Selçuk J. Appl. Math.
Vol. 3, No. 1, pp. 81–99, 2002

Selçuk Journal of Applied Mathematics

Optimization-based reference-matrix rezone strategies for arbitrary Lagrangian-Eulerian methods on unstructured meshes

Mikhail Shashkov[1] **and Patrick Knupp**[2]

[1] Group T-7, Los Alamos National Laboratory, Los Alamos, NM 87544, USA; e-mail: `misha@t7.lanl.gov`
[2] Sandia National Laboratory, New Mexico PO Box 5800, Albuquerque, NM 87185, USA; e-mail: `pknupp@sandia.gov`

Received: May 6, 2002

Summary. The objective of the Arbitrary Lagrangian-Eulerian (ALE) methodology for solving multidimensional fluid flow problems is to move the computational mesh, using the flow as a guide, to improve the robustness, accuracy and efficiency of the simulation. A principal element is the rezone phase in which the computational ("rezoned") mesh is created that is adapted to the fluid motion. Here we describe a general rezone strategy that ensures the geometric quality of the computational mesh, while keeping it as close as possible to the Lagrangian mesh. In terminology of mesh generation community our method can be classified as a two-stage smoothing algorithm. We provide numerical examples to demonstrate the robustness and the effectiveness of our methodology.

Key words: ALE, rezoning, optimization, reference Jacobian, condition number

Mathematics Subject Classification (1991): 49K35, 65K10, 76M25

1. Introduction

In numerical simulations of multidimensional fluid flow, the relationship of the motion of the computational mesh to the motion of the fluid is an important issue. There are two choices that are typically made, one representing a Lagrangian framework and the other an Eulerian framework. In the Lagrangian framework the mesh moves with the local fluid velocity, while in the Eulerian framework the fluid flows through a mesh fixed in space. In the arbitrary Lagrangian-Eulerian (ALE) methods [1–4] the motion of the computational mesh is used as independent degree of freedom to optimize accuracy, robustness, and computational efficiency. The main elements in an ALE simulation are an explicit Lagrangian phase, a rezone phase in which the computational mesh is redefined, and a remapping in which the Lagrange solution is transferred to the rezoned mesh. Of these three processes, it is the rezone phase that is least developed.

The purpose of this paper is to describe a general rezone strategy that ensures the continuing geometric quality of the computational mesh while keeping the rezoned mesh at each time step as close as possible to the Lagrangian. Geometric quality includes such mesh features such as smoothness, invertibility, and cell size, shape, and orientation. The latter three properties are derived from the Lagrangian mesh; size and orientation are mostly preserved while the shape is improved to remove mesh distortions which can result in loss of accuracy or non-invertible cells. By keeping the rezoned mesh 'close' to the Lagrangian mesh, it is believed that solution adaptivity is preserved while level of accuracy is increased. To further enhance solution adaptivity, we intend in future to incorporate error estimates based on solution field (e.g. velocity or acceleration) to the mesh optimization strategy.

The essential new idea in this paper is the recognition that *the Lagrangian solution before rezoning contains sufficient information about the flow to constrain our measure of the smoothness of the mesh.* More specifically, the Lagrangian mesh reflects both the physical motion of the fluid and unphysical distortion. We assume that the unphysical distortion of a computational mesh has a much shorter wavelength, and so can be separated from the physical motion by averaging over a small neighborhood of the cell. This assumption naturally leads to a desire that rezoned mesh be close to the Lagrangian mesh but "smoother" (i. e., better geometrical quality). By requiring the rezoned mesh to remain as close as possible to the Lagrangian mesh, we will minimize the error of the remap phase. Our preliminary results for ALE method using logically-rectangular meshes [5] look very promising. In this paper, we describe our methodology for unstructured meshes.

The general unstructured mesh is defined by positions of the nodes and connectivity. Between the node and each of its neighbors is an "edge-vector". Using connectivity one can use edge-vectors to form the columns of matrices (Jacobians). Valid mesh is uniquely defined by these Jacobian matrices. If two meshes with same connectivity have close set of Jacobian matrices the positions of corresponding nodes are also close. This leads to the following strategy. We first construct Jacobian matrices — termed the reference Jacobian matrices (RJM's) — that are based on the geometry of the nearest neighbors of a Lagrangian cell, effectively smoothing the shorter wavelength mesh deformation . It is not possible to use the RJMs defined at each vertex by itself to derive the new (i. e., rezoned) positions of the vertices because the RJMs related to the cell specifies the vectors that form the edges of a cell and there is no guarantee that these vectors will form a closed figure. Furthermore the cells do not exist in isolation from each other. Each vertex belongs to many cells; a simple rezone strategy applied individually to each cell will lead, in general, to incompatible specifications of the rezoned position of any vertex. *Exactly for this reason one cannot simply use just a local optimization procedure for constructing global rezoned mesh.* The resolution is to construct a global functional that measures the difference between the RJM (which depends on the Lagrangian mesh and the smoothing process) and the rezoned Jacobian (not known before the minimization process) of all cells in the mesh. Minimizing this functional over the entire mesh (as a function of each of the vertex coordinates) then leads to the rezoned mesh. This strategy still does not guarantee that the rezoned mesh is unfolded. However in our framework, this property now can be easily enforced through a simple modification of our global functional to include a barrier function that effectively penalizes any solution where the area of the cell is much smaller than the area derived from the RJM.

