Reference Jacobian Optimization-Based Rezone Strategies for Arbitrary Lagrangian Eulerian Methods

Patrick Knupp

Parallel Computing Sciences Department, Sandia National Laboratories

MS-0847, P.O. Box 5800, Albuquerque, NM 87185-0847

Len Margolin

Center for Nonlinear Studies, Los Alamos National Laboratory

MS-B258, Los Alamos, NM 87545

Mikhail Shashkov

Theoretical Division, T-7, Los Alamos National Laboratory

MS-B284, Los Alamos, NM 87545

Subject Classification Numbers: 65N05, 65N10, 65N15, 80A20

Keywords: Rezone Strategies, ALE Methods, Remapping

Running Head: **Reference Jacobian Rezone Strategies**

Send Proofs to:

Dr. Mikhail Shashkov

Mail Stop B284

Los Alamos National Laboratory

Los Alamos, NM 87545

Phone: (505) 667-4400

Fax: (505) 665-5757

E-mail: shashkov@lanl.gov

Abstract

The philosophy of the Arbitrary Lagrangian-Eulerian (ALE) methodology for solving multi-dimensional fluid flow problems is to move the computational grid, using the flow as a guide, to improve the accuracy and efficiency of the simulation. A principal element of ALE is the rezone phase in which a "rezoned" grid is created that is adapted to the fluid motion. We will describe a general rezone strategy that ensures the continuing geometric quality of the computational grid, while keeping the "rezoned" grid at each time step as close as possible to the Lagrangian grid. Although the methodology can be applied to more general grid types, here we restrict ourselves to logically rectangular grids in two dimensions.

Our rezoning phase consists of two components: a sequence of local optimizations followed by a single global optimization. The local optimization defines a "reference" Jacobians which incorporates our definition of mesh quality at each point of the grid. The set of reference Jacobians then is used in the global optimization. At each node we form a local patch from the adjacent cells of Lagrangian grid and construct a local realization of the Winslow smoothness functional on this patch. Minimization of this functional with respect to the position of the central node defines its 'virtual' location (the node is not actually moved at this stage). By connecting this virtually-moved node to its (stationary) neighbors, we define a reference Jacobian that represents the best locally achievable geometric grid quality.

The "rezoned" grid results from a minimization (where the points are actually moved) of a global objective function that measures the distance (in a least-squares sense) between the Jacobian of the rezoned grid and the reference Jacobian. This objective function includes a "barrier" that penalizes grids whose cells are close to being inverted. The global objective function is minimized by direct optimization leading to the rezoned grid. We provide numerical examples to demonstrate the robustness and the effectiveness of our methodology on model examples as well as for ALE calculations of Rayleigh-Taylor unstable flow.

1 Introduction

In numerical simulations of multidimensional fluid flow, the relationship of the motion of the computational grid 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 grid moves with the local fluid velocity, while in the Eulerian framework the fluid flows through a grid fixed in space.

More generally, the motion of the grid can chosen arbitrarily. The philosophy of the Arbitrary Lagrangian-Eulerian methodology (ALE; cf [22], [3], [4], [35], [23], [24], [39]) is to exploit this degree of freedom to improve the accuracy and efficiency of the simulation. The main elements in an ALE simulation are an explicit Lagrangian phase, a rezone phase in which a new grid is defined, and a remapping in which the Lagrange solution is transferred to the new grid [35]. Of these three processes, it is the rezone phase that is least developed. The purpose of this paper is to develop a general rezone strategy that ensures the continuing geometric quality of the computational grid while keeping the rezoned grid at each time step as close as possible to the Lagrangian grid.

Maintaining geometric quality is one general goal of rezoning. The other general goal is adaptation, in which the grid is moved to better resolve regions of rapid variation of the flow variables, or more generically to reduce some measure of the solution error. Each of these two goals can be attained by the minimization of an appropriately constructed global functional. However a difficulty arises when one tries to combine these — how should one weight the relative importance of these separate goals? In particular, the two global functionals will have distinct (physical) dimensions, and so can only be combined with some dimensional constant. At present, there is no theoretical basis for choosing this constant, thus delegating the decision to the user. Although effective choices can be found for particular problems, the key to a successful ALE code is a robust rezoning algorithm that does not require user intervention.

On the other hand, if one restricts oneself to only optimizing the overall geometric quality of the grid then another problem arises — optimizing the grid while ignoring the details of the flow usually reduces the overall accuracy of the simulation. Moreover, the results of such an optimization strongly depend on the chosen measure of mesh quality and in many cases the optimization problem may not have a unique solution. The resolution of this apparent dilemma is that one must define the measure of geometric quality in the context of the solution. In this paper, we will focus on defining a suitable measure of geometric quality, and will postpone consideration of the issues related to a combined measure that includes explicit adaptation. We will restrict our attention to logically rectangular grids in two dimensions. However, the ideas underlying our rezone strategies are extendable to three dimensions and to unstructured grids.

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 grid 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 the rezoned grid remain close to the Lagrangian grid but "smoother" (i.e., better geometrical quality). By requiring the rezoned grid to remain as close as possible to the Lagrangian grid, we will minimize the error of the remap phase, and further will justify employing a local remapper [4, 5] in which mass, energy and momentum are simply exchanged between neighboring cells. Local remappers are computationally much more efficient than global remappers [16], which make no assumptions about the relationship of the old and new grids.

One can consider a logically rectangular grid in physical space as resulting from a map of the uniform grid in a unit square of logical space. The most fundamental object describing the map is its Jacobian matrix. Therefore, one can expect that if two maps have similar Jacobian matrices, then the maps themselves must be similar, and so will produce grids close to each other. 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 grid deformation. It is not possible to use the RJMs defined at each vertex by themselves to derive the new (i.e., rezoned) positions of the vertices. To begin, each RJM 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 (not on a boundary) belongs to four cells; a simple rezone strategy applied individually to each cell will lead, in general, to four incompatible specifications of the rezoned position of any vertex. The resolution is to construct a global functional that measures the difference between the RJM (which depends on the Lagrangian grid and the smoothing process) and the rezoned Jacobian (not known before the minimization process) of all cells in the grid. Minimizing this functional over the entire grid (as a function of each of the vertex coordinates) then leads to the rezoned grid.

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 as described in [13] and [44]. The barrier function effectively penalizes any solution where the area of the cell is much smaller than the area derived from the RJM. We note that this minimization problem belongs to the class of so-called nonlinear least squares [37] for which effective minimization procedures have been developed.

The outline of our paper is as follows. In the next section, we will review some standard approaches to rezoning, give background and outline the main ideas of our research. In Section 3 we will describe the construction of the grid for a given set of reference Jacobian matrices, one specified for each grid point. The construction involves the formation and minimization of a global function that measures the difference between the specified "desired" RJM's and the Jacobian matrices of the new rezoned grid. We will provide numerical examples to suggest that the constrained optimization problem has a unique solution. We will also demonstrate how our approach works when the RJM is computed from a nonsmooth but continuous map, and in the more difficult case of incompatible RJMs, obtained from several different mappings. In Section 4 we will describe our local optimization procedure, which is based on a local realization of the Winslow smoothing functional, leading to a definition of the RJM's at each node of the grid. We describe the properties of such a procedure and give computational examples. In Section 5 we will present "stationary" numerical examples not in the context of an ALE simulation, but which illustrate the results of our rezone strategy. In Section 6 we will present preliminary results using our rezone strategy in ALE calculations of Rayleigh-Taylor instability problem, and compare these with a purely Lagrangian calculation and also with a particular implementation of the rezone strategy based on the Winslow smoother. In Section 7 we will summarize our results and describe future work.

2 Background and Rationale

2.1 Brief Review of Rezoning Methods

Rezoning methods can be divided into two broad classes — velocity-based and coordinate (grid) based. In the coordinate-based class, there are two further subclasses differentiated by whether the rezoning strategy is local or global. In this section we give a brief description of each of these.

There are many strategies in the velocity-based approach. One of the more popular is based on solving the vector Poisson equation (see for example, [33], [42], [17]), using the fact that this equation with smooth coefficients and smooth right-hand-side has a smooth solution (here smooth is used in its mathematical sense of having continuous derivatives of many orders). This approach is most useful for problems such as the interaction of a fluid with a structure, where the dynamics of the flow are due primarily to the moving structure. For example, in [42] the authors solve an equation for grid velocity $\Delta\,\mathbf{u}_g = 0$ with the normal components of velocity prescribed on the boundary. A velocity-based approach also has been used in shock calculations [17]; here the authors introduce a *near Lagrangian velocity*, which is the solution of the following equation: $\Delta\,\mathbf{u_g} = \mathbf{grad}\,\mathbf{div}\,\mathbf{u}_L$, where $\mathbf{u}_L$ is the Lagrangian velocity. Since $\Delta = \mathbf{grad}\,\mathbf{div} - \mathbf{curl}\,\mathbf{curl}$, the right-hand-side of this equation is equal to

$\Delta \mathbf{u}_L$, but with the curl term removed. Here one hopes that the grid velocity obtained from this equation will have a divergence close to the divergence of the Lagrangian velocity, while its curl is close to zero. The underlying idea is that the Lagrangian grid becomes tangled as a result of the vorticity in the flow. Unfortunately, this approach in general does not guarantee that the rezoned grid is unfolded. Also, whenever the flow is close to incompressible, the velocity divergence will be nearly zero independent of how large the deformations of the fluid might be, and the "near Lagrangian velocity" approach has the trivial solution, $\mathbf{u}_g = 0$.

Let us now briefly review coordinate-based approaches. In the local coordinate-based approach, [3], [19], [39], one decides which nodes will be moved from their Lagrangian positions by using local criteria. In ([3], p.311, [39]), for example, the authors calculate two numbers at each node — the ratio of the minimum and maximum areas among the cells that share that node and also the maximum value of the cosines of the angles for whom that node is the central vertex. Based on these two numbers, the criteria then determine whether a node should be moved. The new locations of the nodes that are to be moved are found using a local smoothing technique. For example, in [39] authors use a method known as the Tipton smoother, which is based on a finite element approximation to Brackbill's 'smoothness' functional [6]. There are several problems with this approach. First, the criteria for choosing which nodes should be moved is based on *ad hoc* grid quality measures; the parameters of the movement criteria need to be tuned at least to the class of problem, or more specifically tailored for the specific problem at hand. Second, the node movement strategy does not guarantee that the final grid is unfolded [12].

Brackbill and Saltzman [6] developed a global coordinate approach based on modifying the Winslow algorithm [45]. In [6] extra terms are added to the smoothness functional to adapt to regions of steep gradients. One difficulty that arises is that there is no theory to guide the relative weights (which have different physical dimensions) of these terms, which are left to the user to specify. In addition, because the functional does not include any information about the Lagrangian grid, the solution grid can be very far from the Lagrangian grid, which precludes the use of a local remapper. Indeed, in general one does not know if a solution to the equations exists (unless it is a pure smoothness functional), and so an alternative strategy has been to generate an approximate solution. This approximation results from undertaking only a few iterations towards the solution, starting from the Lagrangian grid. In this approach, there is no theory for what kind of iterations to use nor how many iterations to allow, nor is there any guarantee that the resulting grid will be unfolded. More details on this rezone strategy and a comparison of results for an ALE calculation of Rayleigh-Taylor instability using a modified Winslow strategy and our new RJM strategy are presented in Section 6.

Finally, we note that rezoning strategies that are based only on formal minimization of the error, such as the moving finite element method ([2], [15]) produce degenerate or nearly degenerate grids, and require special regularizing terms (i.e., penalty functions). There is as yet no theory for the form and magnitude of these terms, which must presently be supplied by the user.

To summarize, most of the strategies currently used for rezoning do not guarantee an unfolded mesh nor do they guarantee that the smoothed mesh is close to the Lagrangian mesh. The coordinate-based approach seems a more appropriate framework in which to base our approach, and here both the local and the global approaches have attractive features.

2.2 Motivation and General Description

In this section we will give an overview of our method, which 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 a "reference" Jacobian (J_{ref}), that incorporates our definition of mesh quality at each point of the grid. The "rezoned" grid results from minimizing a global objective function that measures the distance (in a least-squares sense) between the Jacobians of the rezoned grid and the reference Jacobians — stage II in Fig. 1. The global optimization resolves incompatibilities of the locally defined RJMs in a smooth manner.

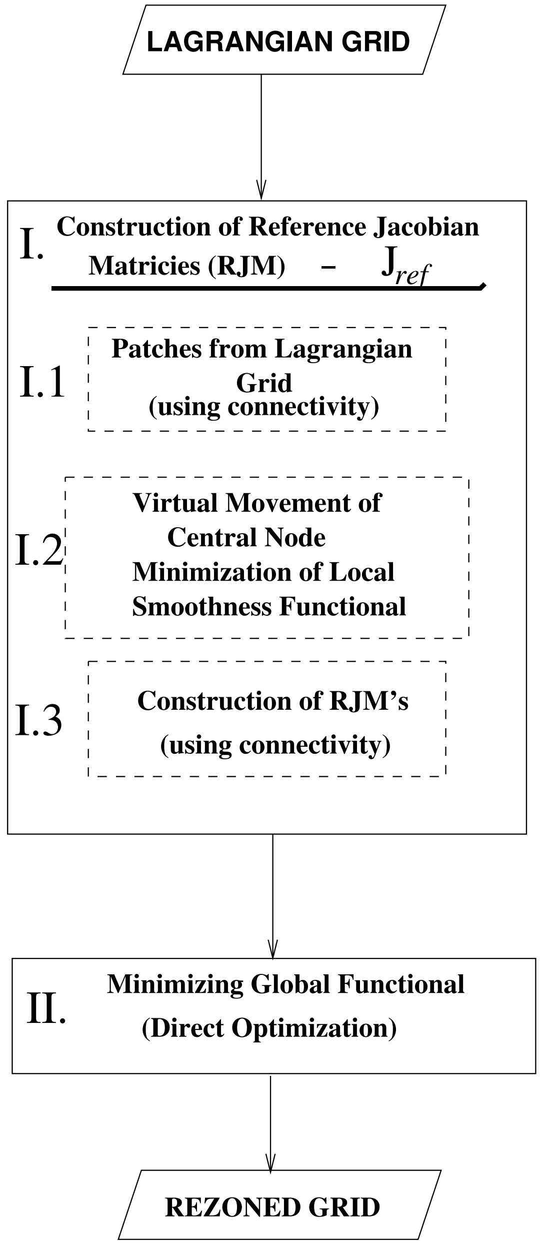
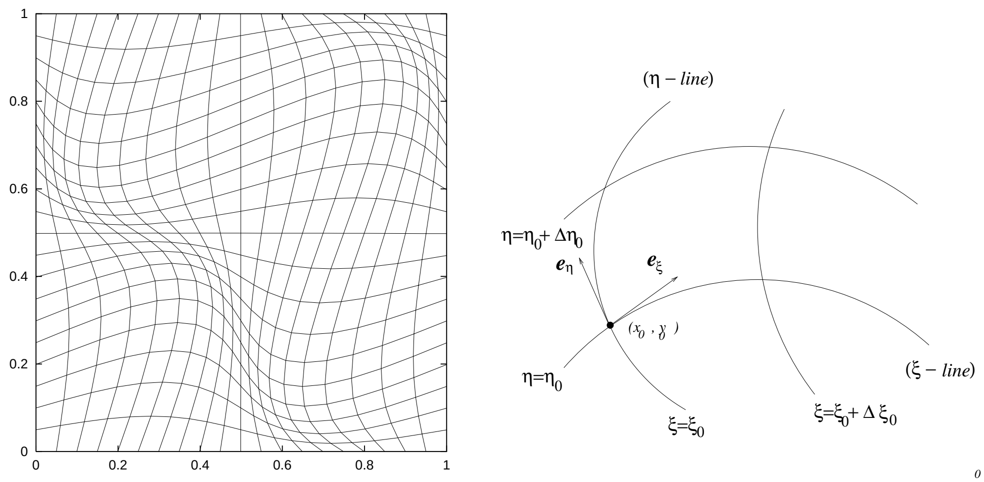


Figure 1: *Flow-chart of rezoning algorithm.*



a) b)

Figure 2: *a) Image of lines $\xi = const$ and $\eta = const$ under map $x(\xi,\eta)$, $y(\xi,\eta)$, b) coordinate lines and tangent vectors.*

Because of the fundamental nature of the Jacobian matrix, we begin by elucidating its role in the map $x(\xi,\eta)$, $y(\xi,\eta)$ from logical space $(\xi,\,\eta)$ to physical space $(x,\,y)$. The most general description of the continuum map is provided by its Jacobian matrix

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix},$$

where $x_\xi = \partial x/\partial\xi$, $x_\eta = \partial x/\partial\eta$, etc. For our purposes it is convenient to represent the Jacobian matrix J in terms of two vectors in physical space $\mathbf{e}_\xi = (x_\xi,\, y_\xi)^T$, $\mathbf{e}_\eta = (x_\eta,\, y_\eta)^T$ (Fig. 2 b)):

$$J = \left(\mathbf{e}_\xi \;\; | \;\; \mathbf{e}_\eta \right). \tag{2.1}$$

Since the map is one-to-one, to each point in physical space [say (x_o, y_o)] there corresponds a unique point in logical space (ξ_o, η_o). Now the straight lines in logical space $\xi = \xi_o$ and $\eta = \eta_o$ map into curved "coordinate" lines in real space, whose intersection is the point (x_o, y_o). The vectors $\mathbf{e}_\xi$ and $\mathbf{e}_\eta$ are the tangent vectors to the coordinate lines at the intersection point. This is illustrated in Fig. 2 b).

