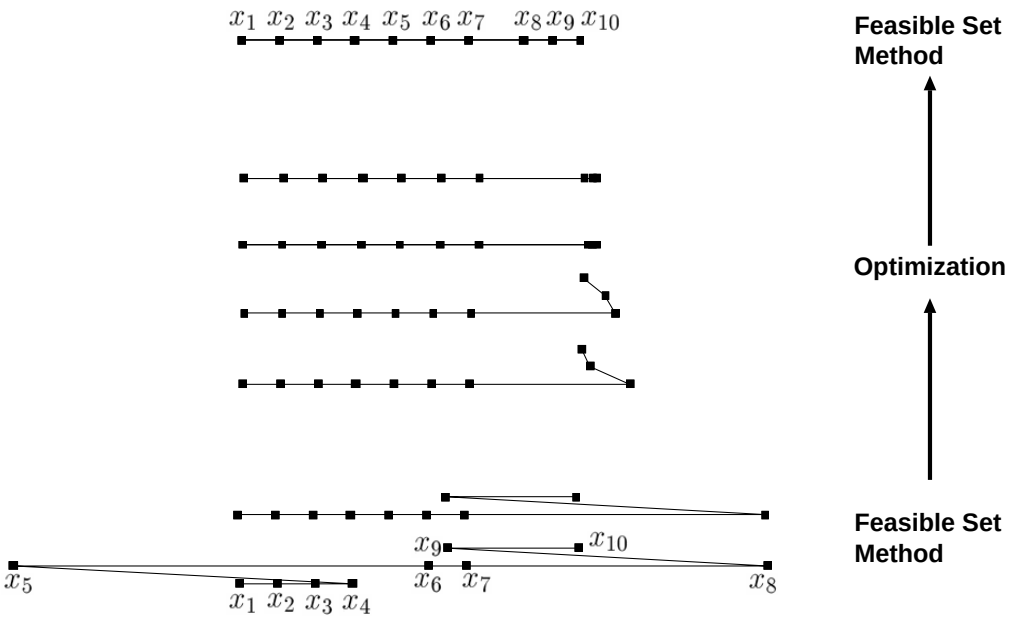
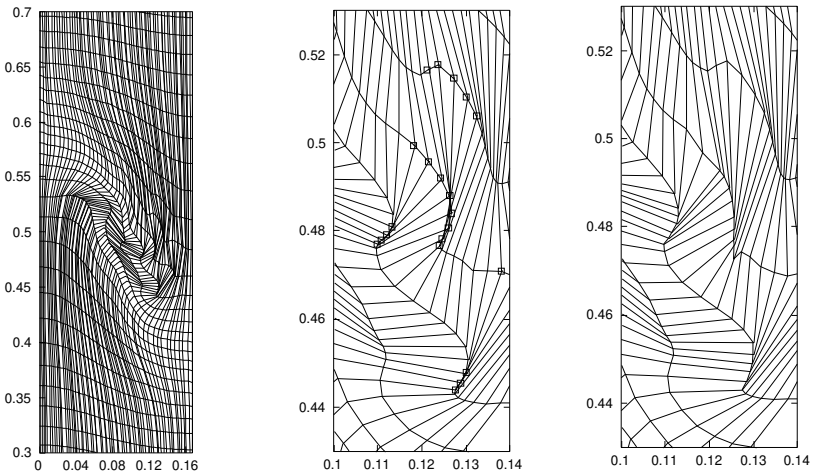


Fig. 12. Three step procedure involving feasible set method and optimization for fixing 1D mesh

untangling of this mesh are illustrated further in Figure 14. Figure 14a shows a zoom-in of the tangled portion of the original mesh. Figure 14b shows the mesh after the first untangling step by the feasible set method which is unable to fix all the elements. Figure 14c shows the mesh after untangling by optimization during which all elements were made at least barely valid. Figure 14d shows the mesh after the second round of untangling by the feasible set method during which all elements were brought to a positive volume state. Finally, Figure 14e shows the mesh after improvement by the Reference Jacobian Matrix based optimization procedure [4,5]. This type of mesh optimization procedure, developed for ALE simulations, improves mesh quality while keeping the mesh close to the original configuration as can be seen by comparing Figures 14d,e.