The flows that we are interested in are usually multimaterial, and it is important to maintain interfaces between different materials during ALE calculations. This means that nodes that are on material interface have to stay on this interface during optimization. This requirement leads to a constrained optimization problem, in which there are less degrees of freedom for nodes on interfaces. This constraint has to be taken into account in the construction of RJMs as well as in the global optimization stage.

To summarize, our method consists of two components: a sequence of local optimizations followed by a single global optimization (see flow-chart in Fig. 1). The local optimization — stage I in Fig 1 — defines "reference" Jacobians (J_{ref}), that incorporate our definition of mesh quality at each point of the mesh. The "rezoned" mesh results from minimizing a global objective function that measures the distance (in a least-squares sense) between the Jacobians of the rezoned mesh and the reference Jacobians — stage II in Fig. 1. The global optimization resolves incompatibilities of the locally defined RJMs in a smooth manner. We use Lagrangian mesh as initial guess for global optimization. Because the Lagrangian mesh is usually close to the minimizing mesh, the global minimization process usually converges quickly. The Lagrangian mesh is assumed to be invertible.

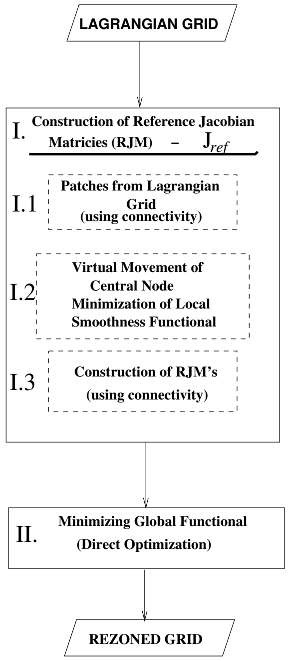


Fig. 1. Flow chart.

The rest of the paper is organized as follows. In section 2 we describe how to construct RJM for given Lagrangian mesh. In section, 3, we describe procedure to construct rezoned mesh for given set of RJMs. The description includes discussion of the global objective function, optimization procedure, special case of constrained optimization, and numerical examples. Finally, in the conclusion (section 4) we summarize results and describe future work.

2. Construction of reference Jacobian matrix (RJM) from Lagrangian mesh

As asserted above, the Jacobian matrix is the natural object with which to analyze and control the first-order properties of the mesh. In

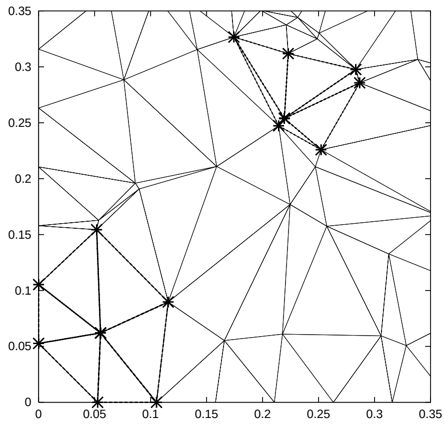


Fig. 2. The patches on the triangular mesh. Only the fragment of the triangular mesh in the unit square is shown.

particular, to produce a smooth mesh that is close to the Lagrangian mesh, we will begin with the Jacobians that correspond to Lagrangian mesh and modify them to form the RJMs. These modifications will be performed locally. For example, on the 2D triangular mesh in Fig. 2, we consider a central mesh point and all nodes which are connected to central node by edge-vectors. This procedure defines patch consisting of all triangles which have central node as a vertex, — stage I.1 in Fig. 1. Two such patches are highlighted in Fig. 2 Suppose that we have some method of choosing a better position for this central node, while keeping all the other mesh points of the patch fixed — stage I.2 in Fig. 1. Virtual displacement leads to new edge vectors (reference edge vectors) and hence a new Jacobian for each of the triangles in the patch — stage I.3 in Fig. 1. These new Jacobians are the RJMs at this mesh point. In this paper we will describe our strategy for virtual displacement of central node in patch based on a theory of algebraic mesh quality [6, 7]. If we denote central node in the patch by C, and enumerate the nodes on the boundary of the patch in counter-clockwise order then typical triangle in this patch looks like the one shown in Fig. 3. We denote edge vectors connecting central node with kth node on the boundary of the patch by $\mathbf{e}_k$, its length is denoted by l_k. The area of triangle with vertices C, k, $k+1$ is denoted by $A_{k+\frac{1}{2}}$. Accordingly to [6, 7] we will form Jacobian matrix $J_{k+\frac{1}{2}} = (\mathbf{e}_k|\mathbf{e}_{k+1})$, and will use condition number, $\kappa(J_{k+\frac{1}{2}})$, of this matrix as measure quality of corresponding triangle from point of view of central node.