The tangent vectors describe the orientation of the coordinate lines, and how rapidly the x- and y-coordinates change along them. The normalized scalar product of the tangent vectors, $\mathbf{e}_\xi \cdot \mathbf{e}_\eta$, is the cosine of the angle between the coordinate lines. The determinant of the Jacobian matrix, $|J|$, which is the cross product of tangent vectors $\mathbf{e}_\xi \times \mathbf{e}_\eta$, is the image area in physical space of an elementary area in logical space. Thus, knowing the tangent vectors (which is the same as knowing the Jacobian matrix) for all ξ and η is a full description of the map. Indeed, it can be shown that given the Jacobian matrix obtained from a valid map (to be defined shortly) at any point (ξ, η) and given consistent (x, y) on the boundary, one can construct the map (that is, the functions $x = x(\xi, \eta)$; $y = y(\xi, \eta)$) for the entire domain. One can easily verify that $x = x(\xi, \eta)$; $y = y(\xi, \eta)$ satisfy two decoupled Poisson equations with Dirichlet boundary conditions

$$\frac{\partial^2 x}{\partial \xi^2} + \frac{\partial^2 x}{\partial \eta^2} = \frac{\partial J_{11}}{\partial \xi} + \frac{\partial J_{12}}{\partial \eta}; \quad \frac{\partial^2 y}{\partial \xi^2} + \frac{\partial^2 y}{\partial \eta^2} = \frac{\partial J_{21}}{\partial \xi} + \frac{\partial J_{22}}{\partial \eta}. \tag{2.2}$$

These Poisson equations have unique solutions that define the map if the elements of the Jacobian matrix constitute a valid map.

Because of the fundamental nature of the Jacobian matrix one can expect that if two maps have similar Jacobian matrices, then the maps themselves must be similar. This leads to the following idea. Let us first construct the Jacobian matrix as a function of (ξ, η) — termed the reference Jacobian matrix (RJM or J_{ref}). This RJM incorporates the desired properties of the map. It is clear that in general it is not possible to construct a valid map for arbitrary specified RJM. For example, if the RJM

$$\tilde{J} = \begin{pmatrix} \tilde{J}_{11} & \tilde{J}_{12} \\ \tilde{J}_{21} & \tilde{J}_{22} \end{pmatrix},$$

is to define a valid map $\tilde{x}(\xi, \eta)$, $\tilde{y}(\xi, \eta)$, then the conditions

$$\partial \tilde{J}_{11}/\partial \eta = \partial \tilde{J}_{12}/\partial \xi \text{ and } \partial \tilde{J}_{21}/\partial \xi = \partial \tilde{J}_{22}/\partial \eta, \tag{2.3}$$

must be satisfied because of the commutability of differentiation

$$\frac{\partial^2 \tilde{x}}{\partial \xi \partial \eta} = \frac{\partial^2 \tilde{x}}{\partial \eta \partial \xi}, \quad \frac{\partial^2 \tilde{y}}{\partial \xi \partial \eta} = \frac{\partial^2 \tilde{y}}{\partial \eta \partial \xi}.$$

Further, to be a valid map, the determinant of $\tilde{J}$ has to be positive everywhere. There are other compatibility conditions that $\tilde{J}$ has to satisfy as well to be a valid map.

Therefore, our strategy will be to find a valid map whose Jacobian matrix *is as close as possible to the RJM.* This can be done by constructing and minimizing a nonlinear functional that measures the deviation of the (unknown) Jacobian matrix, J, of desired map from a J_{ref}, in a least-squares sense at every point of the domain

$$F\left(x(\xi,\eta),\, y(\xi,\eta)\right) = \int_0^1 \int_0^1 \frac{||J - J_{ref}||_F^2}{|J|/|J_{ref}|}\, d\xi\, d\eta\,. \tag{2.4}$$

Here $||\cdot||_F$ 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. To ensure that the functional is bounded from below, we restrict the set of admissible maps to those that are invertible and whose determinant is positive. Note that the nonlinear functional includes a barrier function ($|J|/|J_{ref}|$) in the denominator (c.f., [13] and [44]). 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.

So far, our remarks have focused on the continuum formulation of the rezoning problem. Our *general framework* for constructing rezoned grid will be based on the discrete version of the minimization problem (2.4). In the discrete case, the tangent vectors $\mathbf{e}_\xi$, and $\mathbf{e}_\eta$ become the edges of the cell, (see Fig. 3). In general, each grid point belongs to four different cells and so has four possible pairs of tangent vectors. Each set of tangent vectors defines a triangle and also a Jacobian matrix, similar to (2.1), where roles of $\mathbf{e}_\xi$, $\mathbf{e}_\eta$ are now played by edge tangent vectors. Thus we have four different Jacobian matrices, each related to a distinct triangle at the grid point.

¿From a different point of view, each cell defines four triangles, two formed by the bisection of the cell by one of its diagonals, two more by the other diagonal. The totality of these triangles defined by the cells is of course the same set as that defined by the nodes in the previous paragraph. Now for a rectangular cell to be convex (unfolded) it is necessary that each of these triangles has positive area. Thus the condition that each of triangles associated

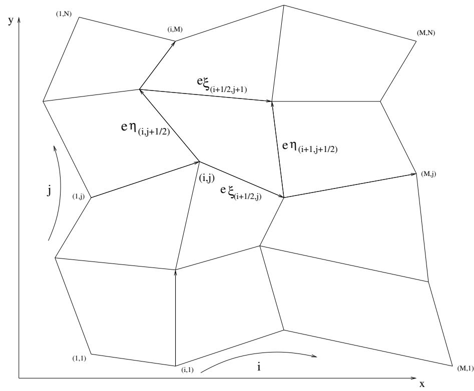


Figure 3: *Logically rectangular grid and discrete tangent vectors.*

with a node has positive area (i.e., the determinant of the $|J| > 0$), extended to the complete set of nodes, guarantees that the mesh is unfolded.

As in the continuous case, the tangent vectors define geometry of the grid: the vector cross product is the twice the area of the triangle, the scalar product defines angles of the triangle, and vectors themselves define the orientation and lengths of the sides of the quadrilateral cells. We note that the compatibility condition (2.3) in the discrete case is played by following equation $\mathbf{e}\xi_{i+\frac{1}{2},j+1} - \mathbf{e}\xi_{i+\frac{1}{2},j} = \mathbf{e}\eta_{i+1,j+\frac{1}{2}} - \mathbf{e}\eta_{i,j+\frac{1}{2}}$, which can be rewritten as

$$\mathbf{e}\xi_{i+\frac{1}{2},j} + \mathbf{e}\eta_{i+1,j+\frac{1}{2}} = \mathbf{e}\eta_{i,j+\frac{1}{2}} + \mathbf{e}\xi_{i+\frac{1}{2},j+1} \,.$$

This last equation guarantees that the cell is a closed figure, or in other words that one can reach node $(i+1, j+1)$ starting from node (i, j) using either pair of cell edges. As in the case of a continuous map, if one has a set of edge tangent vectors that are obtained from a valid grid and coordinates of grid points on the boundary one can reconstruct the entire grid.

Following the logic described for the continuous map, to construct a grid with desired properties we will first define set of reference Jacobian matrices (RJMs) in each triangle, by specifying reference edge vectors. One could employ (and even combine) different strategies to construct the RJMs. Here we provide some motivation for the particular strategy that we will develop in more detail in section 4 — stage I in Fig. 1.

In the general framework of ALE simulations and of our algorithm in particular, we will need to consider three different grids. At the beginning of the computational cycle, we have the grid from the previous (i.e., n-th time level) cycle. We will assume that this grid is "good" quality both in the sense that it is unfolded and also in the sense that it optimally represents the flow at this time. We term this the "initial" grid. A second grid is the Lagrangian grid at the $(n+1)$-th time level. This grid is obtained by moving each of the nodes of the initial grid with the Lagrangian fluid velocity. We term this the "Lagrangian" grid, and note that it is not necessarily optimal and so may require rezoning (i.e., adjusting the grid). In general terms, our rezone strategy is to improve the geometrical quality of Lagrangian grid where necessary, while preserving the Lagrangian grid as much as possible. The grid resulting from the rezoning algorithm is called the "rezoned" grid. This rezoned grid will become the initial grid for the next computational cycle. Finally, we will need to define the flow parameters on the rezoned grid. This is done by remapping (i.e., redefining the solution on the new grid) from the Lagrangian grid.

There are many reasons to maintain the rezoned grid close to the Lagrangian grid. Of greatest importance is that Lagrangian grid contains important information about the flow. For example, a Lagrangian grid follows the interface between different materials. More generally, the Lagrangian grid eliminates the numerical diffusion associated with approximating the advective terms, which vanish in this representation.

A second reason is that the accuracy of the remapping stage strongly depends on how close the Lagrangian and the rezoned grid are. A third reason is that when the Lagrangian and the rezoned grids are close, one can use local remapping procedures (e.g., based on advection) rather than more general global remappers based on interpolation; local remappers are typically much more efficient.

On the other hand, the choice of a purely Lagrangian calculation is usually not optimal. At a minimum, we should require that the grid remain unfolded to maintain a physically realizable solution. In general, the discretization error will be smaller on a "smooth" grid, an advantage that must be weighed against the error resulting from the remap.

As asserted above, the Jacobian matrix is the natural object with which to analyze and control the properties of the grid. In particular, to produce a smooth grid that is close to the Lagrangian grid, we will begin with the Jacobians that correspond to the Lagrangian grid and modify them to form the RJMs These modifications will be devised locally. For example, on the 2D structured grid, we will consider a central grid point and its nearest eight neighbors which form a patch of four cells — stage I.1 in Fig. 1. Each cell can be subdivided into two sets of two triangles by its two principal diagonals. Hence there are a total of sixteen triangles in the patch, twelve of which will be altered by the displacement of the central node.

Now suppose that we have some method of choosing a better position for this central node, while keeping all the other grid points of the patch fixed — stage I.2 in Fig. 1. This "virtual" displacement leads to new edge vectors (reference edge vectors) and hence a new Jacobian for each of the twelve triangles — stage I.3 in Fig. 1. These new Jacobians are the RJMs at this grid point. We will describe a method of choosing this displacement in detail in section 4.

Finally, let us note that geometry is not the only basis for evaluating the smoothness of the grid. For example, one classic strategy in multimaterial simulations is based on mass matching — that neighboring cells should have roughly the same mass. This inertial matching serves to mitigate the unphysical reflection of shock waves at material interfaces. The zoning to achieve mass matching depends on the the density field as well as geometry. Other dependencies include the flow itself, where the zoning may be chosen to represent the expected symmetries of the solution, and even the goals of the modeler, who may have a special interest more finely resolving a particular region of the flow.

In the future, we hope to construct a framework general enough to deal with all of these issues and constraints. For the moment, we are pursuing the less ambitious strategy of building into the framework a flexibility to control and even turn off the rezoning in an automatic fashion.

As in the continuous case, the reference edge vectors that participate in construction of RJMs in general do not satisfy conditions (2.2). We will construct the discrete analog of the functional (2.4) using Jacobians defined in triangles — stage II in Fig. 1. We will solve the minimization problem directly, rather than using the discrete analog of the Euler-Lagrange equations, which are analytically (but not numerically) equivalent. The minimization procedure is discussed in Section 3.2. We note here that RJMs are translationally invariant and therefore in specifying the RJMs related to triangles we not imposing restrictions related to the location of this triangle, but only its shape, size, and orientation. The global optimization procedure takes advantage of this fact and resolves incompatibilities of the locally defined RJMs in a smooth manner.

We conclude this section by noting an idea similar to the RJM to adapt structured meshes [26]. There the strategy was based on a variational principle related to the 'smoothness' functional of [6], and led to Euler-Lagrange equations that were a weighted form of Winslow's equations [45]. It was demonstrated numerically that the method produces grids whose Jacobian matrix agrees with the reference Jacobian matrix, provided the latter was constructed from a valid grid. Furthermore, the method could align mesh coordinate lines with desired directions given by the reference Jacobian matrix. The primary differences between the present method and that in [26] are (1) we do not use Euler-Lagrange equations but rather directly minimize a discrete analog of the functional; (2) we use a barrier method similar to that of [13], [44] to ensure an unfolded grid; (3) we use the Jacobian matrix, rather than its inverse. The reasons for introducing these differences are (1) it is not clear that the method in [26] guarantees positive area cells (especially in three dimensions); (2) the use of the Jacobian matrix yields a simpler algorithm than would result from using its inverse;

(3) the method of direct minimization given here can be extended to unstructured meshes whereas the formulation in [26] is limited to structured meshes[1].

To summarize, we believe that the general framework described above, based on the minimization of a discrete analog of a global functional 2.4 can be extended to rezone a wide class of grids whose desirable properties include adaptivity as well as smoothness, if these properties can be described in terms of RJMs.

3 Construction of the Grid for Given Reference Jacobian Matrix

In this section we describe the construction of the objective function (functional) and our optimization procedure in more detail — stage II in Fig. 1. We assume we are given a reference Jacobian matrix J_{ref} with positive determinant at each grid point; our goal is to construct a global map whose Jacobian is close to the RJM and that also has positive determinant everywhere (so that the grid remains unfolded).

In Section 3.1 we will introduce notation for grid nodes, edges, and triangles, and will construct the discrete Jacobian matrices and the discrete objective function to be minimized. In Section 3.2 we will describe the numerical optimization procedure. At present there is no proof that the minimization of the discrete functional is unique. In Section 3.3, we will present numerical examples that suggest the uniqueness of the resulting grid.

3.1 Discrete Objective Function

We adopt the following notation to describe the nodes, edges, and triangles associated with a logically rectangular grid. Grid points are labeled by two indices, $i = 0, \ldots, I$ and $j = 0, \ldots, J$, which are the coordinates in logical space. The increments of the logical coordinates are $\Delta\xi = \frac{1}{I}$ and $\Delta\eta = \frac{1}{J}$. The grid point (i, j) has coordinates $x_{i,j}$ and $y_{i,j}$ or in vector

[1]We will pursue the extension of the present method to three dimensions and to unstructured meshes in a later paper.

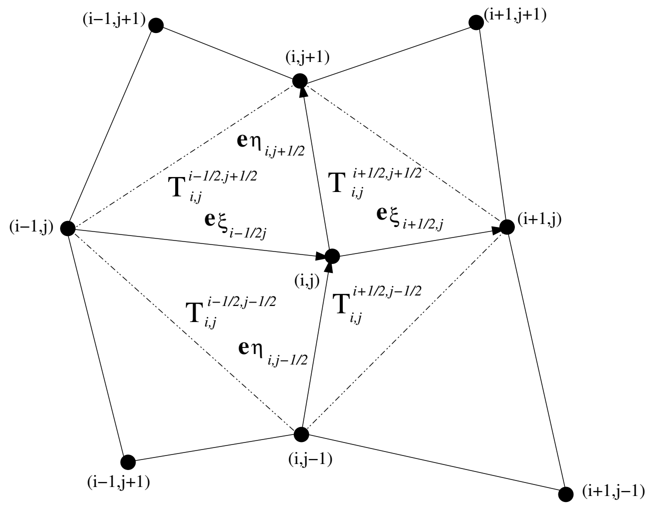


Figure 4: *Discrete tangential vectors and triangle (Jacobian Matrix) indexing.*

notation $\mathbf{x}_{i,j} = (x_{i,j}, y_{i,j})^T$. The computational cells have fractional indices $(i+\frac{1}{2}, j+\frac{1}{2})$, where $i = 0, \ldots, I-1\,;\ j = 0, \ldots, J-1$. ξ-edges have indices $(i+\frac{1}{2}, j)$ and η-edges have indices $(i, j+\frac{1}{2})$.

At each interior node, we define four triangles as shown in Fig. 4; for nodes on the boundary there are only two triangles, and corner nodes have just one. Triangles are labeled by two indices, a subscript that specifies the node, and a superscript that specifies the cell; for example, $T_{i,j}^{i+\frac{1}{2},j+\frac{1}{2}}$ signifies the triangle in cell $(i+\frac{1}{2}, j+\frac{1}{2})$ with sides $(i+\frac{1}{2}, j)$ and $(i+1, j+\frac{1}{2})$ (see Fig. 4).

The discrete analogs of the tangential vectors $\mathbf{e}_\xi$ and $\mathbf{e}_\eta$ defined in Section 2 are associated with cell edges (see Fig. 3, and Fig. 4) and will be written as follows

$$\mathbf{e}\xi_{i+\frac{1}{2},j} = \mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}\,, \quad \mathbf{e}\eta_{i,j+\frac{1}{2}} = \mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}\,. \tag{3.1}$$

Now in the continuous case of a smooth map, the Jacobian matrix (2.1) is uniquely defined everywhere. In the discrete case we can define four different Jacobian matrices (which are each approximations of the continuous Jacobian matrix) using the four sets of discrete tangential vectors that meet at each grid point:

$$
\begin{aligned}
J_{i,j}^{i+\frac{1}{2},j+\frac{1}{2}} &= \left(\mathbf{e}\xi_{i+\frac{1}{2},j} \,|\, \mathbf{e}\eta_{i,j+\frac{1}{2}}\right)\,, \quad J_{i,j}^{i-\frac{1}{2},j+\frac{1}{2}} = \left(\mathbf{e}\xi_{i-\frac{1}{2},j} \,|\, \mathbf{e}\eta_{i,j+\frac{1}{2}}\right)\,, \\
J_{i,j}^{i-\frac{1}{2},j-\frac{1}{2}} &= \left(\mathbf{e}\xi_{i-\frac{1}{2},j} \,|\, \mathbf{e}\eta_{i,j-\frac{1}{2}}\right)\,, \quad J_{i,j}^{i+\frac{1}{2},j-\frac{1}{2}} = \left(\mathbf{e}\xi_{i+\frac{1}{2},j} \,|\, \mathbf{e}\eta_{i,j-\frac{1}{2}}\right)\,.
\end{aligned}
\tag{3.2}
$$

We will use the same index notation for Jacobian matrices as for triangles. This reinforces the relation between discrete Jacobian matrix and its corresponding triangle.

The *discrete* objective function that is minimized has the form:

$$
F(\ldots x_{ij}, y_{ij}, \ldots) = \sum_{cells} \left(\sum_{n \in \, St(c)} \frac{\| \; J_n^c - (J_{ref})_n^c \; \|_F^2}{\mid J_n^c \mid / \mid (J_{ref})_n^c \mid} \right) \tag{3.3}
$$

where J_n^c denotes the Jacobian at node n and cell c, and where the stencil $St(c)$ consists of all vertices (nodes) of the cell c. The domain of the discrete function includes both the interior and boundary nodes. We will constrain the boundary nodes to remain on the boundary of the physical domain.

The individual denominators of the objective function (3.3) each contains the determinant of a Jacobian matrix; therefore F will become infinite if any of these determinants vanish. It was shown in [44] that such an objective function will not produce folded elements provided that the initial guess of the iterative procedure has no folded elements. We refer to such an objective function as possessing a 'barrier' against inverted elements. In our rezoning algorithm, we will insure that the initial mesh is unfolded.