The next example shows the untangling of a structured mesh, called the "horseshoe" mesh shown in Figure 15b obtained by simple Laplacian smoothing of the mesh shown in Figure 15a [19]. The untangled mesh after the 3-stage untangling procedure is shown in Figure 15c. Finally, Figure 15d shows the untangled mesh improved by the condition number based optimization procedure [4,5] which improves the condition numbers of all the elements as much as possible.

Figure 16 presents the example of untangling a mesh of the state of Texas [20]. An originally valid mesh shown in Figure 16a was tangled by a random perturbation of a subset of the interior vertices to result in the mesh shown in Figure 16b. The maximum perturbation was 20% of the domain size. This mesh was then successfully untangled using the 3-step procedure to give the



(a) (b) (c)

Fig. 13. Untangling of Lagrangian mesh from Rayleigh-Taylor Simulation (a) Part of original tangled mesh (b) Zoom-in of tangled mesh (nodes shown by a square are connected to at least one invalid element) (c) Zoom-in of untangled mesh

mesh shown in Figure 16c. For this example, the feasible set method alone left behind 17 patches with negative area triangles and the optimization method alone left behind 6 zero area triangles. The value of β was zero for the optimization step. The result of condition number based mesh optimization on the untangled mesh is presented in Figure 16d to illustrate that good quality meshes can be obtained when the untangling procedure is combined with mesh improvement procedures.

Finally, a untangling of a tetrahedral mesh is presented in Figure 17 to illustrate that the principles presented in this paper can be extended easily to 3D. The original 3D mesh is shown in Figure 17a. The tangled 3D mesh obtained by perturbing the interior nodes is shown in Figure 17b and the untangled mesh is shown in Figure 17c. The minimum and maximum element volumes before untangling were -0.194 and 0.475. The minimum and maximum volumes after the mesh was untangled were 1.39E-07 and 0.242. Therefore, the untangled mesh is suitable for input to a mesh optimization procedure for improving mesh quality.

7 Conclusions

A multi-step method for successful untangling of unstructured 2D meshes has been presented in this paper. The method uses a combination of the feasible set method and optimization method to achieve the greatest degree of success in untangling the mesh while keeping the mesh close to the original mesh as