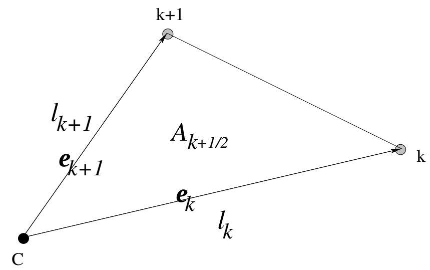


Fig. 3. Typical Triangle in Patch

Simple algebra shows that

$$\kappa\left(J_{k+\frac{1}{2}}\right) = \frac{l_k^2 + l_{k+1}^2}{A_{k+\frac{1}{2}}}.$$

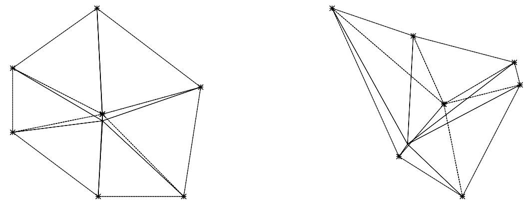
The optimal virtual position of central node, (x_c, y_c) of the patch is found by minimizing following functional

$$F_c(x_c, y_c) = \frac{1}{2}\sum_k \kappa\left(J_{k+\frac{1}{2}}\right).$$

To minimize this functional we use a direct optimization procedure described in detail in section 3.2. Optimization starts with Lagrangian mesh (which is assumed to be valid mesh — all areas of triangles are positive) as initial guess. It is important to note that the virtual movement of the central point always leads to RJMs with positive determinant (areas of triangles), because these areas are in the denominator of each term in functional and effectively play the role of barriers. Fig. 4 shows the original local patches marked in Fig. 2 and corresponding virtual displacement of central point. In Fig. 4 a) we present "smooth" patch. In this case optimal virtual position is almost the same as original position of central node. In case of "non-smooth" patch, shown in Fig. 4 b) optimal virtual position differs significantly from original Lagrangian position. These results show that the local optimization produces RJMs which are significantly different from Jacobian matrices of Lagrangian mesh only in regions where its geometrical quality is low.

3. Construction of the mesh for given set of reference Jacobians

In this section we describe the construction of the global objective function and our optimization procedure in more detail — stage II



a) b)

Fig. 4. Local virtual movement of central node of the patch: a) smooth patch, b) non-smooth patch.

in Fig. 1. Assume that we are given reference Jacobian matrices J_{ref} with positive determinant. Our goal is to construct a rezoned mesh whose set of Jacobian matrices, J, are close to the set of J_{ref} and which also has positive determinant everywhere (so that the rezoned mesh is unfolded). The global objective function is based on the idea presented in [8] that if one performs a least-squares fit between the Jacobian matrices (or their inverses) and the reference Jacobian matrices, one can control basic mesh properties such as smoothness, cell shape, size, and orientation.

3.1. Global objective function

The RJMs described in the previous section can be conveniently denoted using two indices: subscript index, n, for node, and superscript index, t for triangle.

The global objective function, whose minimization yields the rezoned mesh with set of nodes (x_n, y_n) is

$$F(\ldots, x_n, y_n, \ldots) = \sum_n \left(\sum_{t \in St(n)} \frac{\| J_n^t - \left(J_{ref}\right)_n^t \|_F^2}{|J_n^t| / |\left(J_{ref}\right)_n^t|} \right).$$

Here $St(n)$ is set of triangles which share node n as a vertex, and $\| \cdot \|_F$ is the Euclidean (Frobenius) norm of the matrix, i. e., the square root of the sum of the squares of all elements of the matrix, and $| \cdot |$ is the determinant of the matrix. Note that the nonlinear functional includes a barrier function $|J|/|J_{ref}|$ in the denominator. This barrier function penalizes any solution that is close to being degenerate, i. e., where the sought for Jacobian $|J|$ is much smaller than the determinant of the RJM or perhaps even vanishes.

We note that this minimization problem belongs to the class of so-called nonlinear least squares [9] for which effective minimization procedures have been developed.

Below we describe a minimization approach that takes into account the specifics of our problem.

3.2. Optimization procedure

The goal is to minimize a real-valued objective function $F = F(\ldots, x_n, y_n, \ldots)$, $n = 1, 2, \ldots, N$ of $2N$ variables, where N is the number of free nodes in the mesh. There is no theoretical proof that a unique global minimum exists, however our numerical experiments with logically rectangular mesh presented in [5] suggest that unique minimum do exist.