3.2 Numerical Optimization Algorithm

The minimization of the discrete objective function (3.3) belongs to the class of so-called *nonlinear least-squares* problems. There are well-developed numerical methods that deal with this class of optimization problems; however standard methods (see, for example, [37]) require modification to take into account the specifics of our problem.

First, we assume that the initial mesh is unfolded (i.e., $\mid J_n^c \mid > 0$) so that the optimization begins on the 'correct side' of the barrier. After each update of all nodal positions, we check that the new grid is valid (i.e., the areas of all triangles remain positive). When the update results in a folded mesh, the mesh is rejected, and the objective function is set to a large number with respect to some characteristic length. There are two criteria for ending the iteration: first, the iteration is stopped when there is no grid movement that will further reduce the value of objective function; second the iteration is stopped when the largest change in position of any grid node is smaller than some preset, user-supplied value. This algorithm for minimizing the objective function has proved robust. We have not investigated issues of computational efficiency.

In this framework, one can introduce constraints on the movement of internal points as well. This may be desirable, for example, when there is a material interface that should be maintained as a Lagrangian curve. In such a case, the algorithm would require a parametric description of the interface and instead of needing two coordinates, a node on interface will have just one parameter that enters the objective function. We will postpone this development to future work.

Below we present flow-charts that describe a minimization procedure that takes into account the specifics of our problem. We note that as yet there is no theoretical proof that a unique global minimum exists; however our numerical experiments support the conjecture that unique minima do exist.

3.2.1 Optimization Procedure

The basic optimization technique is a line search procedure coupled with a conjugate gradient algorithm to determine the search direction; see Chapters 3 and 5 of [37]. The optimization procedure is iterative, beginning with an initial (Lagrangian) mesh. The feasible region consists of the mesh node configurations that yield a valid 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 $F^0 = F(..., x^0_{i,j}, y^0_{i,j}, ...)$ on the initial mesh and the initial gradient $(\nabla F)^0$. The initial search direction is $p^0 = -(\nabla F)^0$ (for nodes on the boundary the search direction will be tangent to the boundary).

The basic steps in the iteration procedure are:

- increment the iteration counter k; halt if the maximum allowable count is exceeded ($\Rightarrow$ procedure failed to find a 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 ($\Rightarrow$ 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 $F^{k+1} = F(..., x_{i,j}^{k+1}, y_{i,j}^{k+1}, ...)$.

- 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

  $$\beta_{FR} = \mid (\nabla F)^{k+1} \mid^2 / \mid (\nabla F)^k \mid^2$$

  or Polak-Ribiere formulas [37].

  $$\begin{aligned} \beta_{PR} &= \beta_{FR} \\ &- (\nabla F)^{k+1} \cdot (\nabla F)^k / \mid (\nabla F)^k \mid^2 \end{aligned}$$

- go back to the start of this itemization.

3.2.2 Calculation of the Gradient

The evaluation of the objective function is straightforward, given the mesh. The gradient of the objective function is computed numerically. For each node (i, j) of the mesh one has coordinates $x_{i,j}, y_{i,j}$. The (i, j)-th component of the gradient is approximated by

$$\begin{aligned} \partial F/\partial x_{i,j} &= [F(..., x_{i,j} + \epsilon, y_{i,j}, ...) \\ &- F(..., x_{i,j}, y_{i,j}, ...)]/\epsilon \\ \partial F/\partial y_{i,j} &= [F(..., x_{i,j}, y_{i,j} + \epsilon, ...) \\ &- F(..., x_{i,j}, y_{i,j}, ...)]/\epsilon \end{aligned}$$

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.

3.2.3 The Line Search Algorithm

The inexact line search algorithm seeks 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 Numerical Examples

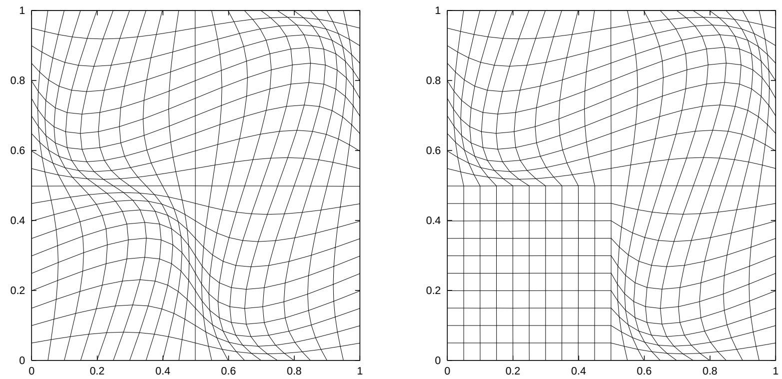
Here we offer three examples to illustrate the utility of our procedure. In the first example, we consider an RJM derived from a smooth map. In the second example, we consider an RJM derived from a map that is not smooth (i.e., is nondifferentiable). In the third example, we consider an RJM that is not derived from a map at all. In each case we use a variety of (valid) initial grids and show that the final grid is independent of this choice. In the first case, the minimal value of the objective function is zero and the final grid is that derived from the smooth map.

Let the smooth map from the unit square to the physical square be

$$x(\xi,\eta) = \xi + \varepsilon\,\sin(2\,\pi\,\xi)\,\sin(2\,\pi\,\eta)\,,\quad y(\xi,\eta) = \eta + \varepsilon\,\sin(2\,\pi\,\xi)\,\sin(2\,\pi\,\eta)\,. \tag{3.4}$$

where ε is a parameter that controls the deviation of the map from the identity. The map on the left in Figure 5 shows the grid derived from the map with $\varepsilon = 0.1$. The RJM is computed analytically by taking the appropriate derivatives of the map. The minimization of the objective function using this reference Jacobian and the three initial grids (a), (b), and (c) in Figure 6 leads consistently to the grid derived from the analytic map.

The second example illustrates the uniqueness of the final grid even when the map is not smooth (nondifferentiable) on a set of curves, regardless of the initial grid used in the



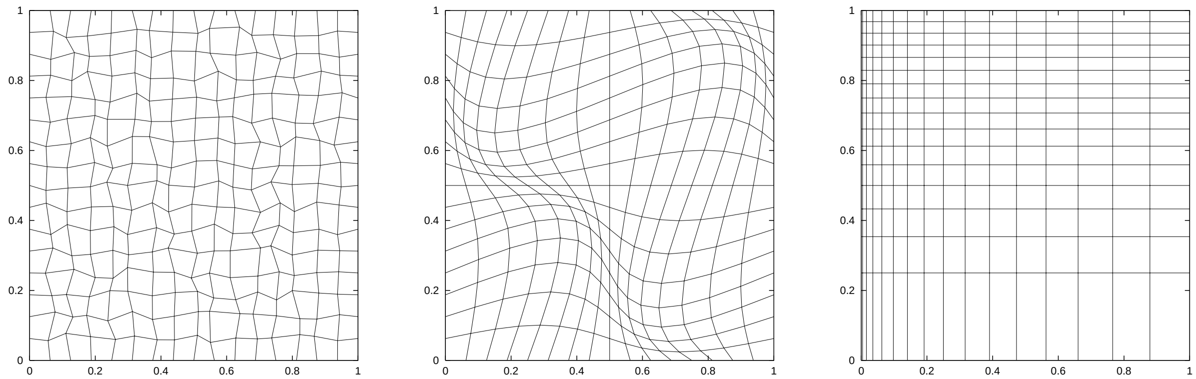
a) b)

Figure 5: *a) Smooth grid, b) Non-smooth grid*

minimization. The nonsmooth grid shown on the right in Fig. 5 is derived from the map that is identical to Equation (3.4) everywhere except on the square $[0 \leq \xi < 0.5] \times [0 \leq \eta < 0.5]$; on this sub-domain the map is the identity $x = \xi, y = \eta$. Although this map is not smooth, it is continuous.

On this grid one can easily derive analytic expressions for the derivatives x_ξ, x_η, y_ξ, y_η everywhere except on the line segments $\xi = 0.5, 0 \leq \eta < 0.5$ and $\eta = 0.5, 0 \leq \xi < 0.5$. Since the reference Jacobian is computed using cell edges, the derivatives exist where needed. When computing the RJM in the discrete case, these derivatives are computed at the middle of the edges using exact formulas and then multiplied by $\Delta\xi = 1/I\,, \Delta\eta = 1/J$ appropriately to obtain Δx and Δy along the edges. We use the same set of initial guesses as in the first example (see grids in Fig. 6) in the minimization process, with the result that the final grid is independent of the initial guess. The lack of sensitivity to the initial grid suggests that the discrete functional has a unique minimum.

In the third example, there is no map corresponding to the reference Jacobian matrix. The reference Jacobian matrix in this example was constructed by composing two maps. The first map is defined on the closed square centered at $\xi = 0.5\,, \eta = 0.5$ with side 0.5 and consists of the identity. The second map is defined by Equation 3.4 on the subdomain consisting of the unit square minus the closed inner square subdomain. The resulting map is



a) b) c)

Figure 6: *Different initial guesses a) Random grid, b) Smooth grid, c) Tensor product grid,*

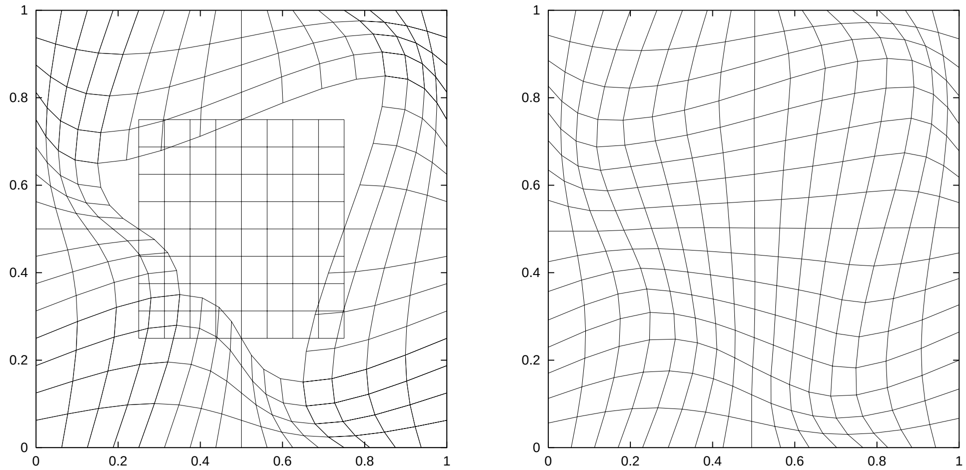
not continuous on the subdomain boundary (see Fig. 7 a)). The reference Jacobian matrix can still be constructed. Once again we use the three initial grids of Fig. 6. The final grid is shown in Figure 7 b); it is smooth and independent of the initial grid.

4 Construction of Reference Jacobian Matrices

Here we describe our local procedure for optimizing geometrical mesh quality and its properties, leading to the construction of the RJMs — stage I in Fig. 1. We present a heuristic connection to the Winslow grid generator.

On a 2D structured mesh, each interior node (i, j) is shared by four quadrilaterals (i.e., cells). The union of these quadrilaterals is an octagon, which we term the local patch of the central node (i, j) — stage I.1 in Fig. 1. There are four tangential vectors connecting the central node to its four nearest neighbors, $\mathbf{e}\xi_{i+\frac{1}{2},j}, \mathbf{e}\eta_{i,j+\frac{1}{2}}, \mathbf{e}\xi_{i-\frac{1}{2},j}$, and $\mathbf{e}\eta_{i,j-\frac{1}{2}}$. These tangential vectors, taken cyclically two at a time, define the four Jacobian matrices associated with the central node (cf. Fig. 8).

Now let us fix the exterior vertices of the octagon in their Lagrangian positions, while allowing the position of the central node to vary. This "virtual" displacement alters each of



a) b)

Figure 7: *a) Discontinuous map, b) Unique grid obtained by optimization*

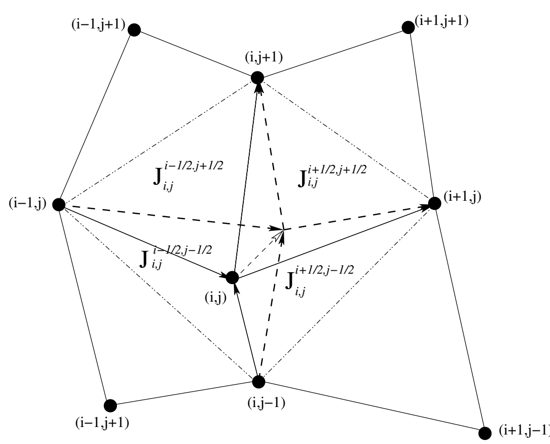


Figure 8: *Virtual movement of central node, and construction of four RJM's. Lagrangian nodes are marked by filled circles.*

the tangential vectors and also the Jacobians associated with the central node, leading to the reference Jacobians.

At this point, we would like to emphasize certain points. First, it is important to note that the virtual displacement at any node does not affect the patch geometry of any other node; hence it cannot affect the reference Jacobians at any other node either. Second, many strategies may be used to define the virtual displacement. Third, possibly excepting the particular choice of strategy, nothing else in this framework of defining the reference Jacobians depends on the grid being structured or 2D.

Based on the previous comments, it is desirable to choose a strategy to displace the central node that also is (or can be generalized to be) independent of whether the grid is structured. This will facilitate the extension of all the results in this paper to unstructured grids. We describe such a strategy next.

4.1 Exposition of Winslow Algorithm and Its Variational Version

To set the stage, we briefly review and interpret the classical approach to grid generation based on Winslow's original paper [45]. Suppose we want to construct a logically rectangular grid in a 2D domain where we can distinguish left, right, top and bottom boundaries. We introduce dependent functions $\xi(x, y)$ and $\eta(x, y)$ (one might think of them as temperatures) which are solutions of corresponding stationary heat equations (Laplace equations):

$$\nabla^2, \xi = 0\,, \quad \nabla^2\, \eta = 0$$

with following boundary conditions

$$\xi|_{(x,y)\in\Gamma_L} = 0,\ \ \xi|_{(x,y)\in\Gamma_R} = 1\,; \quad \xi\,|_{(x,y)\in\Gamma_B} = \xi_B(x, y),\ \ \xi\,|_{(x,y)\in\Gamma_T} = \xi_T(x, y)\,;$$

and

$$\eta\,|_{(x,y)\in\Gamma_L} = \eta_L(x, y),\ \ \eta\,|_{(x,y)\in\Gamma_R} = \eta_R(x, y)\,; \quad \eta|_{(x,y)\in\Gamma_B} = 0, \quad \eta|_{(x,y)\in\Gamma_T} = 1\,.$$

Here, for example, $\xi_L(x,y)$ is a given function whose value monotonically increases from zero to one moves on the left boundary from the bottom to the top boundary; the functions $\xi_R(x,y)$, $\eta_B(x,y)$ and $\eta_T(x,y)$ vary in a similar way.

Because of the maximum principle, the maximum and minimum values of $\xi(x,y)$ and $\eta(x,y)$ lie on the boundary; that is, all values inside the domain must lie between zero and one. The solution consists of two sets of equipotential curves, one set connecting the top and bottom boundaries, the other connecting the left and right boundaries. Being equipotentials, there is no intersection of the curves within either set. Now consider the two particular subsets of equipotentials, $\xi_i(x,y) = i/I\,, i = 0, 1, \ldots, I$, and $\eta_j(x,y) = j/J\,, j = 0, 1, \ldots, J$. The intersections of these two sets of curves can be chosen as the nodes of a logically rectangular grid. Furthermore, since ξ and η are the solutions of Laplace equations, they are smooth functions.

As a practical technique to implement these ideas, Winslow in his pioneering paper [45] suggested that one invert the choice of dependent and independent variables, leading to a following system of quasilinear elliptic equations for $x(\xi,\,\eta)\,, y(\xi\,, \eta)$

$$g_{22}\, x_{\xi\xi} - 2\, g_{12}\, x_{\xi\eta} + g_{11}\, x_{\eta\eta} = 0\,, \quad g_{22}\, y_{\xi\xi} - 2\, g_{12}\, y_{\xi\eta} + g_{11}\, y_{\eta\eta} = 0\,, \tag{4.1}$$

where

$$g_{11} = x_\xi^2 + y_\xi^2\,,\;\; g_{12} = x_\xi\, x_\eta + y_\xi\, y_\eta\,,\;\; g_{22} = x_\eta^2 + y_\eta^2\,, \tag{4.2}$$

This system of equations is discretized on the rectangular grid in (ξ, η) space, and solved numerically to find the positions of the grid nodes $x_{i,j}$ and $y_{i,j}$ (see Section 3.3, for more details).

It is well known (see for example [43]) that Laplace's equation for ξ (as well as for η) has a variational formulation:

$$\min_\xi F_\xi;\; F_\xi(\xi) = \int_V (\mathbf{grad}\,\xi)^2\, dV = \int_V \left\{(\xi_x)^2 + (\xi_y)^2\right\}\, dV \tag{4.3}$$

$$\min_\eta F_\eta;\; F_\eta(\eta) = \int_V (\mathbf{grad}\,\eta)^2\, dV = \int_V \left\{(\eta_x)^2 + (\eta_y)^2\right\}\, dV\,. \tag{4.4}$$

This fact was first used in the framework of grid generation by Brackbill and Saltzman [6], where the functionals F_ξ and F_η were combined and the resulting functional named the "smoothness" functional:

$$F(\xi, \eta) = \int_V \left\{ (\xi_x)^2 + (\xi_y)^2 + (\eta_x)^2 + (\eta_y)^2 \right\} dV \, . \tag{4.5}$$

The minimization of this functional is analytically equivalent to the solution of the two Laplace equations for ξ and η. We next invert the variables in the smoothness functional using the well-known relations

$$\xi_x = y_\eta / |J| \, , \xi_y = -x_\eta / |J| \, , \tag{4.6}$$

$$\eta_x = -y_\xi / |J| \, , \eta_y = x_\xi / |J| \, , \tag{4.7}$$

where

$$J = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix}$$

is the Jacobian matrix of map $x(\xi, \eta), y(\xi, \eta)$ and $|J|$ is its determinant. Using these relations and that $dV = |J| \, d\xi d\eta$, we can rewrite the smoothness functional as follows:

$$F(x, y) = \int_0^1 \int_0^1 \frac{[(x_\xi)^2 + (y_\xi)^2] + [(x_\eta)^2 + (y_\eta)^2]}{|J|} \, d\xi d\eta \, . \tag{4.8}$$

One very important feature that explicitly can be seen in the variational formulation is that this functional has a *barrier*, i.e., the value of the functional approaches infinity when the unknown map approaches any degenerate map where $|J| = 0$. This observation explains how the Winslow grid generator produces unfolded grids [13].