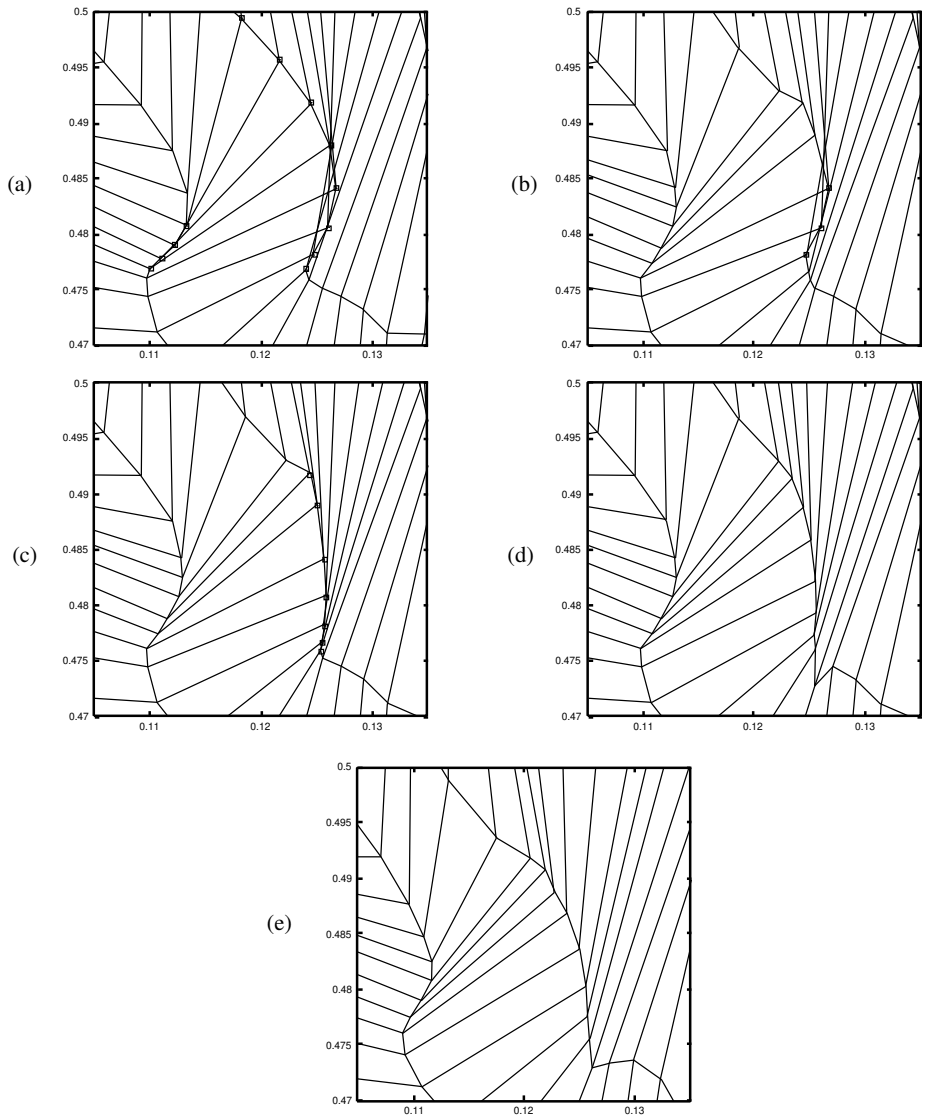


Fig. 14. Steps of untangling Rayleigh-Taylor simulation mesh (a) Zoom-in of original mesh (nodes connected to invalid elements shown as squares) (b) Mesh after feasible set approach, some elements remain with less than a minimum volume (c) Mesh after optimization, some elements remain with zero volume (d) Mesh after second pass of the feasible set approach, all elements have volume greater than required minimum (e) Mesh improved with Reference Jacobian Matrix based optimization procedure [4,5] which seeks to improve the mesh but keep it close to the original mesh

required for remapping in ALE simulations. The methods have shown a high degree of success in untangling complex 2D meshes. The formulation of the procedures allows easy extension to 3D problems and preliminary results are promising. Work is in progress to apply the untangling procedure for general polygonal meshes.

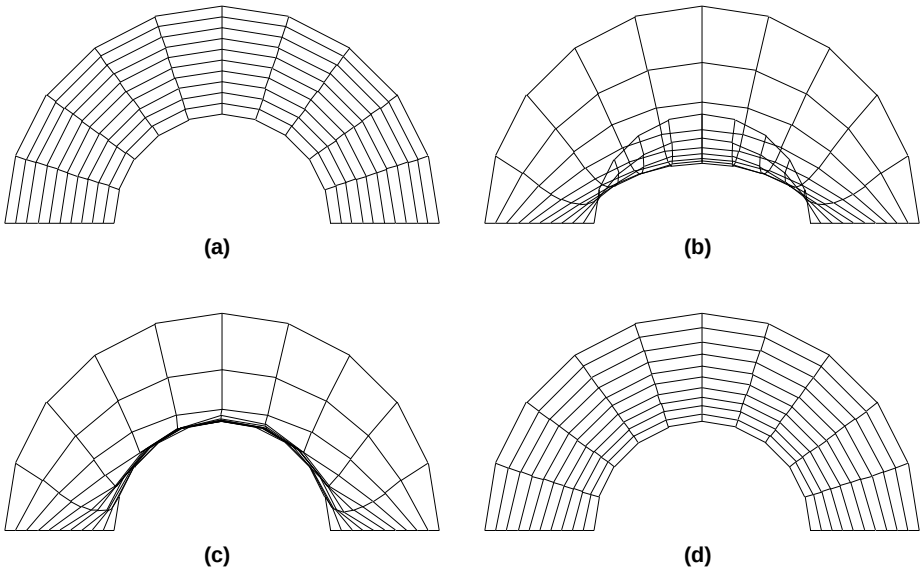


Fig. 15. (a) Valid mesh of horseshoe (b) Mesh tangled by simple Laplacian smoothing (c) Mesh untangled by 3-step untangling procedure (d) Mesh after improvement by condition number based optimization [4,5] which improves the condition numbers of all elements as much as possible

