Proceedings of ALGORITMY 2005
pp. 93–102

ON QUALITY IMPROVEMENT OF TRIANGULAR MESHES USING NODE RECONNECTION*

PAVEL VÁCHAL† AND RAO V. GARIMELLA‡

Abstract. A novel algorithm for rezoning of computational meshes is proposed, based on reconnection of nodes, without any repositioning. Since the ultimate objective is an efficient mesh quality improvement method for ALE simulations, the physical behavior of the simulated problem is taken into account. The proposed process minimizes the error of interpolation of a particular discrete function, such as the mean value of some state variable, which characterizes the system under study. A practical implementation of the method is suggested and its usability demonstrated by numerical results.

Key words. mesh rezoning, node reconnection, edge swapping, ALE

AMS subject classifications. 65M50, 65N50, 65M06, 65N06, 65D05

1. Introduction. The Arbitrary Lagrangian-Eulerian (ALE) algorithm, first proposed in [1], is a very promising approach for simulations of complex physical processes, for example in material science or fluid dynamics. The ALE methods consist of three steps: The Lagrangian step, in which nodes move with the fluid, is followed by the rezone step, where the deformed mesh is modified in order to keep good quality. Finally, in the remapping step, the solution is conservatively interpolated from the old mesh to the new, improved one. Here we focus on the mesh rezone step or in other words the process of improving the mesh quality so that the next Lagrangian step becomes possible. The remapping part is not discussed here. More on this topic can be found e.g. in [2].

Two of the ways to rezone the mesh are by node repositioning and by node reconnection. While the ultimate goal is to use a combination of node repositioning and reconnection to effectively improve a mesh, we believe that to gain full control about the rezoning process, one first has to understand and manage each of the two techniques separately. Recently, we suggested a set of methods based on mesh repositioning, without changes of connectivity [3]. Here we investigate rezoning by pure node reconnection. Positions of nodes stay fixed and the mesh quality is being improved by changing of the edges between them. For simplicity, we are working on an unstructured triangular mesh, where the process can be simply described by a sequence of edge "swaps", i.e. replacements of a selected edge by another one. Node reconnection for non-simplicial meshes is more complex and is not very well defined.

Experience shows that geometrical changes of the mesh should be closely related to behavior of the simulated physical problem to achieve best results. We seek to

* This work was performed under the auspices of the US Department of Energy at Los Alamos National Laboratory, under contract W-7405-ENG-36. The authors acknowledge the partial support of the DOE/ASCR Program in the Applied Mathematical Sciences and the Laboratory Directed Research and Development program (LDRD). The authors also acknowledge the partial support of DOE's Advanced Simulation and Computing (ASC) program. P.V. was partly supported by FRVS grant 2087/2004, Czech Grant Agency grant GACR 202/03/H162 and Czech Technical University internal grant CTU 0411014.

† Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Břehová 7, 115 19 - Prague 1, Czech Republic (vachal@galileo.fjfi.cvut.cz).

‡ Group T-7, Los Alamos National Laboratory, Los Alamos, NM, USA.

meet this requirement by basing the rezoning procedure on the behavior of a selected state variable in the domain, for example density in fluid dynamics or temperature in plasma physics. This allows us to test our rezoning method in ALE simulations of many phenomena, such as Rayleigh-Taylor instability or laser-plasma interactions.

The whole process will be first explained on a smooth general function. Then the practical implementation will be described, utilizing a real state variable given by the discrete set of its mean values over the cells. Finally, numerical results will reveal the basic properties of the proposed method.