Using the Euclidean (Frobenius) norm of a matrix ($||A|| = \left(\sum_{k=1}^2 \sum_{l=1}^2 a_{k,l}^2 \right)^{\frac{1}{2}}$) one can write the smoothness functional 4.8 in the form [30]

$$F(x, y) = \int_0^1 \int_0^1 \frac{||J||^2}{|J|} \, d\xi d\eta \, . \tag{4.9}$$

In 2D this expression also can be interpreted in the framework of a general theory based on matrix norms and condition number [29]. In fact, the condition number of the Jacobian matrix is

$$k(J) = ||J|| \cdot ||J^{-1}|| \, . \tag{4.10}$$

Now in 2D

$$J^{-1} = \frac{1}{|J|} \begin{pmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{pmatrix}$$

so that

$$||J^{-1}|| = \frac{1}{|J|} \, ||J|| \, , \quad \text{and} \quad k(J) = \frac{||J||^2}{|J|} \, . \tag{4.11}$$

Therefore $k(J)$ is the same as the integrand in (4.9). In 3D, the condition number cannot be written in this way and the minimization of a functional based on the integral of condition number yields a different way to construct the grid, which has some theoretical and practical advantages [29].

There is a clear geometrical interpretation for the expression $||J||^2/|J|$, which appears in the smoothness functional. We recall that the Jacobian matrix is formed from the vectors tangential to the coordinate lines (Fig. 3 and formula (2.1)). Thus $||J||^2 = |\mathbf{e}_\xi|^2 + |\mathbf{e}_\eta|^2$ is the sum of the squares of the lengths of the elementary tangential vectors, and $|J| = \mathbf{e}_\xi \times \mathbf{e}_\eta$ is the area of elementary parallelogram constructed from the tangential vectors. That is,

$$F(x,y) = \int_0^1 \int_0^1 \frac{|\mathbf{e}_\xi|^2 + |\mathbf{e}_\eta|^2}{\mathbf{e}_\xi \times \mathbf{e}_\eta} \, d\xi d\eta \, . \tag{4.12}$$

Now let us consider the discrete case. Using the definition (3.1) of the discrete tangential vectors, a discrete analog of (4.12) can be written as

$$F^h = \frac{1}{2} \sum_{(i,j) \in \text{nodes}} \left(\sum_{k,l=\pm\frac{1}{2}}^{1} \frac{|\mathbf{e}\xi_{i+k,j}|^2 + |\mathbf{e}\eta_{i,j+l}|^2}{\mathbf{e}\xi_{i+k,j} \times \mathbf{e}\eta_{i,j+l}} \right) . \tag{4.13}$$

The interior sum contains only two terms corresponding to nodes on the boundary, and only one term for the corner nodes. The factor 1/2 appears in the discrete functional because each quadrilateral is covered twice.

Relying on the geometrical interpretation of the terms participating in this functional, we write

$$F^h = \sum_{(i,j) \in \text{nodes}} \left(\sum_{k,l=\pm\frac{1}{2}}^{1} \frac{l\xi_{i+k,j}^2 + l\eta_{i,j+l}^2}{A_{i,j}^{i+k,j+l}} \right) , \tag{4.14}$$

where $l\xi$ and $l\eta$ are the lengths of the corresponding sides of the triangles, and A is the area of the triangle. We will call this the global Winslow functional. When this global functional is restricted to a patch, with only the central node being allowed to vary, we will call this the local Winslow functional.

4.2 Local Mesh Quality Optimization

We will use the local realization of the Winslow functional to choose the new virtual position of the central node of a patch — stage I.2 in Fig. 1. We group those terms in (4.14) that depend on the position of the central node (i,j) – i.e., terms related to the twelve triangles

$$
\begin{aligned}
&T_{i,j}^{i+\frac{1}{2},j+\frac{1}{2}},\ T_{i,j}^{i-\frac{1}{2},j+\frac{1}{2}},\ T_{i,j}^{i-\frac{1}{2},j-\frac{1}{2}},\ T_{i,j}^{i+\frac{1}{2},j-\frac{1}{2}}; \\
&T_{i+1,j}^{i+\frac{1}{2},j+\frac{1}{2}},\ T_{i+1,j}^{i+\frac{1}{2},j-\frac{1}{2}};\ T_{i-1,j}^{i-\frac{1}{2},j+\frac{1}{2}},\ T_{i-1,j}^{i-\frac{1}{2},j-\frac{1}{2}};\ T_{i,j+1}^{i+\frac{1}{2},j+\frac{1}{2}},\ T_{i,j+1}^{i-\frac{1}{2},j+\frac{1}{2}};\ T_{i,j-1}^{i+\frac{1}{2},j-\frac{1}{2}},\ T_{i,j-1}^{i-\frac{1}{2},j-\frac{1}{2}};
\end{aligned}
\tag{4.15}
$$

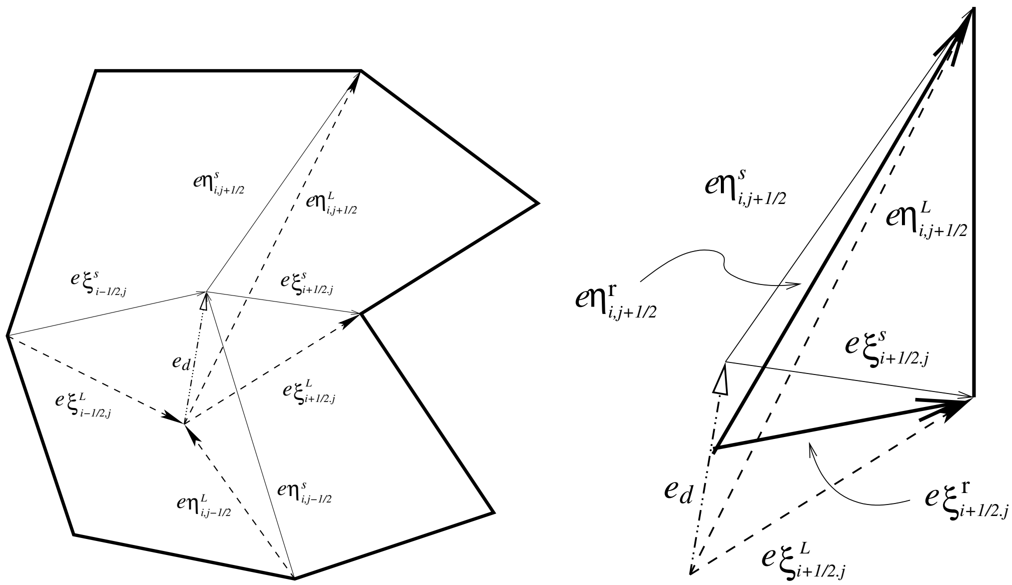
to form the local functional

$$
\begin{aligned}
F_L^h(x_{i,j}^s, y_{i,j}^s) = \sum_{k,l=\pm\frac{1}{2}}^{1} \frac{l\xi_{i+k,j}^2 + l\eta_{i,j+l}^2}{A_{i,j}^{i+k,j+l}} + \\
\sum_{l=\pm\frac{1}{2}}^{1} \frac{l\xi_{i+\frac{1}{2},j}^2 + l\eta_{i+1,j+l}^2}{A_{i+1,j}^{i+\frac{1}{2},j+l}} + \sum_{l=\pm\frac{1}{2}}^{1} \frac{l\xi_{i-\frac{1}{2},j}^2 + l\eta_{i-1,j+l}^2}{A_{i-1,j}^{i-\frac{1}{2},j+l}} + \\
\sum_{k=\pm\frac{1}{2}}^{1} \frac{l\xi_{i+k,j}^2 + l\eta_{i,j+\frac{1}{2}}^2}{A_{i,j+1}^{i+k,j+\frac{1}{2}}} + \sum_{k=\pm\frac{1}{2}}^{1} \frac{l\xi_{i+k,j}^2 + l\eta_{i,j-\frac{1}{2}}^2}{A_{i,j-1}^{i+k,j-\frac{1}{2}}}.
\end{aligned}
\tag{4.16}
$$

The optimized position $x_{i,j}^s$, $y_{i,j}^s$ is determined by (locally) minimizing this local functional of two variables (the coordinates of the central node) with all other coordinates held fixed at their Lagrangian values.

We define a feasible set $\mathcal{F}$ consisting of all virtual positions of the central point that ensure positive areas of all twelve triangles (4.15). Note that when the patch — the octagon of four cells surrounding the central node — is convex, the feasible set consists of the diamond formed by vertices $(i+1,j)$, $(i,j+1)$, $(i-1,j)$, $(i,j-1)$. More generally, the feasible set is the intersection of twelve half-planes, each being defined by the side of the triangle opposite the central node; it is a convex polygon and is not empty because $\mathbf{x}_{i,j}^L \in \mathcal{F}$ by assumption.

Clearly, if we use the optimization procedure described in Section 3.2 and use $\mathbf{x}_{i,j}^L$ as the initial guess, then the point $\mathbf{x}_{i,j}^s = (x_{i,j}^s, y_{i,j}^s)^T$ that leads to the minimum of the local



a) b)

Figure 9: *Local Smoothing, a) Lagrangian and smoothed tangential vectors of the four RJMs at node* (i,j), $\mathbf{e}_d$ *is the displacement vector connecting Lagrangian and smoothed central nodes, b) Reference tangential vectors* $\mathbf{e}\xi^r_{i+\frac{1}{2},j}$, $\mathbf{e}\eta^r_{i+\frac{1}{2},j}$ *that define the Reference Jacobian matrix* $J^{i+\frac{1}{2},j+\frac{1}{2}}_{i,j}$.

functional belongs to feasible set, $\mathbf{x}^s_{i,j} \in \mathcal{F}$. In addition, because the feasible set is a convex polygon, the points $\mathbf{x}^r_{i,j}$

$$\mathbf{x}^r_{i,j} = (1-\alpha_{i,j})\,\mathbf{x}^L_{i,j} + \alpha_{i,j}\,\mathbf{x}^s_{i,j} \in \mathcal{F}\,, \quad 0 \le \alpha_{i,j} \le 1\,.$$

are all in the feasible set. That is, any point on the line segment connecting the Lagrangian and smoothed central points is a admissible displacement (see Fig. 9). The same idea can be expressed in terms of the reference tangential vectors; if we define

$$\mathbf{e}\xi^r_{i+\frac{1}{2},j} = \mathbf{x}^L_{i+1,j} - \left\{(1-\alpha_{i,j})\,\mathbf{x}^L_{i,j} + \alpha_{i,j}\,\mathbf{x}^s_{i,j}\right\}\,,\;\; \mathbf{e}\eta^r_{i,j+\frac{1}{2}} = \mathbf{x}^L_{i,j+1} - \left\{(1-\alpha_{i,j})\,\mathbf{x}^L_{i,j} + \alpha_{i,j}\,\mathbf{x}^s_{i,j}\right\}\,,$$

then the RJM in the corresponding triangle is (stage I.3 in Fig. 1)

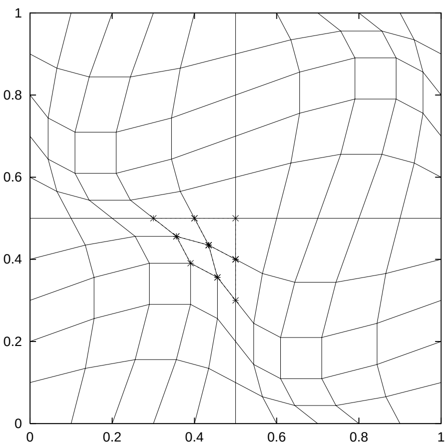
$$(J_{ref})^{i+\frac{1}{2},j+\frac{1}{2}}_{i,j} = \left((\mathbf{e}\xi^r)_{i+\frac{1}{2},j}\,|\,(\mathbf{e}\eta^r)_{i,j+\frac{1}{2}}\right)\,.$$

The parameter $\alpha_{i,j}$ can be used to control how much we want to smooth the Lagrangian grid and so represents an additional degree of freedom that one can exploit. For example, one might use $\alpha_{i,j}$ to build in information about the density field and so prevent the rezoner from undoing mass matching. For the remainder of this paper, however, we will assume $\alpha_{i,j} = 1$.

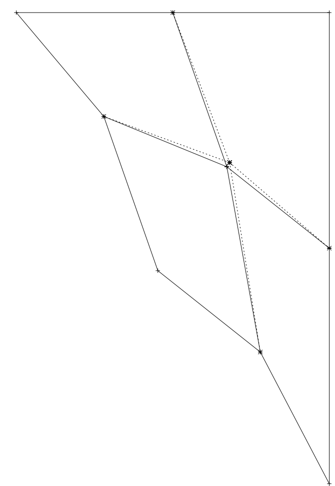
To demonstrate the behavior of the local optimization algorithm (I.2 in Fig. 1), we consider again the smooth grid shown in Fig. 10 a) and the nonsmooth grid shown in Fig. 10 c) (see detailed description of these grids in Section 3) and isolate the particular patches that are marked by "$*$" in these figures. In Fig. 10 b) and d) these patches are enlarged with solid lines. The dotted lines in 10 b) and d) show the new reference tangential vectors. Note that in the the smooth grid, the virtual displacement is small, whereas in the nonsmooth grid, the virtual displacement is much larger.

To summarize, the virtual position of the central node $\mathbf{x}^s_{i,j}$ as determined by minimizing (4.16) retains memory of the Lagrangian grid because the positions of all vertices except the central node that appear in the local functional are Lagrangian positions. That the virtual displacement contributes to a smoother grid can be seen by its connection to the Winslow functional. However our formulation (unlike the Winslow formulation) is not restricted to structured grids.

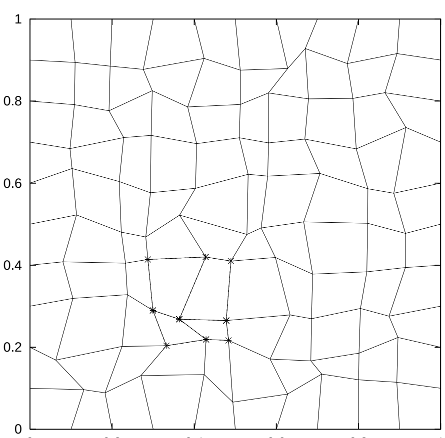
It is clear that this approach for constructing the RJM is readily extended to unstructured grids and to 3D. The role of the patch (the octagon on a 2D structured grid) will be played by the union of all cells that share the central node. The RJMs will be still defined in triangles, or in tetrahedra in 3D, even for unstructured grids. The local functional will contain all triangles (or tetrahedra) for which the RJM depends on the position of the central node. For a 2D unstructured grid, each term in the local functional has the same form as in the structured grid. In 3D for a tetrahedral or hexahedral grid, each term in the local based on Winslow functional will be the sum of the areas of the three faces divided by the volume of corresponding tetrahedron; alternately, one can use the functional based on the condition number, which has the advantage that in even in 3D it is dimensionless.



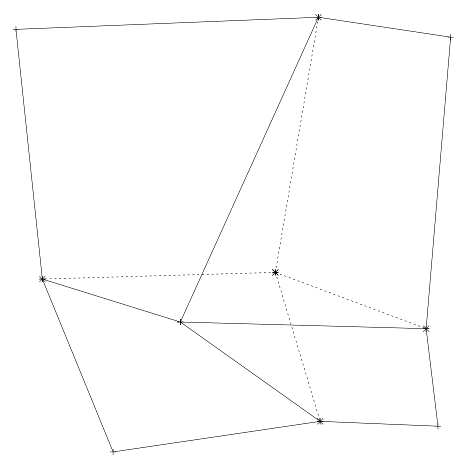
a)



b)



c)



d)

Figure 10: *Examples of Local Smoothing*

For general unstructured grids in 3D, there are other extensions of the proposed functional, which will be explored in future work.

5 Examples of Rezoned Grids

Having the RJMs at each node, we can use the algorithm described in Section 3 to construct the rezoned grid. We begin by considering the Lagrangian grids introduced in the previous section (Fig. 10 a), and Fig. 10 c)).

In Fig. 11 we compare the Lagrangian and rezoned grids. As one might expect, when the Lagrangian grid is smooth (Fig. 11 a)), the rezoned grid is very similar. There are some differences on the boundary, because our procedure effectively uses one-sided estimates for the reference tangential vectors on the boundary. As we increase the resolution, the discrepancies on the boundary and in the interior become less noticeable. In the case of a random grid (Fig. 11 b)), the rezoned grid is significantly different from the Lagrangian grid. The rezoned random grid is much smoother than, but (in an average sense) still close to the Lagrangian grid.

Next we demonstrate how the rezone procedure works when the Lagrangian grid consists of subdomains and where the grid is smooth within the subdomains, but is not smooth on the interface between subdomains. We revisit the grid from the previous Section (Fig. 5 b)). To make effect of smoothing more evident, we consider a coarser (11×11) grid. In Fig. 12 we superpose the original and smoothed grids. These results show that if original grid is smooth enough, then the rezoner has little effect. The main result of smoothing is to create a smooth transition zone between the subdomains whose Jacobians in the original map are different.

Finally we consider a more extreme example, shown in Fig. 13 a). This is called the *Shestakov* grid [41], and has been used in many papers to test performance and robustness of finite difference and finite element schemes. The rezoned grid shown in Fig. 13 b) is noticeably smoother.