8 Acknowledgments

The work of the authors was performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

The work of P. Vachal was also supported in part by the Czech Grant Agency grant 201/00/0586.

The authors wish to thank Dr. Richard Liska (Czech Technical University in Prague, Czech Republic) and Dr. Markus Berndt (Los Alamos National Laboratory, Los Alamos, NM, USA) for implementation of the feasible set computation by geometric intersections. The authors also thank Dr. Patrick Knupp (Sandia National Laboratories) for valuable input during the development of these procedures.

References

[1] C.W. Hirt, A.A. Amsden, and J.L. Cook. An Arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*,

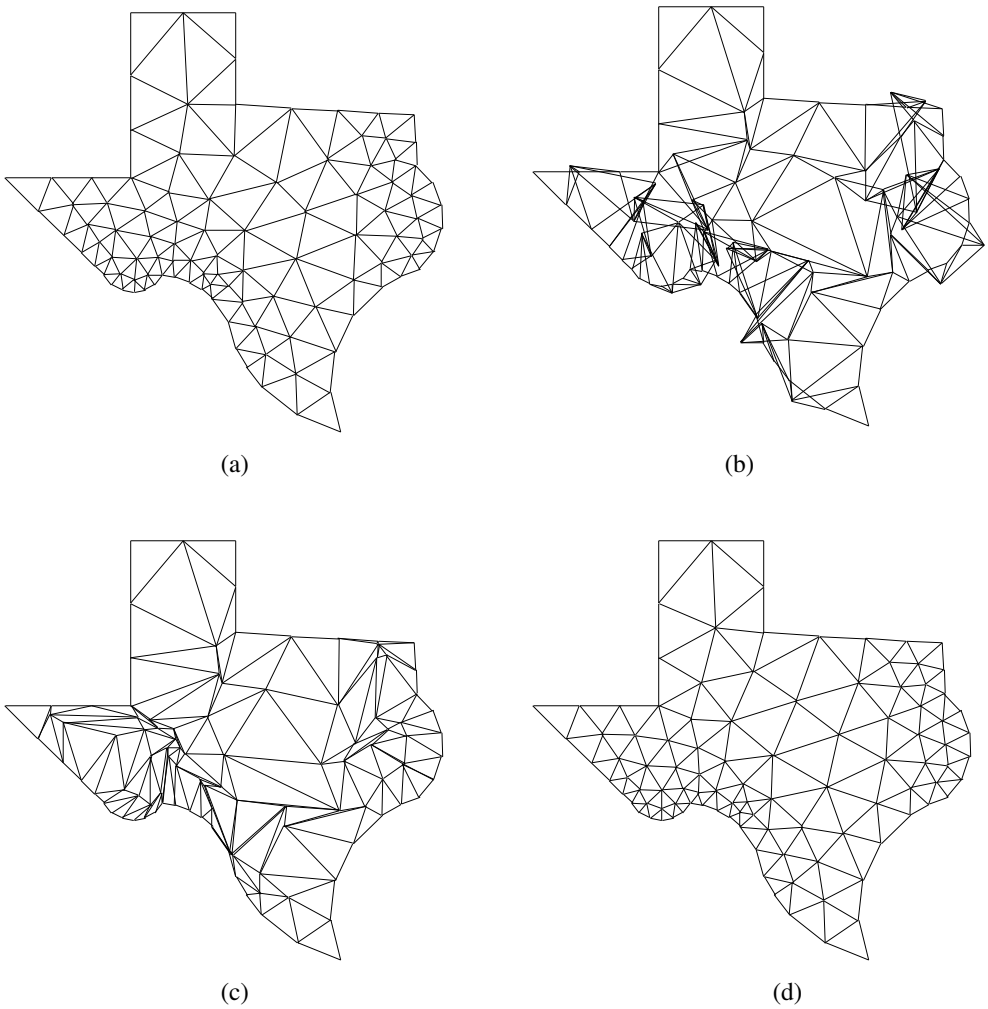


Fig. 16. Untangling of triangular mesh of Texas outline (a) Original valid mesh (b) Tangled Mesh (random movement of interior vertices) (c) Untangled mesh with poor quality elements (d) Improved mesh obtained by condition number based optimization.

135(2):203–216, 1997. Originally published in Journal of Computational Physics, 1974.

[2] D.J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:235–395, 1992.

[3] P. M. Knupp, L. G. Margolin, and M. J. Shashkov. Reference Jacobian optimization-based rezone strategies for Arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176:93–128, 2002.

[4] M. J. Shashkov and P. M. Knupp. Optimization-based reference-matrix rezone strategies for Arbitrary Lagrangian-Eulerian methods on unstructured grids. In *Proceedings of the Tenth Anniversary International Meshing Roundtable*, pages 167–176, Newport Beach, CA, October 2001. `http://cnls.lanl.gov/~shashkov`.