The basic optimization technique is a line search procedure coupled with a conjugate gradients algorithm to determine the search direction, as described in Chapters 3 and 5 of [9]. The optimization procedure is iterative, beginning with an initial (Lagrangian) mesh. The feasible region consists of the mesh node configurations that yield an invertible mesh. The initial mesh is assumed to lie in the feasible region. This assumption is checked at the beginning of the optimization procedure. The optimization procedure is initialized by computing the value of the objective function $F_0 = F(\ldots, x_n^0, y_n^0, \ldots)$ on the initial mesh and the initial gradient $(\nabla F)_0$. The initial search direction is $p_0 = -(\nabla F)_0$.

Basic steps in the iteration procedure are

- increment the iteration counter k; halt if maximum allowable count exceeded (procedure failed to find local minimum),
- compute the maximum component of the gradient vector in absolute value (i. e., the ℓ_∞ norm); halt if this is less than some initial tolerance τ (procedure has produced an iterate close to a local minimum) $\tau = 1.e - 06$ works well in practice,
- given the search direction p_k, perform the line search to get the step length $\alpha_k > 0$,
- if $\alpha_k = 0$, check the solution by changing the search direction to be minus the gradient (steepest descent) and re-doing the line search; if the step length is still zero, then the iteration successfully halts at a local minimum,
- if $\alpha_k > 0$, update the iterate (mesh): $x_{k+1} = x_k + \alpha_k p_k$,
- evaluate the objective function on the new iterate $F_{k+1} = F(\ldots, x_n^{k+1}, y_n^{k+1}, \ldots)$,
- evaluate the gradient $(\nabla F)_{k+1}$ on the new iterate,

- compute a new search direction based on conjugate gradient scheme
$$p_{k+1} = -(\nabla F)_{k+1} + \beta_k p_k$$
with β_k given by the Fletcher – Reeves
$$\beta_{FR} = |(\nabla F)_{k+1}|^2 / |(\nabla F)_k|^2$$
or Polak – Ribiere formulas [9]
$$\beta_{PR} = \beta_{FR} - (\nabla F)_{k+1} \cdot (\nabla F)_k / |(\nabla F)_k|^2$$
- go back to the start of this itemization

3.2.1. Calculation of the gradient Evaluation of the objective function is straightforward, given the mesh. The gradient of the objective function is computed numerically. For each node n of the mesh one has coordinates x_n, y_n. The nth component of the gradient is approximated by

$$\frac{\partial F}{\partial x_n} = \frac{F(\ldots, x_n + \epsilon, y_n, \ldots) - F(\ldots, x_n, y_n, \ldots)}{\epsilon},$$
$$\frac{\partial F}{\partial y_n} = \frac{F(\ldots, x_n, y_n + \epsilon, \ldots) - F(\ldots, x_n, y_n, \ldots)}{\epsilon}$$

where ϵ is chosen as some small fixed number ($1.e - 07$ works well in practice) which ensures that the trial mesh remains in the feasible region. The alternative to this numerical computation of the gradient is to first analytically calculate the gradient of F given the formula for the objective function, then to evaluate the analytic formula for the gradient on the computer. The advantage of numerically computing the gradient without using an analytic formula is that, in general the analytic formulas are very complex and time consuming to evaluate. Since the optimal search direction is not known, the small errors in the gradient due to the approximation do not significantly affect the speed of the overall optimization procedure.

3.2.2. The line search algorithm The inexact line search algorithm seeks to find a scalar $\alpha > 0$ such that the univariate function

$$\phi(\alpha) = F(x_k + \alpha p_k)$$

is minimized or is made significantly smaller than $F(x^k)$. A fixed search direction p_k is given. The line search begins with an initial guess for α, say $\alpha = 1$, and an initial mesh, x_k. Objective function values are compared at various trial meshes and the initial value of α is either increased geometrically by a factor ρ or decreased by $1/\rho$. The line search procedure consists of the following steps

- Find an α for which the trial mesh $x_t = x_k + \alpha p_k$ lies inside the feasible region. This is done by decreasing α by the factor $1/\rho$ until the trial mesh lies in the feasible region. Since x^k lies in the feasible region, we know there exists α sufficiently small such that x_t is feasible.
- *if* $\phi(\alpha) \geq \phi(0)$, decrease α until $\phi(\alpha) \leq \phi(0)$. Continue decreasing α until an increase in $\phi(\alpha)$ is found; return the last value of α.
- *else*, increase α until either no further decreases in ϕ are found or until x_t lies outside the feasible region.

3.3. Constrained optimization

In previous sections we have assumed that nodes are free to move in any direction both in the formation of RJMs and in the global optimization procedure. However, nodes which are on the boundary of the computational domain have to stay there to maintain integrity of the geometry. This means that we need to restrict their movement to be tangent to the boundary. In many practical applications there are different materials in computational domain and the interface between them has to be preserved. Thus movement of the nodes on the interface also has to be restricted to the tangential direction. In this section we describe how this can be taken into account for the case of the logically rectangular meshes. In this case boundaries and interfaces coincide with mesh lines ($i =$ const, or $j =$const).