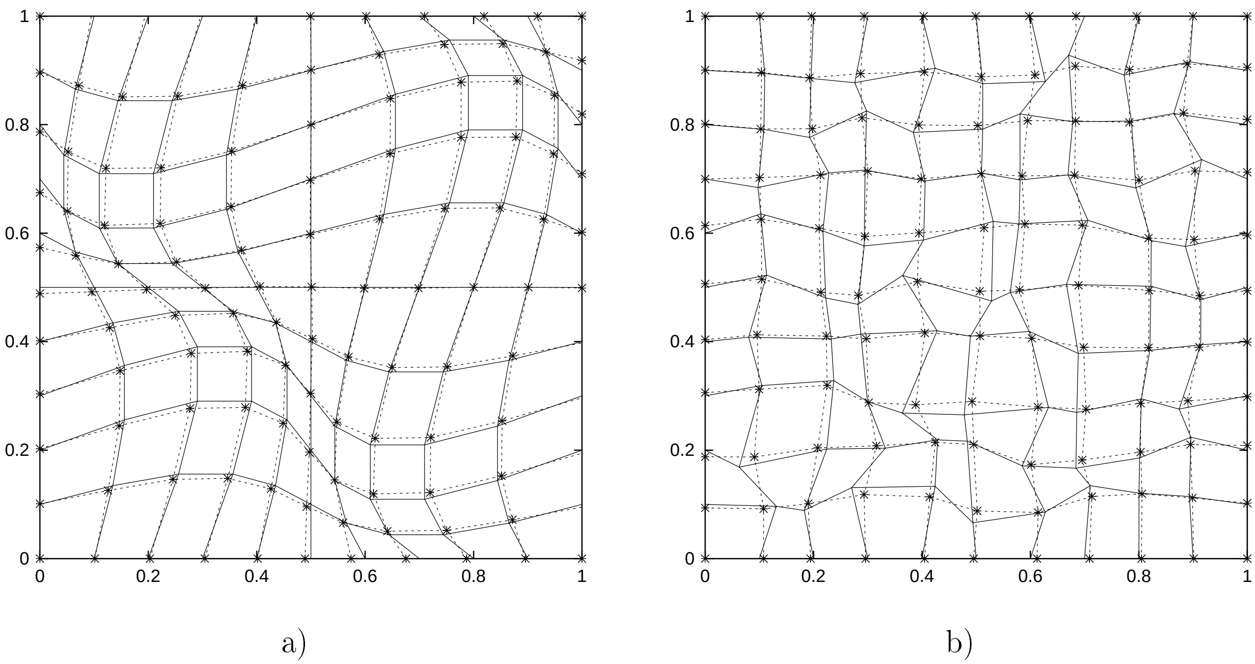


Figure 11: *Comparison of Lagrangian and rezoned grids, a) Smooth Lagrangian grid and corresponding rezoned grid, b) Nonsmooth (random) Lagrangian grid and corresponding rezoned grid*

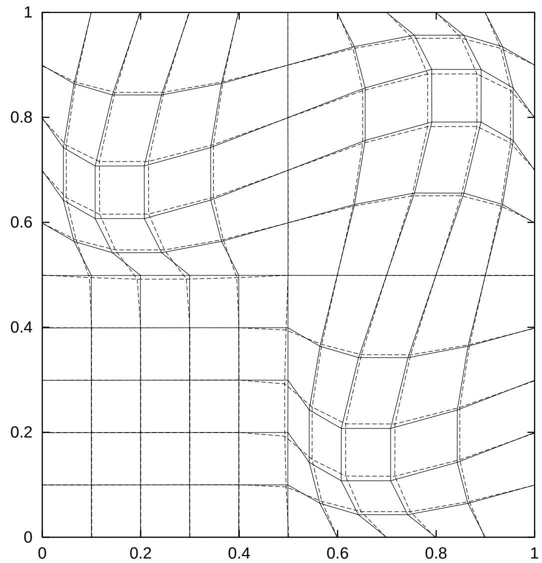


Figure 12: *Comparison of Lagrangian and rezoned grids, when the Lagrangian grid consists of subdomains and where the grid is smooth within the subdomains, but is not smooth on the interface between subdomains.*

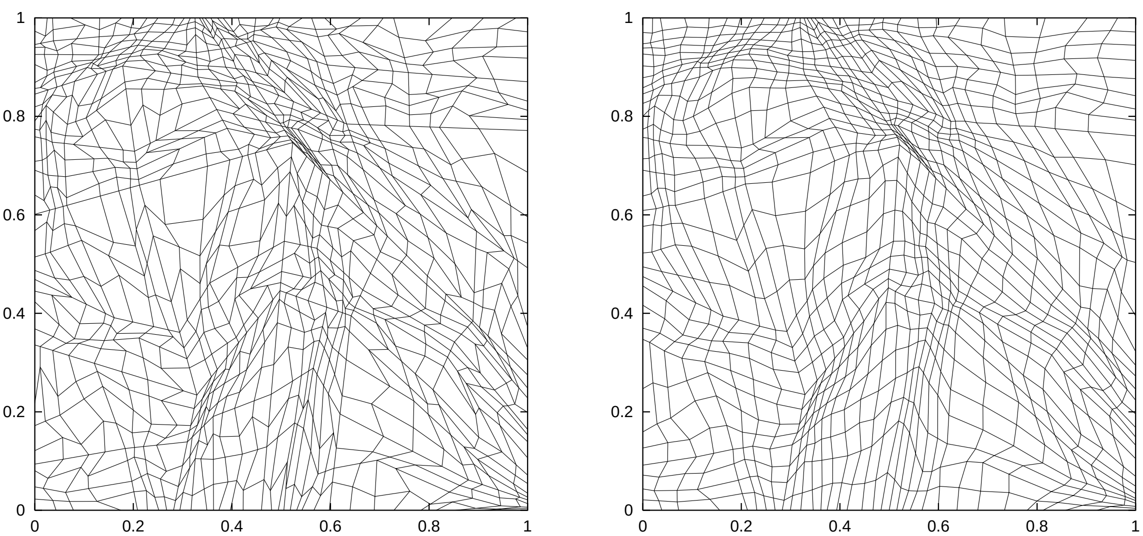
6 Preliminary ALE Calculations

In this Section we describe a preliminary ALE calculation of the Rayleigh-Taylor instability problem, using our new RJM rezone strategy. We say preliminary because a complete ALE algorithm involves many more issues than just the rezoning strategy described in this paper. In the next subsection we briefly describe the Lagrangian phase and the remapper used in our calculation, whose results are presented in subsection 6.3.

We also compare our ALE calculation with one obtained by using rezone strategy based on Winslow smoothing procedure.

6.1 Lagrangian Phase and Remapping

The explicit Lagrangian phase uses a state-of-the-art methodology described in a recent series of papers [9, 10, 11, 7]. It employs compatible differencing techniques [11], which ensure a conservative finite difference method (i.e., mass, momentum, and total energy are conserved



a) b)

Figure 13: *Shestakov grid and its smoothed version.*

to roundoff error); a new tensor artificial viscosity [10]; and, what is most important here, subzonal forces that are designed to reduce nonphysical grid movement [9]. The Lagrangian method uses a standard staggered mesh discretization for flow variables on a logically rectangular grid: coordinates and velocity components are located on vertices (which we will call later points) and the thermodynamic variables (density, pressure, and internal energy) are associated with cell centers, [11].

In the subzonal forces approach, one first introduces subzones [Fig. 14 a)] formed by connecting the centers of opposite sides of the cell by straight lines. These lines intersect in geometrical center of the cell. The quadrilaterals formed this way are called subzones. Each subzone has its own Lagrangian mass and volume, and therefore density. The internal energy density in each subzone is taken equal to that of the parent zone. The difference between the subzonal density and that of the parent zone is used to construct subzonal forces. (see [9] for details). The subzonal forces oppose unphysical grid motions.

It is important to note that in a purely Lagrangian simulation, the unphysical grid velocities are made manifest by the accompanying grid distortion. However even in ALE simulations where the distortion can be mitigated by rezoning, these modes may still contaminate the velocity field and so must be removed.

A typical example is termed an “hourglass” pattern and is shown in Fig. 14 b). This pattern of velocity does not change the volume of the cells; in conventional Lagrangian methods there is no way to produce restoring forces to oppose this motion. However the volumes of the individual subzones do change with hourglassing, and these changes are used to generate the subzonal pressure forces that combat the growth of the hourglass patterns. The subzonal force approach has proved very successful in many problems [9], but its interaction with other physical motions (vorticity, for example), is not very well understood. In particular, we will show that the subzonal forces used in a purely Lagrangian simulation of Rayleigh-Taylor instability allows calculation to survive, but the results unfortunately are not accurate. However in the framework of ALE one can use the subzonal forces, because

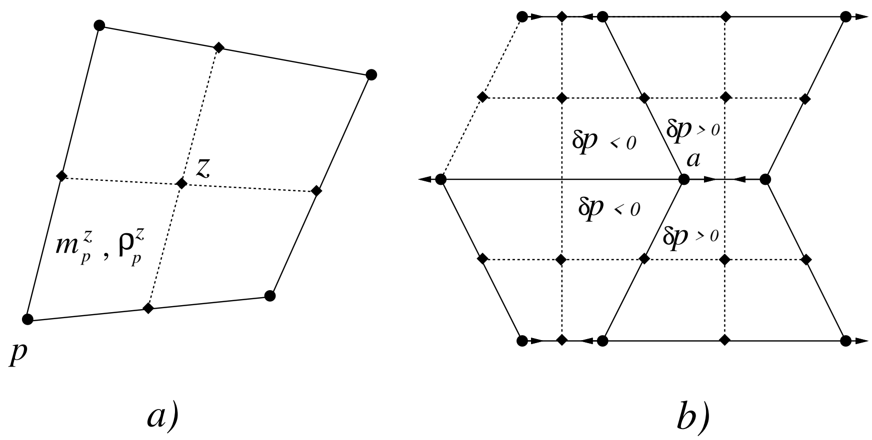


Figure 14: *Subzonal forces. a) Subzones, density and mass specified in subzones, p is the index of vertex (point) and z is index of the zone, b) Hourglass motion and corresponding change in subpressures preventing developing of this motion.*

our rezoning strategy can in particular mitigate the hourglass mesh distortion, thus reducing the magnitude of the subzonal forces.

Remapping in our ALE calculations is done using the cell-centered algorithm of Dukowicz and Kodis [16], with modifications necessitated by our staggered grid (i.e., momentum is not a cell-centered quantity) and by the subzonal forces (i.e., one must remap subzonal masses). Our preliminary strategy, to enable the ALE calculations presented later in this section, is to define all quantities (including momentum) in the subzonal centers, and then to apply the Dukowicz – Kodis algorithm directly to the subzones. The final step is to use the conservation laws to group the subzones appropriately, either around the vertices or the cell centers, to restore velocity to the nodes, and internal energy and density to the cell centers.

6.2 Rezoning Strategy Based on Winslow Smoothing Method

In this subsection we describe one possible rezone strategy based on direct use of Winslow smoothing method and point out potential problems related to this approach. We will show results employing this strategy in ALE calculations in the next section.

First we introduce a standard discretization of the Winslow equations (4.1), (4.2)—(see, for example, [39]). The first and second derivatives of x with respect to ξ and η, which appear in the Winslow equations, are approximated using central differences

$$\begin{aligned}
&(x_\xi)_{i,j} \approx \frac{x_{i+1,j} - x_{i-1,j}}{2\,\Delta\xi}\,, \quad (x_\eta)_{i,j} \approx \frac{x_{i,j+1} - x_{i,j+1}}{2\,\Delta\eta}\,,\\
&(x_{\xi\xi})_{i,j} \approx \frac{x_{i+1,j} - 2\,x_{i,j} + x_{i-1,j}}{\Delta\xi^2}\,, \quad (x_{\eta\eta})_{i,j} \approx \frac{x_{i,j+1} - 2\,x_{i,j} + x_{i,j-1}}{\Delta\eta^2}\,,\\
&(x_{\xi\eta})_{i,j} \approx \frac{x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}}{4\,\Delta\xi\,\Delta\eta}\,,
\end{aligned} \tag{6.1}$$

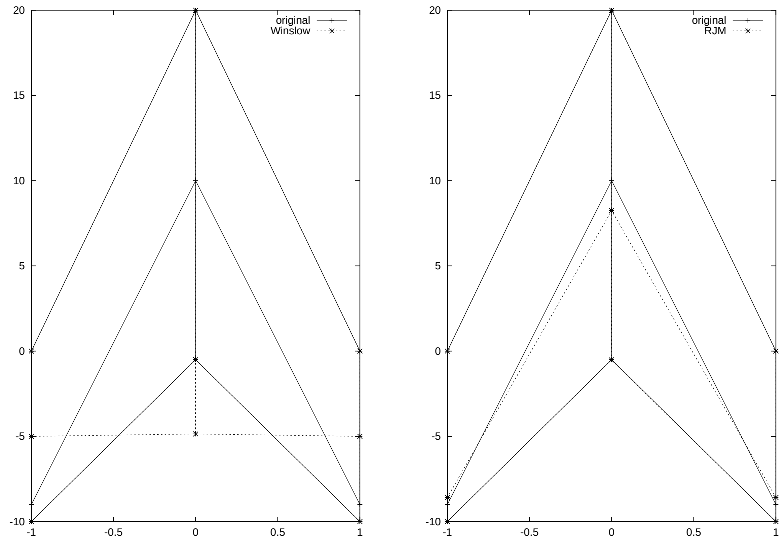
and similar formulas for the derivatives of y.

Now let us assume that we are given a grid with coordinates $\left(x^k_{i,j},\, y^k_{i,j}\right)$ (where k refers to the iteration number); we denote the coefficients g_{22}, g_{12}, g_{11} computed from these coordinates and the approximate formulas 6.1 by α^k, β^k and γ^k correspondingly. Then one iteration of Winslow smoothing (see, [39]) is

$$\begin{aligned}
&x^{k+1}_{i,j} = \frac{1}{2\,(\alpha^k + \gamma^k)}\;\cdot\\
&\left(\alpha^k\,(x^k_{i,j+1} + x^k_{i,j-1}) + \gamma^k\,(x^k_{i+1,j} + x^k_{i-1,j}) - \frac{1}{2}\,\beta^k\,(x^k_{i+1,j+1} - x^k_{i-1,j+1} + x^k_{i-1,j-1} - x^k_{i+1,j-1})\right)\\
&y^{k+1}_{i,j} = \frac{1}{2\,(\alpha^k + \gamma^k)}\;\cdot\\
&\left(\alpha^k\,(y^k_{i,j+1} + y^k_{i,j-1}) + \gamma^k\,(y^k_{i+1,j} + y^k_{i-1,j}) - \frac{1}{2}\,\beta^k\,(y^k_{i+1,j+1} - y^k_{i-1,j+1} + y^k_{i-1,j-1} - y^k_{i+1,j-1})\right)
\end{aligned} \tag{6.2}$$

The authors of [39] suggest using a few iterates based on the formulas (6.2), starting with the Lagrangian grid as an initial guess rathan than iterating to convergence. They further suggest restricting the rezone to some set of "marked" nodes; to choose the "marked" nodes, they use an *ad hoc* criterion based on the comparison of angles and areas of the cells surrounding these nodes.

As a simple example, we consider the "chevron" grid shown in Fig. 15 in bold lines, consisting of only four cells. Here, we allow only the central node of the grid and the central nodes of the left and right boundaries to move, thus preserving the integrity of the shape of the domain. For the nodes on the boundary, the 1D analog of the Winslow smoothing



a) b)

Figure 15: *Comparison of Winslow and and RJM rezone strategies for "chevron" grid: a) Winslow, b) RJM*

step is to choose the new node position halfway between the neighboring nodes. After two iterations of (6.2) we get the folded grid shown in Fig. 15 a) in dotted lines. It is important to understand that this is not a contrived example, but may represent a small patch taken from a much larger grid where the nodes on the boundary of the patch have not been "marked". In contrast, our RJM rezone strategy produces the grid that is shown in Fig. 15 b). Clearly the RJM rezone has preserved the essence of the original Lagrangian grid while improving the overall grid quality.

The failure of the Winslow strategy in the above example depends mainly on the use of "marked" grids. To reduce the likelihood of an invalid grid, in the next section we will apply the rezone to all nodes using formulas 6.2. We will further restrict ourselves to using two iterations. We will call this approach our Winslow strategy, without repeating the disclaimer that this implementation is not unique. We also remind the reader that this particular implementation may not exactly correspond to that used in any other ALE codes.

However, it is a simple implementation that captures the spirit of rezoning based on Winslow smoothing.

6.3 Preliminary ALE Results for Rayleigh-Taylor Instability

The Rayleigh-Taylor instability problem that we simulate here consists of two ideal gases with densities $\rho_h = 2$ and $\rho_l = 1$; in both cases the adiabatic constant $\gamma = 1.4$. 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 and with magnitude $g = 0.1$. The interface has been deliberately perturbed as described by the formula $y_i(x) = \frac{1}{2} + A \cos(6\,\pi\,x),\ \ A = 0.01$. Initially both gases are at rest (velocity is zero everywhere); the pressure distribution is approximately hydrostatic and is defined in the lighter gas:

$$P = 1 + \rho_h\, g\, 0.5 + \rho_l\, g\, (0.5 - y)$$

and in the heavier gas:

$$P = 1 + \rho_h\, g\, (1 - y)\,.$$

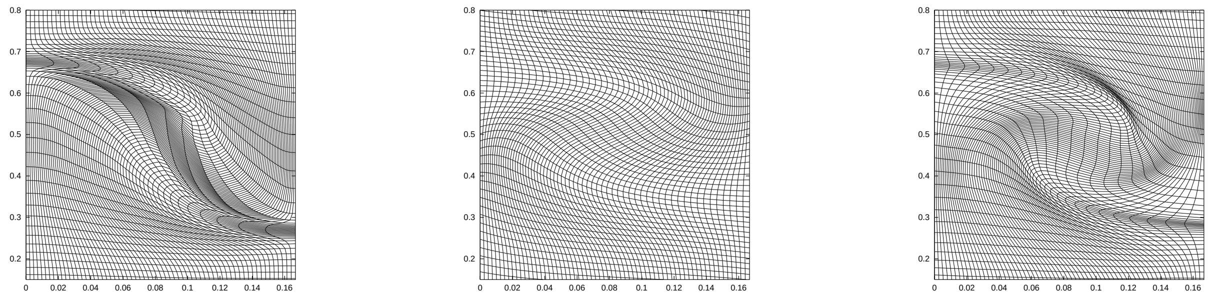
It is well known that this configuration is unstable and as time progresses, the heavier gas will sink and the lighter gas will rise through the formation of bubbles and spikes. Further details of the general theory of Rayleigh-Taylor instabilities can be found in [31].