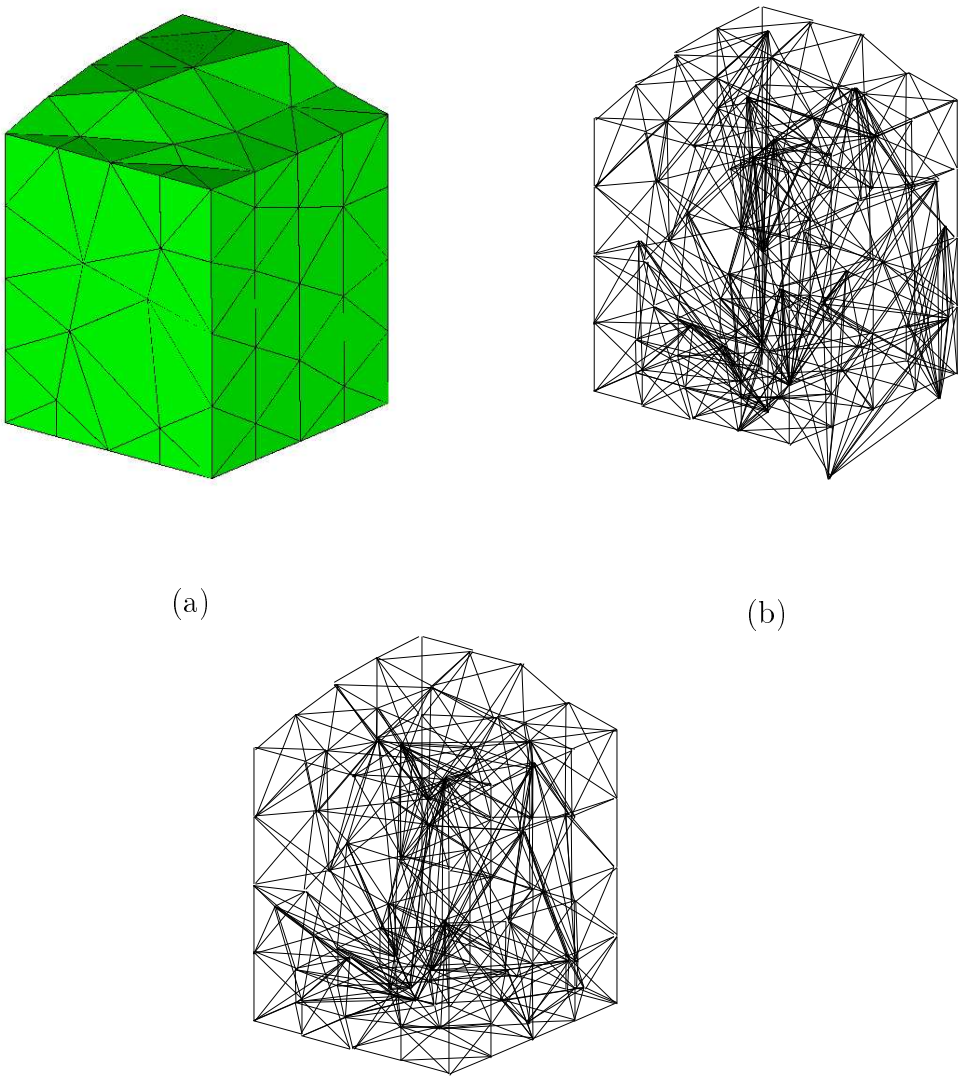


Fig. 17. (a) Original tetrahedral mesh (b) Tangled mesh (Minimum element volume -0.194) (c) Untangled mesh (Minimum element volume 1.39E-07)

[5] R. V. Garimella, M. J. Shashkov, and P. M. Knupp. Optimization of surface mesh quality using local parameterization. In *Proceedings of the Eleventh International Meshing Roundtable*, pages 41–52, Ithaca, NY, September 2002. `http://cnls.lanl.gov/~shashkov`.

[6] T.J. Tautges, T. Blacker, and S.A. Mitchell. The Whisker Weaving Algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes. *International Journal of Numerical Methods in Engineering*, 39(19):3327–3349, 1996.

[7] T. D. Blacker and M. B. Stephenson. Seams and wedges in plastering: A 3D hexahedral mesh generation algorithm. *Engineering with Computers*, 9:83–93,

1993.

[8] R. Löhner and P. Parikh. Three-dimensional grid generation by the advancing front method. *International Journal for Numerical Methods in Engineering*, 8:1135–1149, 1988.

[9] B.K. Karamete, M.W. Beall, and M.S. Shephard. Triangulation of arbitrary polyhedra to support automatic mesh generators. *Internation Journal for Numerical Methods in Engineering*, 49(1):167–191, 2000.

[10] T.S. Li, Y.C. Wong, S.M. amd Hon, C.G. Armstrong, and R.M. McKeag. Smoothing by optimisation for a quadrilateral mesh with invalid elements. *Finite Elements in Analysis and Design*, 34:37–60, 2000.