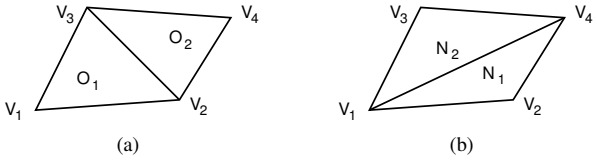

FIG. 2.1. *Definition of Edge swapping: A pair of cells sharing the original edge (a). New triangles after swapping of the edge (b).*

2. Swapping Algorithm. Our main goal is rezoning in a solution-sensitive manner. We choose to achieve this by forming an objective function and minimizing it using edge swapping:

- Loop through all interior edges. For each edge,
  1. Compute the contributions of the two triangular cells sharing this edge (i.e. triangles O_1 and O_2 in Fig. 2.1(a)) to the global objective function.
  2. Imagine that the edge "swapped" as in Fig. 2.1(b), forming two new triangles N_1 and N_2. Compute contributions of these new triangles to the objective function.
  3. Compare the values according to selected criterion.[1] If the objective function is reduced by swapping, then perform the swap, i.e. remove the original edge and create the new one.
- Repeat for next edge, until the mesh stays unchanged during one whole loop.

In the practical implementation, we used the framework called MSTK [4] to represent and handle the mesh. This framework frees up application programmers from worrying about the details of mesh data structures and allows them to focus on implementing their high level application.

3. Objective Function. There is a wide set of mesh quality measures, most of which are based purely on the element geometry (condition number for particular vertices, average cell areas, etc.) As mentioned above, we suggest an approach which also utilizes the physics of the simulated problem.

In the cells of the mesh, we construct a piecewise constant or piecewise linear interpolation of discrete function ρ, which in some sense represents the physical system we deal with. For example, ρ may be density in fluid dynamics. The quality of a particular cell is then taken as an estimate of the L_2 error of the discrete approximation integrated over the cell. In other words, from the two possible mesh configurations, we select that one, which allows better piecewise constant or piecewise linear interpolation

[1] We always compare two pairs of numbers, representing quality of the triangles before and after swapping. We can for example demand the improvement of their average quality by some amount, or require that the worse triangle after swapping is better than the worse triangle before. In the current implementation, we use a combination of multiple criteria with various priorities.

of the underlying function in the sense of least squares. How we estimate the L_2 error for the piecewise constant or piecewise linear reconstruction is described below.

Consider a smooth two-dimensional function $f(x, y)$, defined on the domain of the mesh. We focus on one particular triangle T with centroid (x_0, y_0). The Taylor expansion at any point (x, y) of the triangle be

$$f(x,y) = f(x_0,y_0) + (x-x_0)\left.\frac{\partial f}{\partial x}\right|_{\substack{x_0\\y_0}} + (y-y_0)\left.\frac{\partial f}{\partial y}\right|_{\substack{x_0\\y_0}} + \mathcal{O}\left((x-x_0)^2, (y-y_0)^2\right) \tag{3.1}$$

Ignoring terms of second order and higher, we can rewrite (3.1) as

$$f(x,y) - f(x_0,y_0) \approx \delta x \left.\frac{\partial f}{\partial x}\right|_{\substack{x_0\\y_0}} + \delta y \left.\frac{\partial f}{\partial y}\right|_{\substack{x_0\\y_0}}$$

where δx denotes $(x - x_0)$ and δy equals to $(y - y_0)$. Note, that the left-hand term is the difference of the function and its value in centroid.[2] Let $I_T^{(0)}$ be the integral of square of the right-hand part over triangle T, i.e.

$$I_T^{(0)} = \iint\limits_T \left(\delta x \left.\frac{\partial f}{\partial x}\right|_{\substack{x_0\\y_0}} + \delta y \left.\frac{\partial f}{\partial y}\right|_{\substack{x_0\\y_0}}\right)^2 \mathrm{d}x\ \mathrm{d}y \tag{3.2}$$

We can see, that smaller values of $I_{T_i}^{(0)}$ for all mesh cells T_i mean better approximation of function f by its piecewise constant reconstruction and in this sense we can use $I_T^{(0)}$ as a measure of mesh quality.