The time evolution of this problem leads to a rollup of the interface and the generation of significant vorticity. Thus this problem is poorly suited for Lagrangian methods, and is usually tackled using Eulerian techniques. Our goal here is not to suggest that ALE codes can supplant Eulerian codes, especially for the late stages of the evolution. Rather our goal is to demonstrate how the ALE methodology with RJM rezone strategy can improve robustness, accuracy, and computational efficiency over purely Lagrangian methods, and also over an alternate ALE method that uses the simple Winslow rezone strategy described in the previous section. To this end, we will present and compare four simulations here: a purely Lagrangian simulation, an ALE simulation using Winslow rezone strategy, an ALE simulation using the RJM rezone strategy described in this paper, and an Eulerian simulation with front tracking. The latter, using FronTier [20, 21], is much more highly resolved and is offered as a reference solution to allow the comparison of Lagrangian and ALE simulations.

The grid dimensions for both the Lagrangian and ALE calculations are 65×65. The initial grid consists of lines parallel to y axis (i=const); on each such vertical line, half the points are equispaced between the interface and the top boundary and the other half are equispaced between the interface and the bottom boundary. The width of the simulation domain is 1/6 its height, so that the zones have an initial aspect ratio of $1 : 6$. The reason for this choice is that at the times for which we present results ($t = 7.0$ and $t = 8.5$), the zones near the interface attain an aspect ratio of approximately $1 : 1$, which improves the accuracy and robustness of the simulation at the late stages. Indeed, for a simulation with relatively large distortion such as the Rayleigh-Taylor problem and when the connectivity of the grid is fixed in time, the choice of the initial grid can make a noticeable difference in the quality of the final results. However our experiments with different initial grids show that the results of the Lagrangian calculations cannot be improved significantly. For example, using an initial grid with mass matching across the interface does not improve the final shape of the interface.

The strength of the subzonal pressure forces is controlled by parameter called a "merit factor" [9], and a standard value of this parameter is unity. However when we run the Rayleigh-Taylor problem with this merit factor, we get a smooth interface that is far from correct. This is a clear indication that the subzonal pressures can change the physics of the problem. On the other hand, we have found that choosing too small a merit factor will not adequately control the unphysical motions. In our experiments on the Rayleigh-Taylor problem, we have found that choosing the merit factor equal to 0.1 is sufficient to keep the purely Lagrangian grid from tangling.

In our ALE calculations we use the same initial grid as above, and apply rezoning each 20-th step. We continue to use the subzonal pressures with the same merit factor; however because the rezoning maintains a much smoother grid, the subzonal forces are much smaller and as we will see from numerical results they do not affect the physics perceptibly.



a) b) c)

Figure 16: *Comparison of fragments of the grids at $t = 7.0$ for Lagrangian calculations, ALE calculations with Winslow rezone strategy, and ALE calculations with new RJM rezone strategy. a) Lagrangian, b) Winslow, c) RJM.*

In Fig. 16 we present fragments of the grids at $t = 7.0$ for the purely Lagrangian calculations, the ALE calculations using Winslow rezone strategy, and the ALE calculations using our new RJM rezone strategy. The corresponding comparison of these solutions with the FronTier (320×60 grid) results is presented in Fig. 17. Clearly the purely Lagrangian calculation produces a very distorted grid, which reduces accuracy of the solution as can be seen from Fig. 17 a). The ALE calculation using the Winslow rezone strategy produces overly smoothed grid, leading to the result in Fig. 17 b). The width of the mixed layer is wrong, as is its shape. Moreover the volumes of the light and heavy fluids are wrong. The ALE calculations with our new RJM rezone strategy produces a grid that more closely follows the solution (as much as a logically rectangular grid is capable), and leads to the results presented in 17 c). Here the width of the mixed layer is almost perfect, as are the positions of the upper and lower boundaries. The volumes of the two fluids are also well preserved.

Finally we compare results obtained by the ALE method using our RJM rezone strategy with FronTier results at $t = 8.5$. In Fig. 18 we present a fragment of the ALE grid to demonstrate how closely a logically-rectangular grid can reproduce the solution features.

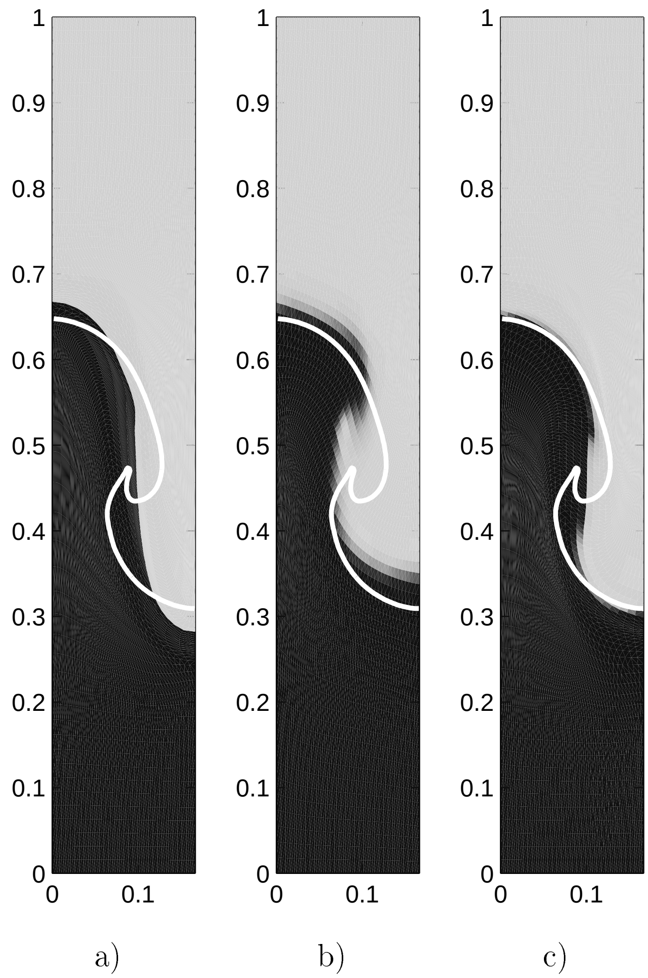
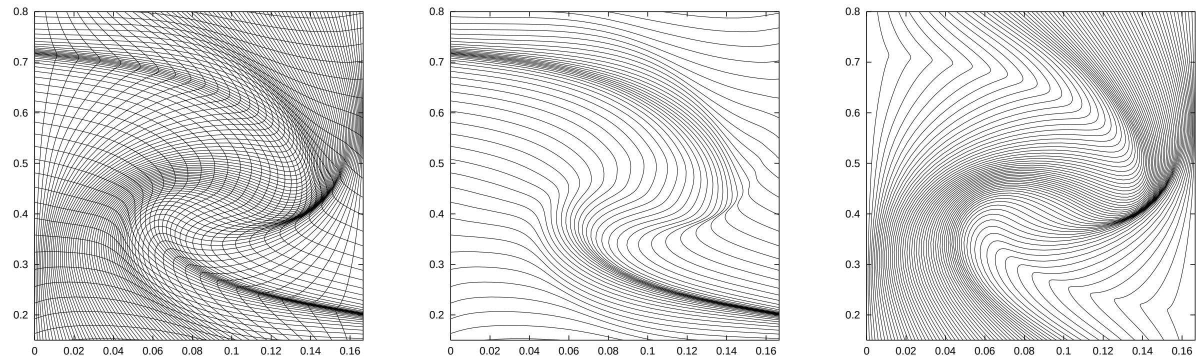


Figure 17: *Comparison of Lagrangian calculations, ALE calculations with Winslow potential rezone strategy, and ALE calculations with new RJM rezone strategy at* $t = 7.0$*: a) Lagrangian, b) Winslow, c) RJM*



a) b) c)

Figure 18: *Fragment of grid for ALE calculations using RJM rezone strategy for* $t = 8.5$ *a) Fragment of the grid, b)* ξ *- lines, c)* η *-lines.*

To make comparison of the density field more visually explicit we map our results from the computational domain to the unit square. In Fig. 19 we superpose the ALE and FronTier results. The interface obtained by FronTier is shown as a solid line, while the density distribution obtained by the our ALE method is shown in grayscale. The ALE results capture the width of mixing layer closely; of equal interest, the ALE simulation reproduces some of the details of the interface shape.

7 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 grid with optimized geometrical quality whose 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 grid has appropriate geometrical quality while keeping the

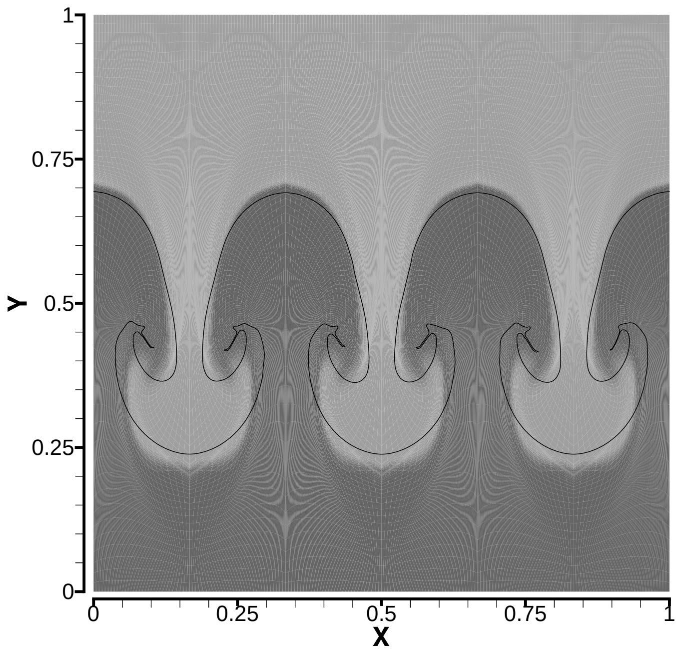


Figure 19: *Comparison of ALE calculations with results obtained by front tracking code FronTier at* $t = 8.5$.

grid nearly Lagrangian. This is achieved by a special local procedure of constructing RJM's, which is based on local optimization.

We have implemented our RJM approach in context of 2D structured code, to demonstrate feasibility. We have presented preliminary ALE calculations for a Rayleigh-Taylor instability problem and have shown significant improvements in accuracy when compared with a purely Lagrangian method. We note however that these calculations also indicate the limitations related to the use of structured grids. The flexibility of the RJM framework will allow us to extend our ideas to unstructured grids and further to Free-Lagrangian methods, where the connectivity of the grid may change with time [36].

Although our rezoning strategy has focused on preserving geometric quality of the grid, the RJM framework is also general enough to allow us to include more information about the solution in the construction of the RJM. In particular, in the future we plan to incorporate error estimates [1, 38] to create truly adaptive ALE algorithms, where error can be minimized by appropriate choice of the computational grid.

It is difficult to compare the computational effort — i.e., the accuracy for a given level of effort — of our RJM/ALE algorithm with other ALE algorithms. Partly this is because it is difficult to quantify accuracy *per se* in the complex problems that require ALE schemes. Also, one should expect that computational efficiency will be strongly problem dependent, with simpler schemes performing better on simpler problems. A rezoning strategy highly optimized by the user for a particular problem may run very efficiently, but require considerably more time and effort to set up. Finally, we note that in applications with many coupled physical processes, hydrodynamic algorithms usually represent a small part of the overall effort, and even factors of ten in efficiency may be insignificant.

The simplicity of a purely Lagrangian simulation should be preferred whenever applicable. This is clearly not the case in the Rayleigh-Taylor problem of Section 6, where the Lagrangian simulation is both inaccurate and subject to severe mesh distortion. At the other end of the spectrum, the RJM/ALE algorithm that we have described is both accurate and robust, while allowing flexibility for choosing alternate rezone strategies. To paraphrase Albert Einstein, an effective hydrodynamic algorithm should be as simple as possible, but not simpler.

Acknowledgment

The authors thank B. Wendroff, R. Bos, T. Dey, J. Grove, V. Mousseau, E. Caramana, M. Berndt, G. Lapenta, J. Campbell, J. Brackbill, J. Dukowicz, and many others for fruitful discussions and constructive comments. 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 L. Margolin and 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. L. Margolin and M. Shashkov also acknowledge the support of DOE's Accelerated Strategic Computing Initiative (ASCI) and support from LDRD-ER (# 2000045) at Los Alamos National Laboratory.

References

[1] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, A Wiley-Interscience Publication, John Wiley & Sons, Inc., 2000.

[2] M. J. Baines *Moving Finite Elements*, Oxford Science Publications, Clarendon Press, Oxford, 1994.

[3] D. J. Benson, *An Efficient, Accurate, Simple ALE Method for Nonlinear Finite Element Programs*, Computer Methods in Applied Mechanics and Engineering, **72** (1989), pp. 305–350.

[4] D. J. Benson, *Computational Methods in Lagrangian and Eulerian Hydrocodes*, Computer Methods in Applied Mechanics and Engineering, **99** (1992), pp. 235-394.

[5] D. J. Benson, *Momentum Advection on a Staggered Mesh*, Journal of Computational Physics, **100** (1992), pp. 143–162.

[6] J.U. Brackbill and J.S. Saltzman, *Adaptive Zoning for Singular Problems in Two Dimensions*, Journal of Computational Physics, **46** (1982), pp. 342–368.

[7] J. Campbell and M. Shashkov, *The Compatible Lagrangian Hydrodynamics Algorithm on Unstructured Grids*, LA-UR-00-3231, (http://cnls.lanl.gov/∼ shashkov), Submitted to Journal of Computational Physics.

[8] J. Campbell and M. Shashkov, *A Tensor Artificial Viscosity Using a Mimetic Finite Difference Algorithm*, LA-UR-00-4507, (http://cnls.lanl.gov/∼ shashkov), Submitted to Journal of Computational Physics.

[9] E. J. Caramana and M. J. Shashkov, *Elimination of Artificial Grid Distortion and Hourglass-Type Motions by means of Lagrangian Subzonal Masses and Pressures*, Journal of Computational Physics, **142** (1998), p. 521 .

[10] E. J. Caramana, M. J. Shashkov and P. P. Whalen, *Formulations of Artificial Viscosity for Multi-Dimensional Shock Wave Computations*, Journal of Computational Physics, **144** (1998), p. 70.

[11] E. J. Caramana, 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** (1998), p. 227.

[12] D. E. Carroll, Sandia National Laboratories, NM, (private communication, 2000).

[13] A. Charakhch'yan and S. Ivanenko, *A variational form of the Winslow grid generator*, J. Comput. Phys., **136** (1997), pp. 385–398.

[14] J. Donea, S. Giuliani, and J. P. Halleux, *An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Dynamic Fluid-Structure Interactions*, Computer Methods in Applied Mechanics and Engineering, **33** (1982), pp. 689–723.

[15] J.K. Dukowicz, *A Simplified Adaptive Mesh Technique Derived from the Moving Finite Element Method*, Journal of Computational Physics, **56** (1984), pp. 324–342.

[16] J.K. Dukowicz and J.W. Kodis, *Accurate Conservative Remapping (Rezoning) for Arbitrary Lagrangian-Eulerian Computations*, SIAM J. Stat. Comput., **8** (1987), pp. 305–321.

[17] J.K. Dukowicz, M.C. Cline and F.L. Addessio, *A General Topology Godunov Method*, Journal of Computational Physics, **82** (1989), pp. 29–63.

[18] L. Freitag and P. Knupp, *Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number*, Proceedings of the 8th Intl. Meshing Roundtable, S. Lake Tahoe CA, 1999, pp. 247-258,

[19] S. Giuliani, *An Algorithm for Continuous Rezoning of the Hydrodynamic Grid in Arbitrary Lagrangian-Eulerian Computer Codes*, Nuclear Engineering and Design, **72** (1982), pp. 205–212.

[20] J. Glimm, J. W. Grove, X.-L. Li, K.-M. Shyue, Q. Zhang, and Y. Zeng. *Three dimensional front tracking*, SIAM J. Sci. Comp., **19** (1998), pp. 703–727.

[21] J. Glimm, J. Grove, X.-L. Li, W. Oh, and D. H. Sharp. *A critical analysis of Rayleigh-Taylor growth rates*, J. Comp. Phys., 2000, To appear, SUNY Preprint SUNYSB 00-19; LANL LA-UR 99-5582.

[22] C.W. Hirt, A.A. Amsden and J.L. Cook, *An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds*, Journal of Computational Physics, **14** (1974), pp. 227–253, reprinted in **135** (1997), pp. 203–216.

[23] D.S. Kershaw, M.K. Prasad, M.J. Shaw, and J.L. Milovich, *3D Unstructured Mesh ALE Hydrodynamics with the Upwind Discontinuous Finite Element Method*, Computer Methods in Applied Mechanics and Engineering, **158** (1998), pp. 81–116.

[24] P. Kjellgren and J.Hyvarien. *An Arbitrary Lagrangian-Eulerian Finite Element Method*, Computational Mechanics, **21** (1998), pp. 81–90.

[25] P. Knupp, *Mesh generation using vector-fields*, J.Comput.Phys., **119**, pp142-148, 1995.

[26] P. Knupp, *Jacobian-weighted Elliptic Grid Generation*, SIAM J. Sci. Comput., Vol. 17, No. 6, pp1475-1490, 1996.

[27] P.Knupp, *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. Engng., **48** (2000), pp. 401–420.

[28] P.Knupp, *Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II - A framework for volume mesh optimization and the condition number of the Jacobian matrix*, Int. J. Numer. Meth. Engng., **48** (2000), pp. 1165-1185.

[29] P. Knupp, *Matrix Norms and the Condition Number - A general framework to improve mesh quality via node-movement*, Proceedings of the 8th Intl. Meshing Roundtable, S. Lake Tahoe CA, 1999, pp. 13- 22.

[30] P. Knupp and N. Robidoux, *A Framework for Variational Grid Generation: Conditioning the Jacobian Matrix with Matrix Norms*, SIAM J. Sci. Comput., **21** (2000), pp. 2029–2047.

[31] H. J. Kull, *Theory of the Rayleigh-Taylor Instability*, Physics Reports, **206** (1991), pp. 197–325.

[32] V.D. Liseikin, *Grid Generation Methods*, Springer, 1999, pp. 91–93.

[33] R. Lohner and C. Yang, *Improved ALE Mesh Velocities for Moving Bodies*, Communications in Numerical Methods in Engineering, **12** (1996), pp. 599–608.