[11] L. Freitag and P. Plassmann. Local optimization-based simplicial mesh untangling and improvement. *International Journal of Numerical Methods for Engineering*, (49):109–125, 2000.

[12] P.M. Knupp. Hexahedral and tetrahedral mesh untangling. *Engineering with Computers*, 17:261–268, 2001.

[13] K. Kovalev, M. Delanaye, and Ch. Hirsch. Untangling and optimization of unstructured hexahedral meshes. In S.A. Ivanenko and V.A. Garanzha, editors, *Proceedings of Workshop on Grid Generation: Theory and Applications*, Moscow, Russia, 2002. Russian Academy of Sciences. `http://www.ccas.ru/gridgen/ggta02/papers/Kovalev.pdf`.

[14] L. Freitag and P. Plassman. Local optimization-based untangling algorithms for quadrilateral meshes. In *Proceedings of the Tenth Anniversary International Meshing Roundtable*, pages 397–406, Newport Beach, CA, October 7-10 2001. `http://www.imr.sandia.gov/papers/imr10/freitag01.ps.gz`.

[15] E.J. Caraman, D.E. Burton, M.J. Shashkov, and P.P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics*, 146:227–262, 1998.

[16] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, Cambridge, UK, 1996.

[17] Reklaitis, Ravindran, and Ragsdell. *Engineering Optimization - Methods and Applications.* Wiley Interscience, 1983.

[18] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer, 1999.

[19] P.M. Knupp and S. Steinberg. *The Fundamentals of Grid Generation.* CRC Press, 1993.

[20] R. E. Bank. *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide 8.0.* Society for Industrial and Applied Mathematics, Philadelphia, 1998.