First, we need to decide how to define boundaries and interfaces. We have chosen the following approach. To be specific, we describe our approach for an interface defined by the segmented line given by nodes $(x_{i,j}, y_{i,j} \;:\; i = i_{int},\; j = j_{\min}, \ldots, j_{\max})$. In Lagrangian mesh we are given coordinates of the nodes on this segmented line, $x_{i_{int},j}$, $y_{i_{int},j}$, which we will call reference nodes. For convenience, we introduce notations $x^r_j = x_{i_{int},j}$, $y^r_j = y_{i_{int},j}$. To describe location of the arbitrary point on this segmented line we first compute parameter t_j for each reference node by summing lengths of all segments in segmented line which occur before this reference node:

$$t^r_j = \sum_{j'=j_{\min}-1}^{j'=j} \sqrt{(x^r_{j'-1} - x^r_{j'})^2 + (y^r_{j'-1} - y^r_{j'})^2}.$$

For any parameter t, with $t^r_{j_{\min}} \leq t \leq t^r_{j_{\max}}$, coordinates $x(t)$, $y(t)$ of point on segmented line are defined as follows. If for some j_b we have $t^r_{j_b} \leq t \leq t^r_{j_b+1}$, then $x(t)$, $y(t)$ are defined by linear interpolation with respect to t of coordinates $x^r_{j_b}$, $y^r_{j_b}$, and $x^r_{j_b+1}$, $y^r_{j_b+1}$.

To summarize, the segmented line will be the "interface" on which we want interface points to remain during optimization. This interface is described by the set of triplets (x_j^r, y_j^r, t_j), which are x, y coordinates, and parameter t value in reference point. This gives enough information to find position on the interface for any value of t, $t^r_{j_{\min}} \le t \le t^r_{j_{\max}}$.

In our global minimization problem each internal node which is not on an interface or boundary is represented by two degrees of freedom, which are its coordinates $x_{i,j}$, $y_{i,j}$ and during optimization procedure we are trying to find these values which minimize our objective function. Nodes on interfaces or boundaries will be represented just by one degree of freedom — that is by parameter $t_{i_{int},j}$. During optimization we will adjust the value of this parameter. Because objective function does not explicitly depend on t, but rather depends explicitly on x and y, then each time when we need to compute objective function for new values of parameter t, we first compute corresponding values of x and y using parametric representation of interface. In particular, when we need to compute numerical derivative of objective function with respect to t, we first compute x and y which correspond to $t + \varepsilon$ and then compute value of objective function.

To preserve geometric integrity, especially for non-smooth boundaries we have an option in our algorithm not to move some nodes at all and keep them in Lagrangian positions (for example, corners of square). In some cases it is convenient to fix boundary nodes if some special procedure is used for distribution of nodes on the boundary.

3.4. Numerical examples

For ALE calculations it is very important that significant changes in the mesh occur in regions where original Lagrangian mesh has low geometrical quality, and in regions where mesh was good already it is almost untouched. This is demonstrated in the next two examples.

First, we present an example of a triangular mesh (actually Delaunay mesh for given set of nodes) in unit square, where distribution of nodes are very irregular. In Fig. 5 we present Lagrangian mesh and in Fig. 6 we present rezoned mesh.

The maximum value of condition number is reduced dramatically, and also minimal angle is increased and maximum angle is decreased significantly even though these quantities are not explicitly involved in the objective function. Longer wave-length features of the mesh have been preserved.

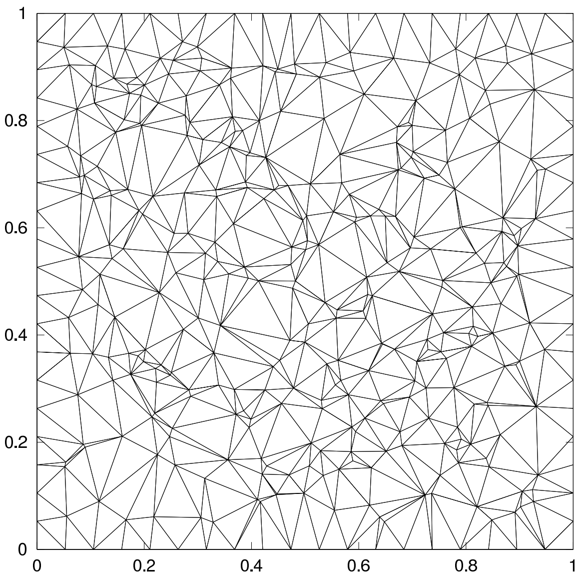


Fig. 5. Lagrangian mesh (Delaunay mesh): min angle=1.8^o, max angle =159^o; condition number: min=2, max=64.8.