[34] L. G. Margolin and J. J. Pyin, *A Method of Treating Hourglass Patterns*, in Proc., Numerical Methods in Laminar and Turbulent Flow, edited by C. T. Taylor and N. M. Hafey (1987), p. 149.

[35] L.G. Margolin, *Introduction to "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds"*, Journal of Computational Physics, **135** (1997), pp. 198–202.

[36] N. Mikhailova, V.Tishkin, N.Turina, and A.Favorskii, and M. Shashkov *Numerical modeling of two-dimensional gasdynamics flows on a variable-structure mesh*, - U.S.S.R. Comput. Maths. Math. Phys. **26** (1986), pp. 74–84.

[37] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer-Verlag New York Inc., 1999.

[38] D. Kroner and M. Ohlberger, *A Posteriori Error Estimates for Upwind Finite Volume Schemes for Nonlinear Conservation Laws in Multi Dimensions*, Math. of Comp., **69** (1999), pp. 25–39.

[39] J. S. Peery and D. E. Carroll, *Multi-Material ALE Methods in Unstructured Grids*, Computer Methods in Applied Mechanics and Engineering, **187**, (2000), pp. 591–619.

[40] M. Shashkov, *Conservative Finite-Difference Methods on General Grids*, CRC Press, Boca Raton, 1995.

[41] A. I. Shestakov, J. A. Harte, and D. S. Kershaw. *Solution of the Diffusion Equation by the Finite Elements in Lagrangian Hydrodynamic Codes*, Journal of Computational Physics, **76**, pp. 385-413, (1988).

[42] C. Stoker, C. Gay, F. Bay, and J.-L. Chenot, *A Velocity Approach for the ALE-method applied for 2D and 3D Problems*, Simulation of Material Processing: Theory, Methods and Applications, Huetink and Baaijens (eds)., 1998 Balkema, Rotterdam.

[43] A. N. Tikhonov and A. A. Samarskii, *Equations of Mathematical Physics*, A Pergamon Press Book, The Macmillan Company, New York, 1963.

[44] J.G. Tinoco-Ruiz, *Funcionales Discretos para la Generacion de Mallas Suaves y Convexas sobre Regiones Planas Irregulares*, Ph.D. Thesis, CIMAT, Guanajuato, Mexico, 1997.

[45] A. Winslow, *Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh*, J. Comput. Phys., **1**, (1966) pp. 149-172.

Figure Captions

1. Flow-chart of rezoning algorithm.
2. a) Image of lines $\xi = const$ and $\eta = const$ under map $x(\xi, \eta)$, $y(\xi, \eta)$, b) coordinate lines and tangent vectors.
3. Logically rectangular grid and discrete tangent vectors.
4. Discrete tangential vectors and triangle (Jacobian Matrix)
5. a) Smooth grid, b) Non-smooth grid
6. Different initial guesses a) Random grid, b) Smooth grid, c) Tensor product grid.
7. a) Discontinuous map, b) Unique grid obtained by optimization
8. Virtual movement of central node, and construction of four RJM's. Lagrangian nodes are marked by filled circles.
9. Local Smoothing, a) Lagrangian and smoothed tangential vectors of the four RJMs at node (i, j), $\mathbf{e}_d$ is the displacement vector connecting Lagrangian and smoothed central nodes, b) Reference tangential vectors $\mathbf{e}\xi^r_{i+\frac{1}{2},j}$, $\mathbf{e}\eta^r_{i+\frac{1}{2},j}$ that define the Reference Jacobian matrix $J_{i,j}^{i+\frac{1}{2},j+\frac{1}{2}}$.
10. Examples of Local Smoothing
11. Comparison of Lagrangian and rezoned grids, a) Smooth Lagrangian grid and corresponding rezoned grid, b) Nonsmooth (random) Lagrangian grid and corresponding rezoned grid

12. Comparison of Lagrangian and rezoned grids, when the Lagrangian grid consists of subdomains and where the grid is smooth within the subdomains, but is not smooth on the interface between subdomains.

13. Shestakov grid and its smoothed version.

14. Subzonal forces. a) Subzones, density and mass specified in subzones, p is the index of vertex (point) and z is index of the zone, b) Hourglass motion and corresponding change in subpressures preventing developing of this motion.

15. Comparison of Winslow and and RJM rezone strategies for "chevron" grid: a) Winslow, b) RJM

16. Comparison of fragments of the grids at $t = 7.0$ for Lagrangian calculations, ALE calculations with Winslow rezone strategy, and ALE calculations with new RJM rezone strategy. a) Lagrangian, b) Winslow, c) RJM.

17. Comparison of Lagrangian calculations, ALE calculations with Winslow potential rezone strategy, and ALE calculations with new RJM rezone strategy at $t = 7.0$: a) Lagrangian, b) Winslow, c) RJM

18. Fragment of grid for ALE calculations using RJM rezone strategy for $t = 8.5$ a) Fragment of the grid, b) ξ - lines, c) η -lines.

19. Comparison of ALE calculations with results obtained by front tracking code FronTier at $t = 8.5$.

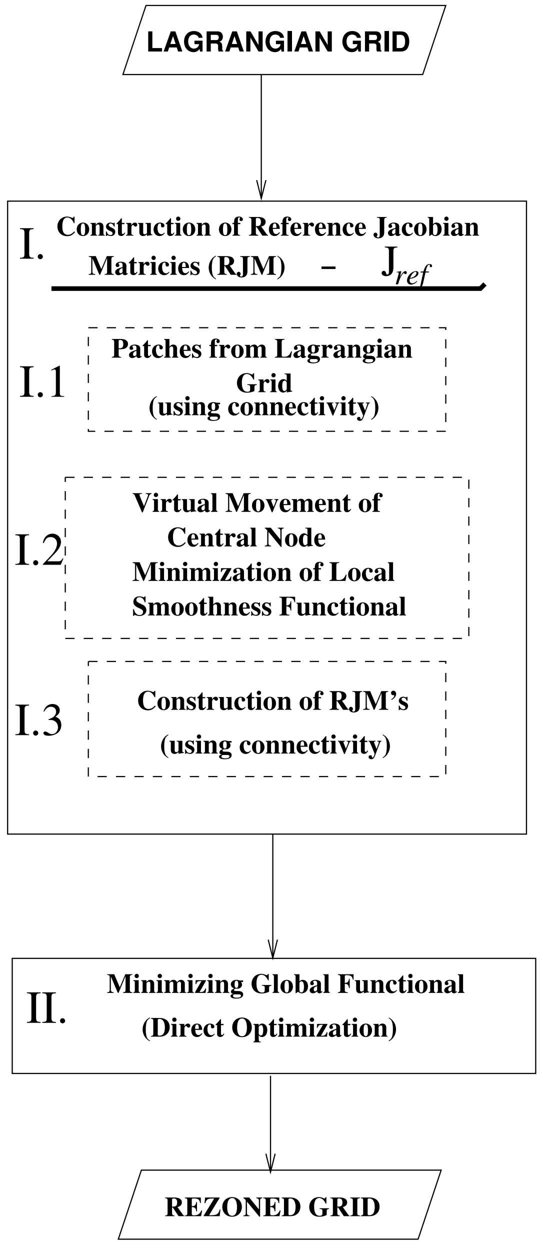
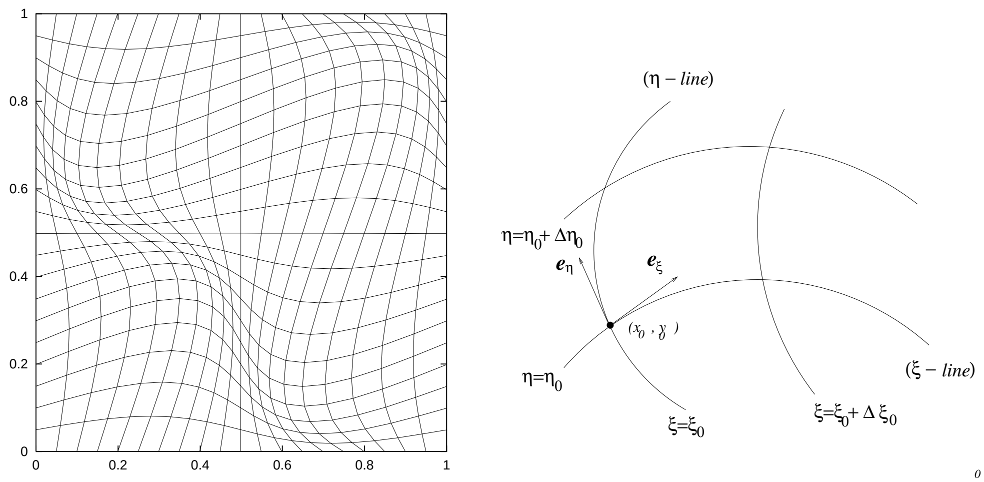

LAGRANGIAN GRID
I.
Construction of Reference Jacobian
Matricies (RJM) – J_{ref}
I.1
Patches from Lagrangian
Grid
(using connectivity)
I.2
Virtual Movement of
Central Node
Minimization of Local
Smoothness Functional
I.3
Construction of RJM's
(using connectivity)
II.
Minimizing Global Functional
(Direct Optimization)
REZONED GRID

Figure 1:



a) b)

Figure 2:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

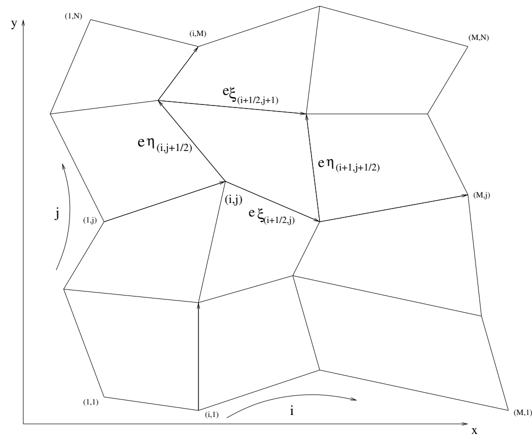


Figure 3:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

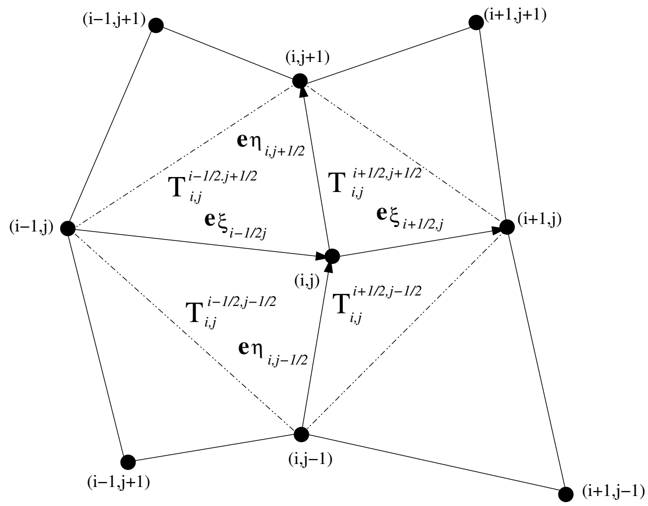
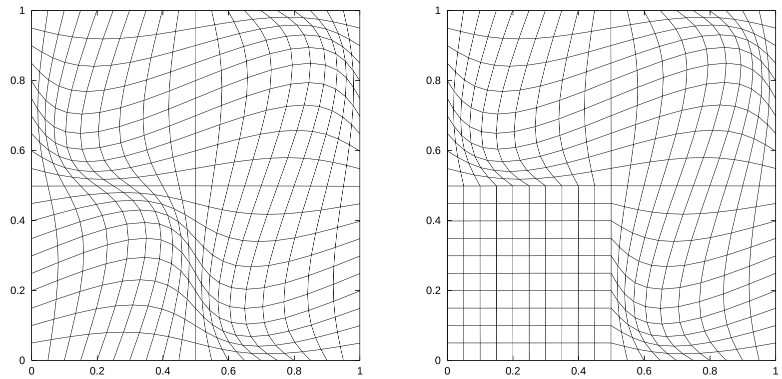


Figure 4:

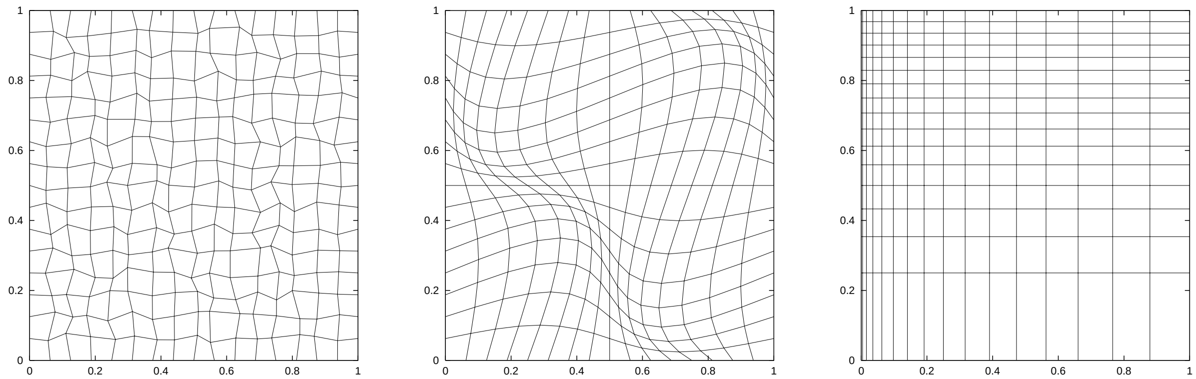
Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov



a) b)

Figure 5:

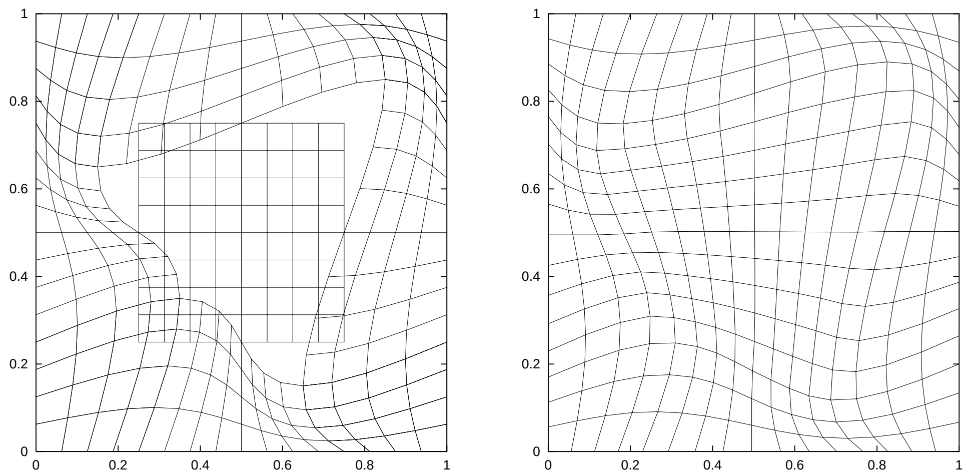
Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov



a) b) c)

Figure 6:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov



a) b)

Figure 7:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

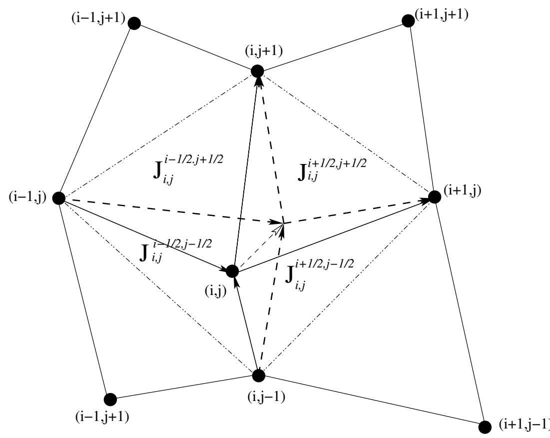
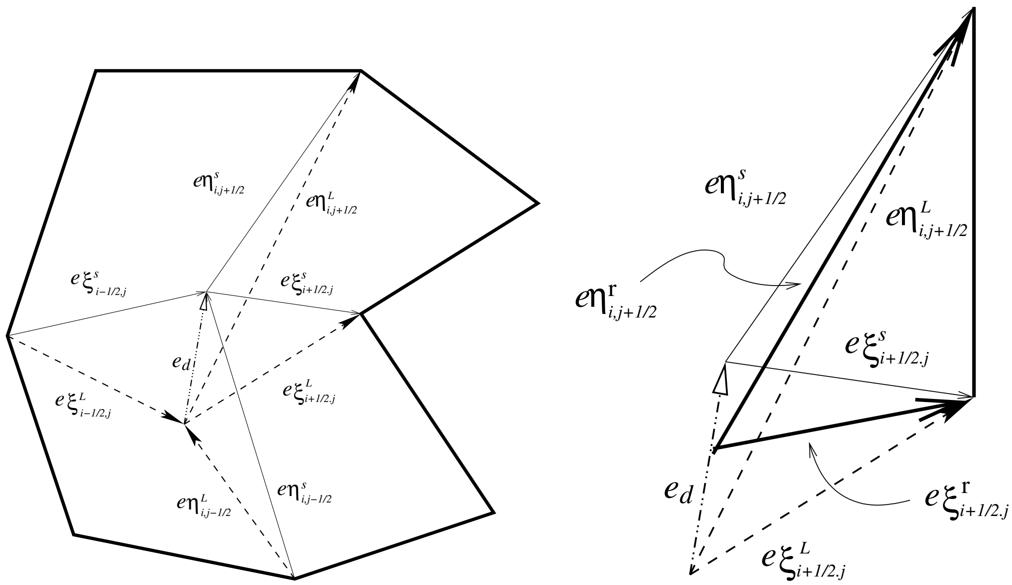


Figure 8:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov



a) b)