Taking into account also second-order terms of Taylor expansion, we have

$$\begin{aligned} f(x,y) \;=\; & f(x_0,y_0) + \delta x \left.\frac{\partial f}{\partial x}\right|_{\substack{x_0\\y_0}} + \delta y \left.\frac{\partial f}{\partial y}\right|_{\substack{x_0\\y_0}} + \\ & + \delta x\ \delta y \left.\frac{\partial^2 f}{\partial x \partial y}\right|_{\substack{x_0\\y_0}} + \frac{(\delta x)^2}{2}\left.\frac{\partial^2 f}{\partial x^2}\right|_{\substack{x_0\\y_0}} + \frac{(\delta y)^2}{2}\left.\frac{\partial^2 f}{\partial y^2}\right|_{\substack{x_0\\y_0}} + \mathcal{O}\left(\delta x^3, \delta y^3\right) \end{aligned}$$

Similarly as above, we can write

$$\begin{aligned} f(x,y) - f(x_0,y_0) - \delta x \left.\frac{\partial f}{\partial x}\right|_{\substack{x_0\\y_0}} - \delta y \left.\frac{\partial f}{\partial y}\right|_{\substack{x_0\\y_0}} \approx \\ \approx \delta x\ \delta y \left.\frac{\partial^2 f}{\partial x \partial y}\right|_{\substack{x_0\\y_0}} + \frac{(\delta x)^2}{2}\left.\frac{\partial^2 f}{\partial x^2}\right|_{\substack{x_0\\y_0}} + \frac{(\delta y)^2}{2}\left.\frac{\partial^2 f}{\partial y^2}\right|_{\substack{x_0\\y_0}} \end{aligned}$$

This time, taking square of the right-hand side and integrating it over triangle T, we have

$$I_T^{(1)} = \iint\limits_T \left(\delta x\ \delta y \left.\frac{\partial^2 f}{\partial x \partial y}\right|_{\substack{x_0\\y_0}} + \frac{(\delta x)^2}{2}\left.\frac{\partial^2 f}{\partial x^2}\right|_{\substack{x_0\\y_0}} + \frac{(\delta y)^2}{2}\left.\frac{\partial^2 f}{\partial y^2}\right|_{\substack{x_0\\y_0}}\right)^2 \mathrm{d}x\ \mathrm{d}y \tag{3.3}$$

and decreasing values of $I_T^{(1)}$ mean improving approximation by piecewise linear reconstruction.

[2] For constant and linear functions, value in centroid corresponds to mean value.

There are two questions left: how to choose the function f and how to compute its derivatives. The easiest way is to select an analytical function and compute its values and derivatives anywhere we need it. However, in real applications, we want to use a discrete function ρ, representing some physical quantity. Then we can take first derivatives as slopes of some piecewise linear interpolation, for example Barth reconstruction (see e.g. [5]). Other ways to compute the derivatives, particularly Green's Theorem and integration of forward differences, are described in the following section.

4. Approximation of Derivatives. Now, let us suggest, how to compute the derivatives of discrete function ρ. First we briefly mention a way based on Green's theorem and then we derive another approach, based on minimization of suitable functional using Taylor expansion.

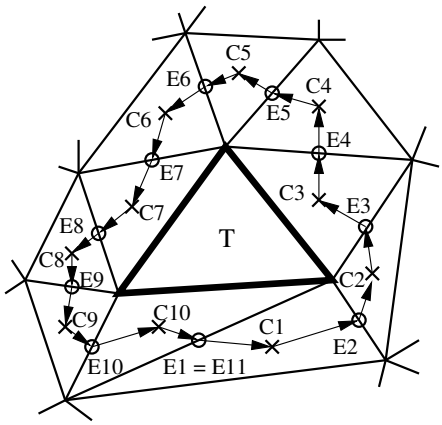

FIG. 4.1. *Approximation of first derivative using Green's Theorem.*

4.1. First-Order Derivatives from Green's Formula. First approach follows [6] and uses the Green's formulas

$$\frac{\partial \rho}{\partial x} = \lim_{S \to 0} \frac{1}{S} \int\limits_{\partial S} \rho \, \mathrm{d}y \qquad , \qquad \frac{\partial \rho}{\partial y} = - \lim_{S \to 0} \frac{1}{S} \int\limits_{\partial S} \rho \, \mathrm{d}x \tag{4.1}$$