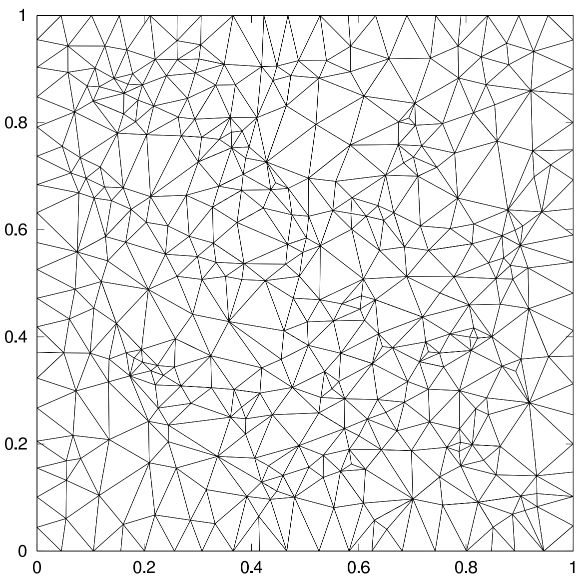


Fig. 6. Rezoned mesh: min angle=12.5^o, max angle=139.8^o; condition number: min=2, max=9.52.

Next, we present an example of our rezoning algorithm for triangular Lagrangian mesh in the domain which is unit square with one corner cut away by quadrant of circle of radius 0.2 (see Fig. 7). In

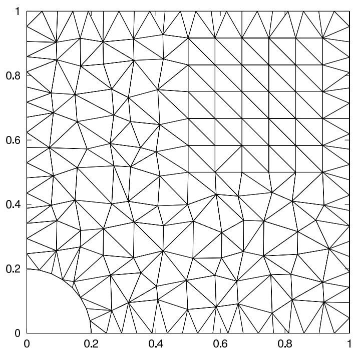


Fig. 7. Lagrangian mesh: min angle = 16.7^o, max angle = 129.2^o; condition number: min=2, max=7.25.

this example all internal nodes are free, nodes on curved piece of the boundary and in the corners are fixed, and nodes on straight segments of the boundary can move along these segments. The mesh in the subregion near right, top corner (obtained from square mesh by subdividing squares by diagonals) is very regular. For ALE, we desire that our smoothing algorithm not change the mesh in this subregion very much. The area close to curved boundary has some very distorted triangles, so we desire that our smoothing algorithm improve the quality in that area. The rezoned mesh is presented in Fig. 8; an overlay of original and rezoned mesh is given in Fig. 9. Indeed, the quality of the distorted triangles has improved while the regular portion of the mesh is unchanged. Overall quality of the mesh is improved (minimal angle is increased from 16.7^o to 22.0^o and maximum angle is decreased from 129.2^o to 120.2^o.

We next present an example of our rezoning algorithm for the case of constrained movement on nodes on interface. This example is snapshot of model calculations of Rayleigh – Taylor instability problem [10]. The Rayleigh – Taylor instability problem that we simulate consists of two ideal gases with two different densities. Initially, the heavier gas is above the lighter gas in a rectangular vessel $[0 : 1/6] \times [0 : 1]$, with a gravitational field directed vertically downward. The interface has been deliberately perturbed. Initially both gases are at rest (velocity is zero everywhere); the pressure distribution is approximately hydrostatic. It is well known that this configuration is unstable and as time progresses, the heavier gas will sink and

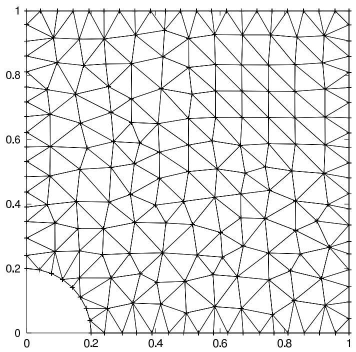


Fig. 8. Rezoned mesh: min angle = 22.0^o, max angle = 120.3^o; condition number: min=2, max=5.61.

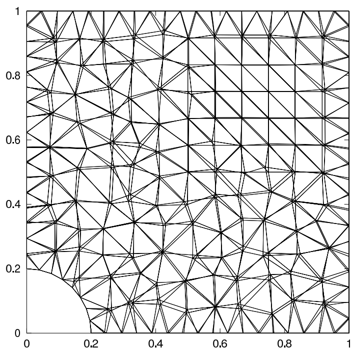


Fig. 9. Comparison of Lagrangian and rezoned meshes

the lighter gas will rise through the formation of bubbles and spikes. The time evolution of this problem leads to a rollup of the interface and the generation of significant vorticity. In Fig. 10 we present meshes obtained as result of Lagrangian step in framework of ALE calculations for some time moment corresponding to developed stage of instability. In Fig. 11 we present comparison of original Lagrangian interface with one obtained by unconstrained and constrained optimization. It is clear from this comparison that unconstrained optimization can produce interface which is even visually distinct from the original Lagrangian interface. Comparisons of fragments of the corresponding meshes is shown in Fig. 12. It is very important to understand that in ALE calculations of gasdynamics flow rezoning is performed on each time step and small differences in meshes will grow with time and can lead to quite different results.