Figure 9:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

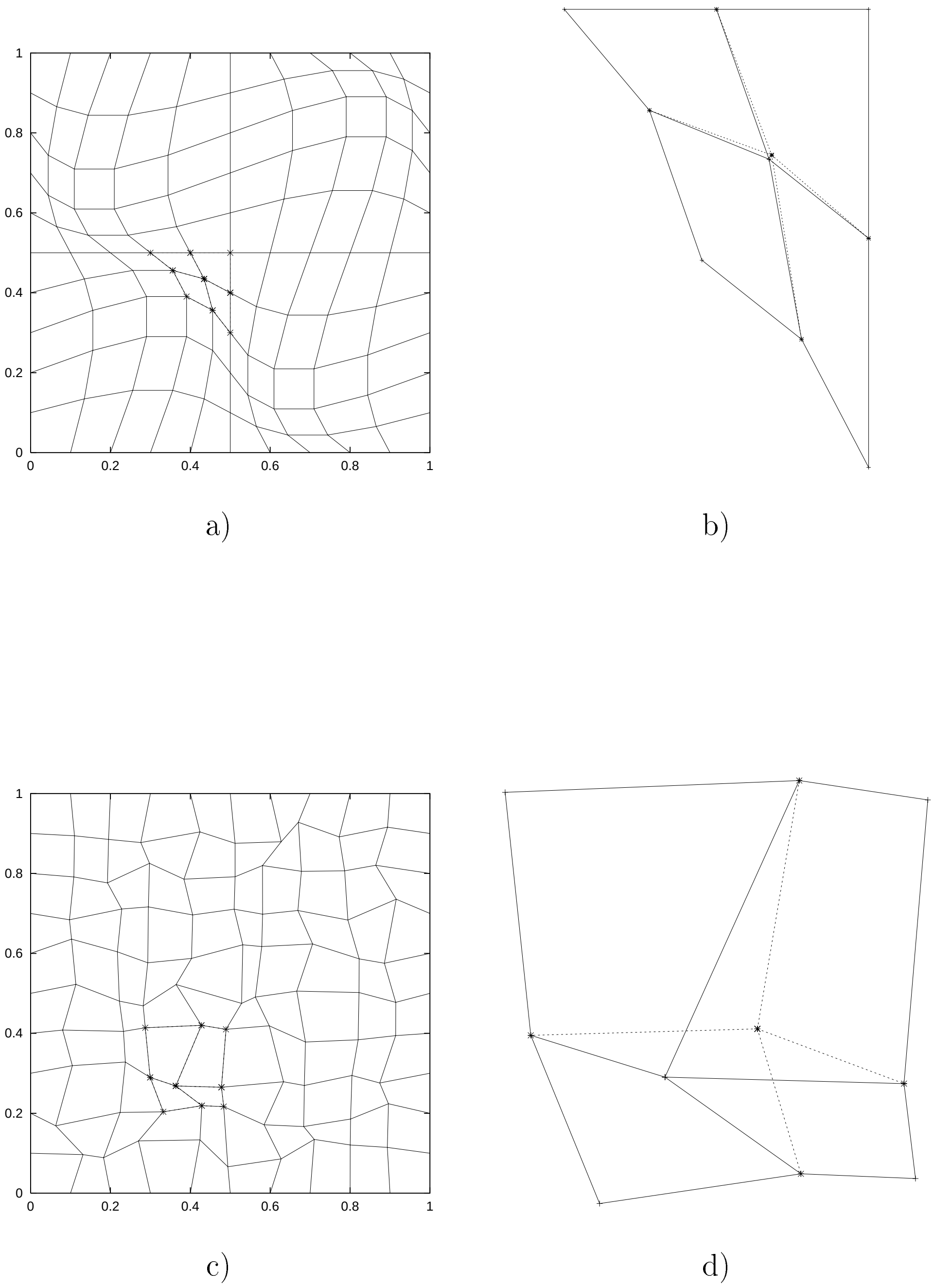


Figure 10:

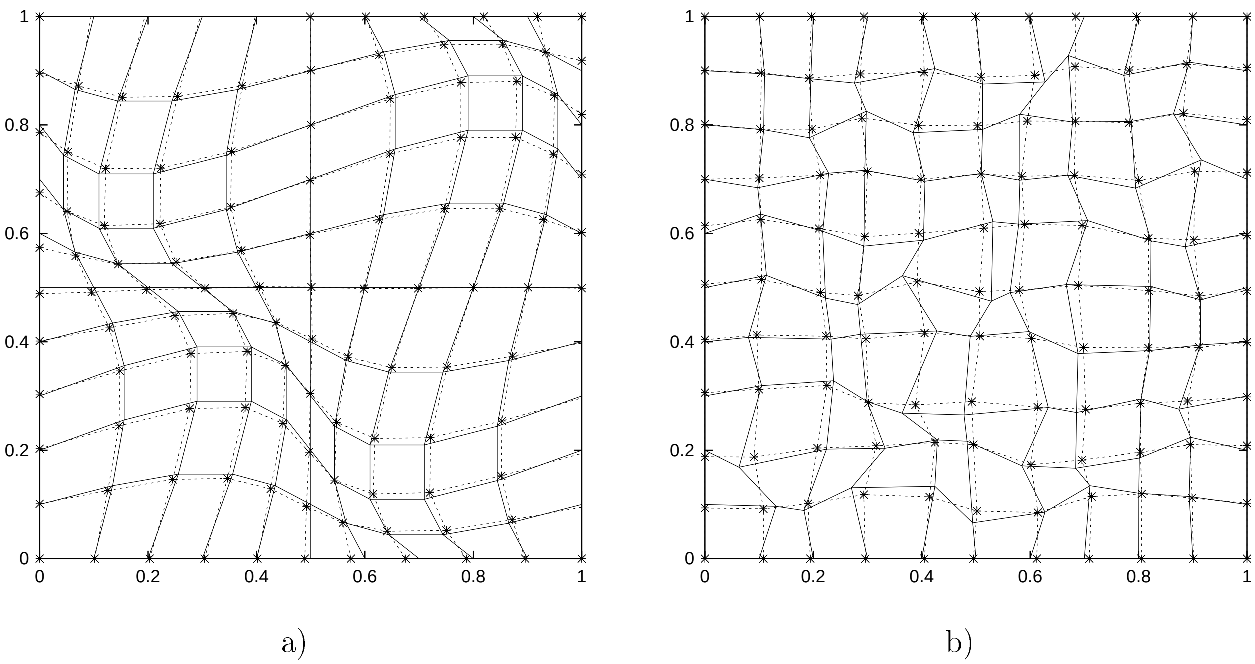


Figure 11:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

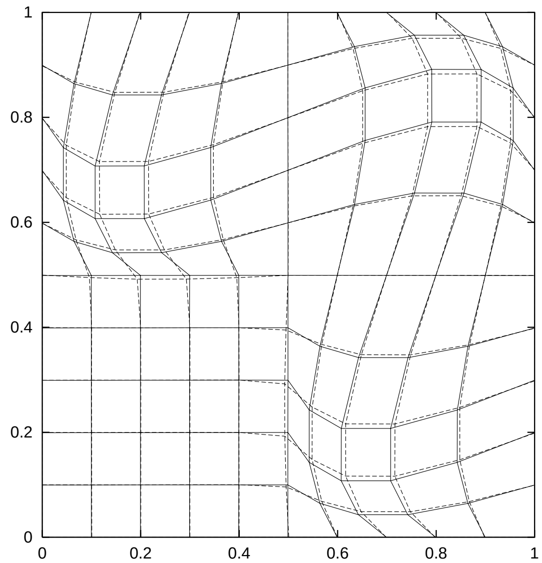
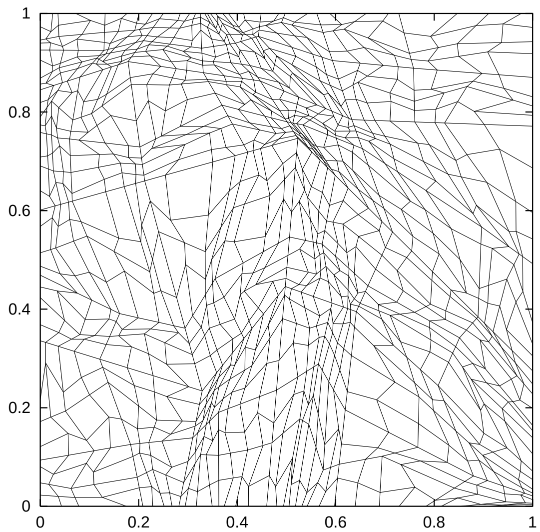
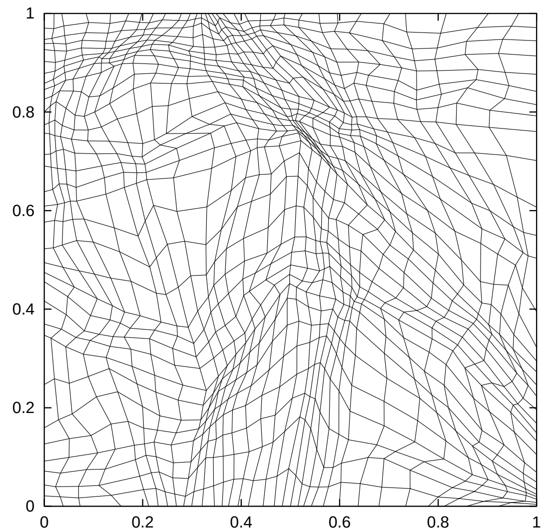


Figure 12:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov



a)



b)

Figure 13:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

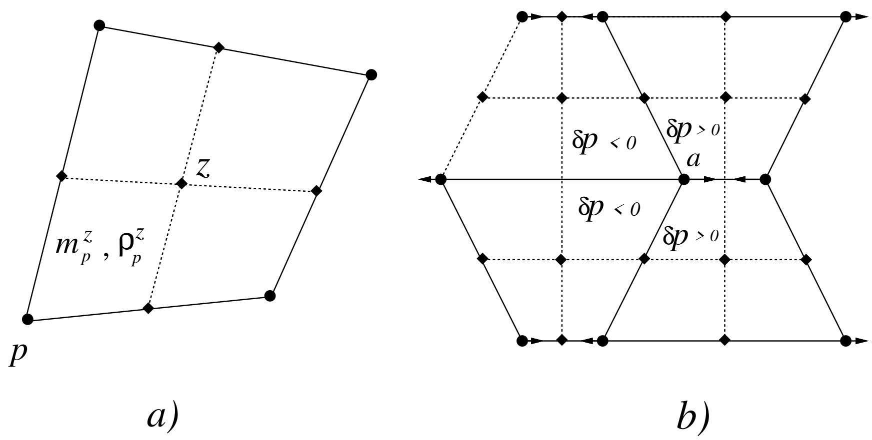
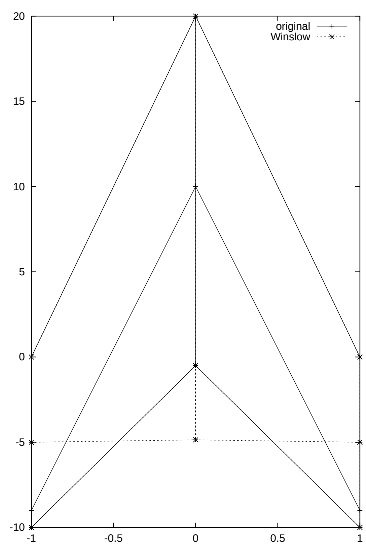
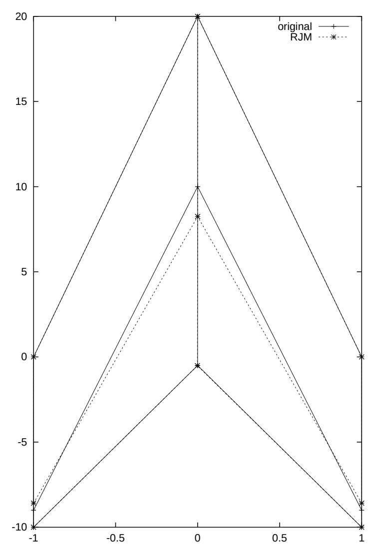


Figure 14:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

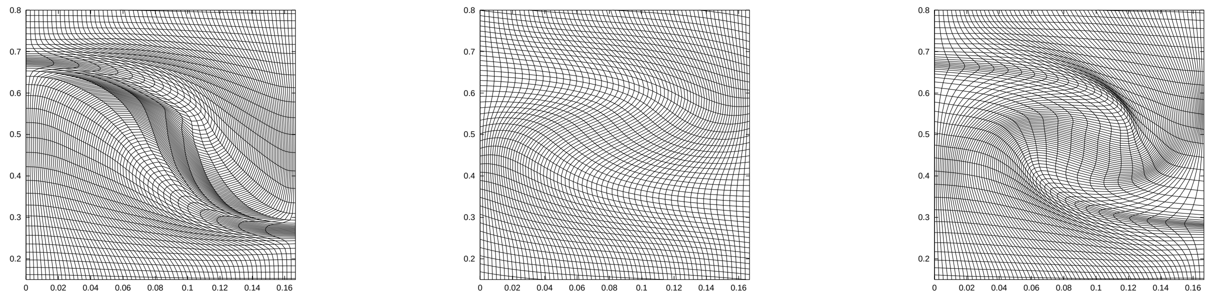

original
Winslow
20
15
10
5
0
-5
-10
-1
-0.5
0
0.5
1

a)


original
RJM
20
15
10
5
0
-5
-10
-1
-0.5
0
0.5
1

b)

Figure 15:



a) b) c)

Figure 16:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

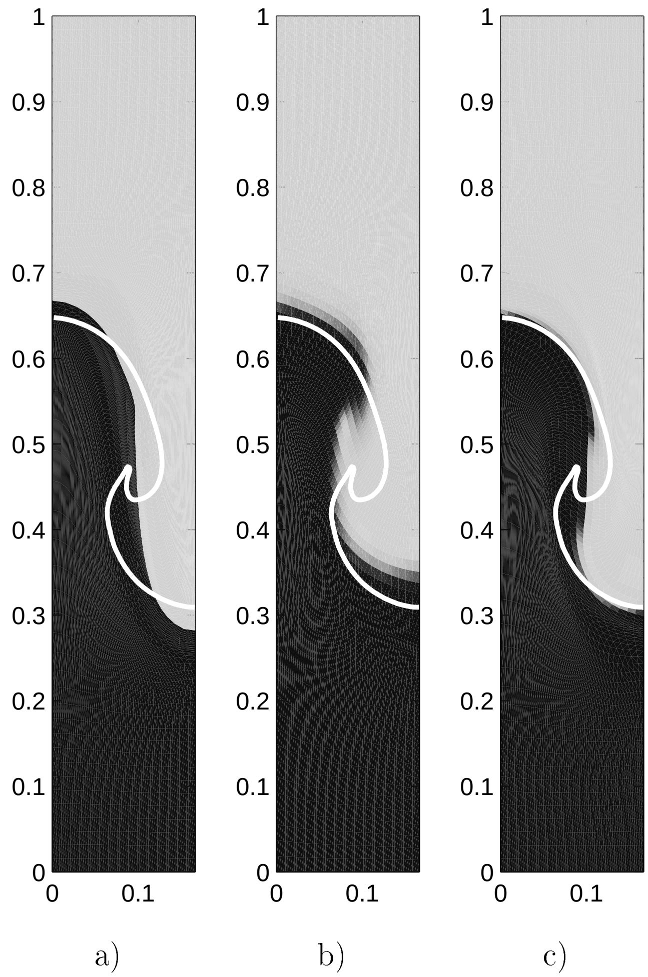
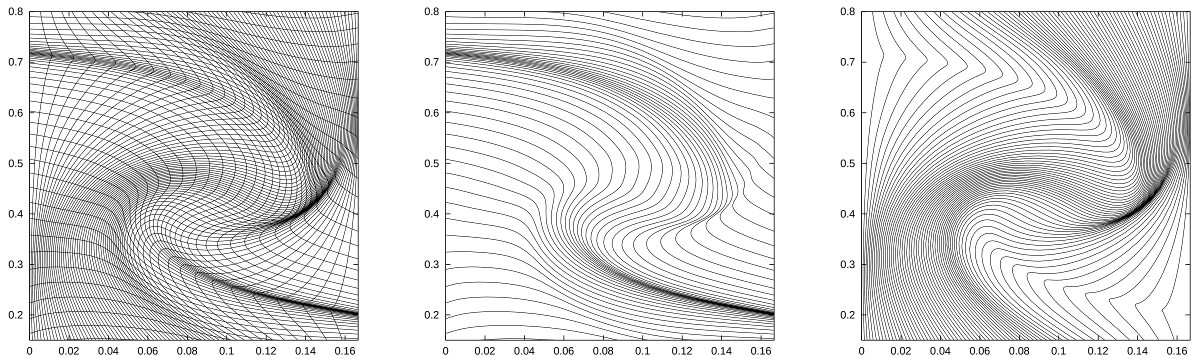


Figure 17:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov



a) b) c)

Figure 18:

Journal of Computational Physics, L. Margolin, P. Knupp and M. Shashkov

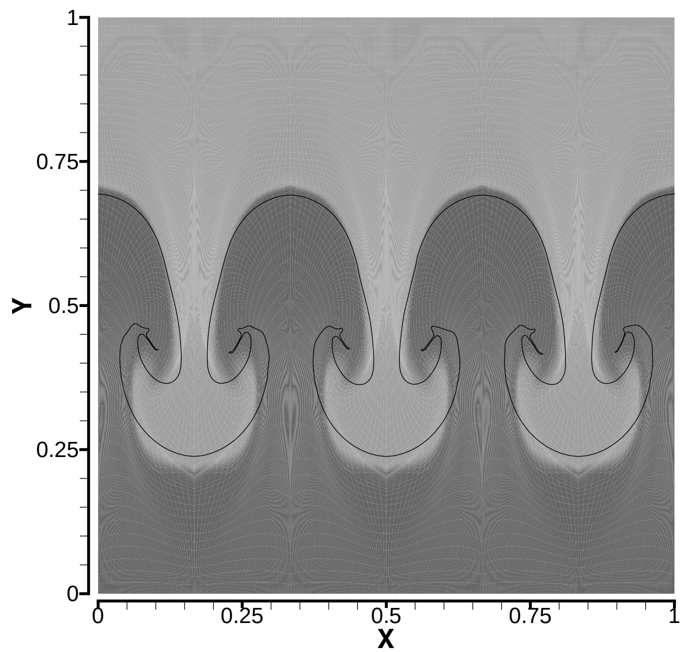


Figure 19:

Comparison of ALE calculations with results obtained by front tracking code FronTier at $t = 8.5$.