For a particular cell, we apply them on region Ω surrounded by a path through centers of neighboring cells and edges as in Fig. 4.1. If the value of ρ is constant in each cell (e.g. if ρ is given by its discrete mean values), then the first formula in (4.1) takes the form

$$\left. \frac{\partial \rho}{\partial x} \right|_T = \frac{1}{V_\Omega} \sum_{k=1}^{N_T} \rho_{C_k} \left(y_{E_{k+1}} - y_{E_k} \right) \tag{4.2}$$

where N_T is number of cells neighboring to T ($N_T = 10$ in Fig. 4.1) and the volume V_Ω can be computed similarly with Green's formula

$$S = \frac{1}{2} \int\limits_{\partial S} x \, \mathrm{d}y - y \, \mathrm{d}x$$

applied to region Ω. The second formula in (4.1) is used in an analogous manner. Problems with boundary cells are solved by surrounding the mesh by zero-volume ghost cells.

4.2. Derivatives from Minimization of Taylor Terms. This approach is based on the following idea: There is a discrete function ρ given by mean values $\bar{\rho}$ in cells. In each cell, we want to construct an interpolation of order p (here $p = 0$ for constant or $p = 1$ for linear interpolation), which is given by Taylor expansion up to the selected order. At this point, the derivatives are unknown. We omit all expansion terms of order $p + 2$ and higher and compute the unknown derivatives so, that the square of terms of order $p + 1$ is minimal. That means, we replace the residual of Taylor expansion by the $p + 1$-st terms, which we compute from integrals over surrounding cells.

First we present an approach which is sufficient if only first derivatives are needed, i.e. if $p = 0$. Then we make a straightforward extension to a case when $p = 1$ and one needs also second derivatives. Besides our suggestion, there are of course many other ways to approximate the derivatives.

4.2.1. First Derivatives. Let us define the functional

$$F\left(\rho_x^C, \rho_y^C\right) = \sum_{N \in \Omega^C} \left(\bar{\rho}^N - \frac{1}{V^N} \int_N \rho^C(x, y) \, \mathrm{d}x \, \mathrm{d}y \right)^2, \tag{4.3}$$

where V^N is the volume of cell N, Ω^C is the set of all cells neighboring to cell C and ρ_C is the piecewise linear reconstruction

$$\rho^C = \bar{\rho}^C + \rho_x^C \left(x - x^C\right) + \rho_y^C \left(y - y^C\right) \tag{4.4}$$

in cell C with centroid (x^C, y^C). Value in centroid, denoted $\bar{\rho}^C$, is identical to mean value for constant as well as linear functions. Derivatives ρ_x^C and ρ_y^C are unknown. Subdividing the integral into small parts and introducing new symbols for these parts, we get

$$F = \sum_{N \in \Omega^C} \left(\bar{\rho}^N - \bar{\rho}^C - \frac{\rho_x^C I_x^{CN} + \rho_y^C I_y^{CN}}{V^N} \right)^2, \tag{4.5}$$

where

$$\begin{aligned} I_x^{CN} &= \int_N \left(x - x^C\right) \, \mathrm{d}x \, \mathrm{d}y = \int_N x \, \mathrm{d}x \, \mathrm{d}y - V^N x^C \\ I_y^{CN} &= \int_N \left(y - y^C\right) \, \mathrm{d}x \, \mathrm{d}y = \int_N y \, \mathrm{d}x \, \mathrm{d}y - V^N y^C. \end{aligned} \tag{4.6}$$