To conclude this section we present mesh obtained as result of pure Lagrangian calculation (no rezone) of Rayleigh-Taylor problem, Fig. 13, and mesh obtained as result of ALE calculations (when described

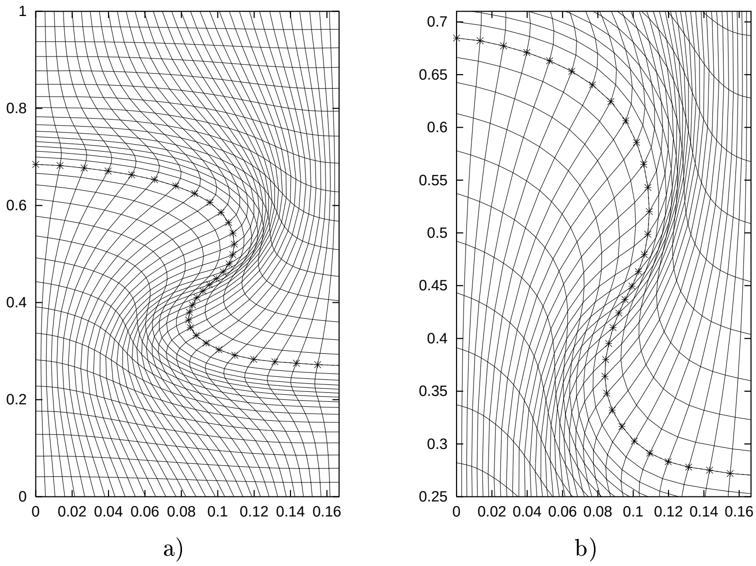


Fig. 10. Lagrangian mesh, with interface nodes marked by asterisk: a) entire mesh, b) fragment of the mesh.

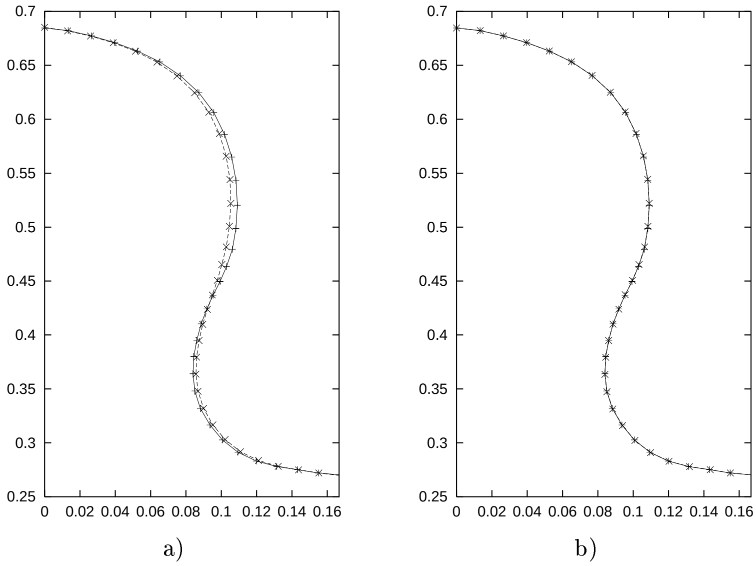


Fig. 11. Comparison of original Lagrangian interface (solid line) with one obtained by unconstrained (a) and constrained (b) optimization.

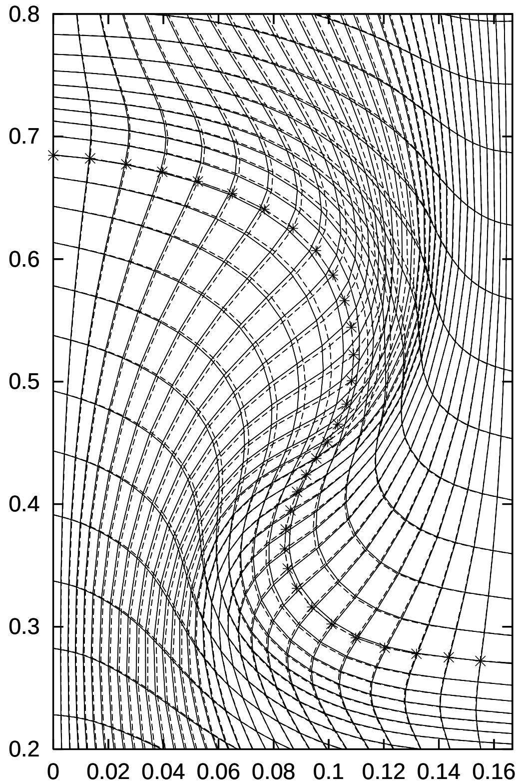


Fig. 12. Comparison of rezoned mesh obtained by unconstrained optimization, solid line, and rezoned mesh obtained by constrained optimization, dashed line. Nodes on interface obtained by constrained optimization marked by asterisk.

rezoning is performed each time step), Fig. 14. The ALE mesh is much smoother and yet more closely follows the flow features. More detailed results for this problem are presented in [5]. These results show that our ALE calculations capture the width of mixing layer much more closely than the Lagrangian approach; of equal interest, the ALE simulation reproduces some of the details of the interface shape.