We want to compute the derivatives ρ_x^C and ρ_y^C so, that our piecewise linear reconstruction (4.4) is as close to the original function ρ as possible. To achieve this, we minimize the functional F, i.e. we set its derivatives with respect to unknown variables to zero, which gives the system

$$\frac{\partial F\left(\rho_x^C, \rho_y^C\right)}{\partial \rho_x^C} = 0 \qquad , \qquad \frac{\partial F\left(\rho_x^C, \rho_y^C\right)}{\partial \rho_y^C} = 0. \tag{4.7}$$

The derivatives are

$$\begin{aligned}\frac{\partial F}{\partial \rho_x^C} &= -2 \sum_{N\in\Omega^C} \left(\bar{\rho}^N - \bar{\rho}^C - \frac{\rho_x^C I_x^{CN} + \rho_y^C I_y^{CN}}{V^N} \right) \frac{I_x^{CN}}{V^N} \\ \frac{\partial F}{\partial \rho_y^C} &= -2 \sum_{N\in\Omega^C} \left(\bar{\rho}^N - \bar{\rho}^C - \frac{\rho_x^C I_x^{CN} + \rho_y^C I_y^{CN}}{V^N} \right) \frac{I_y^{CN}}{V^N},\end{aligned} \tag{4.8}$$

so that the system (4.7) becomes

$$\begin{aligned}\sum_{N\in\Omega^C} \left(\rho_x^C \frac{I_x^{CN}}{V^N}\frac{I_x^{CN}}{V^N} + \rho_y^C \frac{I_x^{CN}}{V^N}\frac{I_y^{CN}}{V^N} - \left(\bar{\rho}^N - \bar{\rho}^C\right) \frac{I_x^{CN}}{V^N} \right) = 0 \\ \sum_{N\in\Omega^C} \left(\rho_x^C \frac{I_y^{CN}}{V^N}\frac{I_x^{CN}}{V^N} + \rho_y^C \frac{I_y^{CN}}{V^N}\frac{I_y^{CN}}{V^N} - \left(\bar{\rho}^N - \bar{\rho}^C\right) \frac{I_y^{CN}}{V^N} \right) = 0,\end{aligned}$$

or, subdividing the sums,

$$\begin{pmatrix} a_{x,x}^C & a_{x,y}^C \\ a_{x,y}^C & a_{y,y}^C \end{pmatrix} \cdot \begin{pmatrix} \rho_x^C \\ \rho_y^C \end{pmatrix} = \begin{pmatrix} b_x^C \\ b_y^C \end{pmatrix}, \tag{4.9}$$

where

$$a_{\alpha,\beta}^C = \sum_{N\in\Omega^C} \frac{I_\alpha^{CN}}{V^N}\frac{I_\beta^{CN}}{V^N} \qquad , \qquad b_\alpha^C = \sum_{N\in\Omega^C} \left(\bar{\rho}^N - \bar{\rho}^C\right) \frac{I_\alpha^{CN}}{V^N} \tag{4.10}$$

for $\alpha, \beta \in \{x, y\}$. The unknown values of derivatives ρ_x^C and ρ_y^C are then

$$\begin{pmatrix} \rho_x^C \\ \rho_y^C \end{pmatrix} = \frac{1}{D} \begin{pmatrix} a_{y,y}^C b_x^C - a_{x,y}^C b_y^C \\ a_{x,x}^C b_y^C - a_{x,y}^C b_x^C \end{pmatrix} \qquad , \qquad D = a_{x,x}^C a_{y,y}^C - \left(a_{x,y}^C\right)^2. \tag{4.11}$$

Before we proceed one order higher, let us remark, that - depending on the problem we solve - it can be useful to preserve local bounds by restricting the slopes ρ_x^C and ρ_y^C for example with Barth-Jasperson limiter. This is described in [7].