4. Conclusion and future work

In this paper we have described a general framework for rezone strategies based on the reference Jacobian approach. This approach allows us to construct a mesh with optimized geometrical quality whose

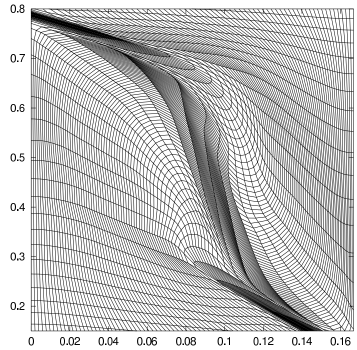


Fig. 13. Mesh obtained as result of pure Lagrangian calculations

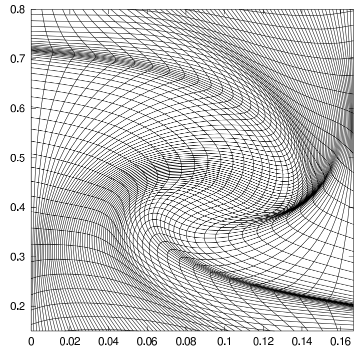


Fig. 14. Mesh obtained as result of ALE calculations

Jacobian matrix is closest (in the sense of least squares) to a locally prescribed reference Jacobian matrix. We have described how this general framework can be used to ensure that the mesh has appropriate geometrical quality and is guaranteed to be valid, while keeping the mesh nearly Lagrangian and aligned with the boundary and any internal interfaces. This is achieved by a special procedure of constructing RJM's, based on local optimization of the condition number.

The reference Jacobian framework is general enough to allow us to include more information about the solution in the construction of the RJM. In particular we are planning to incorporate error estimates [11, 12] to create truly adaptive ALE algorithms, where error will be minimized by appropriate choice of computational mesh. Our approach also can be used for generating meshes similar to reference mesh in context of design process [13].

In principle, our approach can be naturally extended to three dimensional unstructured meshes, which is the subject of our current work.

Acknowledgment

The work of P. Knupp was funded by the Department of Energy's Mathematics Information and Computational Sciences Program (SC-31) and was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. The work of M. Shashkov was performed under the auspices of the US Department of Energy under contract W-7405-ENG-36 and DOE/BES Program in the Applied Mathematical Sciences Contract KC-07-01-01. M. Shashkov acknowledge the support of DOE's Accelerated Strategic Computing Initiative (ASCI) and support from LDRD-ER (# 2000045) at Los Alamos National Laboratory. Authors thank L. Margolin for many fruitful discussions and suggestions.

References

1. Margolin, L. (1997): Introduction to arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys., **135**, 198–202.
2. Mair, H. U. (1999): Review: Hydrocodes for Structural Response to Underwater Explosions, Shock and Vibration 6/2, pp. 81–96.
3. Wang, J. and Gadala, M. (1997): Formulation and survey of ALE method in nonlinear solid mechanics, Finite Elements in Analysis and Design, **24**, No. 4, 253–269.
4. Perry, J. and Carroll, D. (2000): Multi-material ALE methods in unstructured grids, Comput. Meth. Appl. Mech. Engin., **187**, No. 3–4, 591–619.
5. Margolin, L., Knupp, P., and Shashkov, M. (2002): Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian-Eulerian methods, J. Comput. Phys., **176**, 93–128.
6. Knupp, P. (2000): Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I — a framework for surface mesh optimization, Int. J. Numer. Meth. Engr., **48**, 401–420.
7. Knupp, P. (2000): Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II — a framework for volume mesh optimization and condition number of the Jacobian matrix, Int. J. Numer. Meth. Engr. **48**, 1165–1185.
8. Knupp, P. (1996): Jacobian-weighted elliptic grid generation, SIAM J. Sci. Comput., **17**, No. 6, 1475–1490.
9. Nocedal, J. and Wright, S. (1999): *Numerical Optimization*, Springer, New York.
10. Kull, H. J. (1991): Theory of the Rayleigh – Taylor instability, Physics Reports, **206**, 197–325.
11. Ainsworth, M. and Oden, J. T. (2000): *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley & Sons. Inc.
12. Kroner, D and Ohlberger, M. (1999): A posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions, Math. of Comp., **69**, 25–39.

13. Yamada, A., et. al. (2000): An Approach for generating meshes similar to reference mesh, in: Proceedings of 9th International Mashing Roundtable, pp. 101–107.