4.2.2. First and Second Derivatives at Once. We define the functional

$$G\left(\rho_x^C, \rho_y^C, \rho_{xy}^C, \rho_{xx}^C, \rho_{yy}^C\right) = \sum_{N\in\Omega^C} \left(\bar{\rho}^N - \frac{\int\limits_N \rho^C(x,y)\ dx\ dy}{V^N} \right)^2 \tag{4.12}$$

where V^N is the volume of cell N and ρ_C is the piecewise parabolic reconstruction

$$\begin{aligned}\rho^C =& \bar{\rho}^C + \rho_x^C(x - x^C) + \rho_y^C(y - y^C) + \\ &+ \rho_{xy}^C(x - x^C)(y - y^C) + \frac{1}{2}\rho_{xx}^C(x - x^C)^2 + \frac{1}{2}\rho_{yy}^C(y - y^C)^2.\end{aligned} \tag{4.13}$$

Subdividing the integral into small parts and introducing new symbols for these parts, we get

$$G = \sum_{N\in\Omega^C} \left(R_N^C - \frac{\rho_x^C I_x^{CN} + \rho_y^C I_y^{CN} + \rho_{xy}^C I_{xy}^{CN} + \frac{1}{2}\rho_{xx}^C I_{xx}^{CN} + \frac{1}{2}\rho_{yy}^C I_{yy}^{CN}}{V^N} \right)^2, \tag{4.14}$$

where

$$
\begin{aligned}
I_x^{CN} &= \int_N \left(x - x^C\right)\ \mathrm{d}x\ \mathrm{d}y \qquad , \qquad I_{xx}^{CN} = \int_N \left(x - x^C\right)^2\ \mathrm{d}x\ \mathrm{d}y \\
I_y^{CN} &= \int_N \left(y - y^C\right)\ \mathrm{d}x\ \mathrm{d}y \qquad , \qquad I_{yy}^{CN} = \int_N \left(y - y^C\right)^2\ \mathrm{d}x\ \mathrm{d}y \\
I_{xy}^{CN} &= \int_N \left(x - x^C\right)\left(y - y^C\right)\ \mathrm{d}x\ \mathrm{d}y
\end{aligned}
\tag{4.15}
$$

and R_N^C is the difference of mean values $R_N^C = \bar{\rho}^N - \bar{\rho}^C$.

Our goal is now to compute the five unknown derivatives ρ_x^C, ρ_y^C, ρ_{xy}^C, ρ_{xx}^C and ρ_{yy}^C so, that such piecewise parabolic reconstruction is as close to the original function ρ as possible. To achieve this, we minimize the functional G, i.e. we set its derivatives with respect to unknown variables to zero, which gives the system

$$
\frac{\partial G\left(\rho_x^C, \rho_y^C, \rho_{xy}^C, \rho_{xx}^C, \rho_{yy}^C\right)}{\partial \rho_\alpha^C} = 0 \qquad , \qquad \alpha \in \{x, y, xy, xx, yy\} \tag{4.16}
$$

The derivatives are

$$
\frac{\partial G}{\partial \rho_\alpha^C} = -2 \sum_{N \in \Omega^C} \frac{I_\alpha^{CN}}{V^N} \left(R_N^C - \frac{\rho_x^C I_x^{CN} + \rho_y^C I_y^{CN} + \rho_{xy}^C I_{xy}^{CN} + \frac{1}{2}\rho_{xx}^C I_{xx}^{CN} + \frac{1}{2}\rho_{yy}^C I_{yy}^{CN}}{V^N} \right)
$$

for $\alpha \in \{x, y, xy\}$ and

$$
\frac{\partial G}{\partial \rho_\alpha^C} = - \sum_{N \in \Omega^C} \frac{I_\alpha^{CN}}{V^N} \left(R_N^C - \frac{\rho_x^C I_x^{CN} + \rho_y^C I_y^{CN} + \rho_{xy}^C I_{xy}^{CN} + \frac{1}{2}\rho_{xx}^C I_{xx}^{CN} + \frac{1}{2}\rho_{yy}^C I_{yy}^{CN}}{V^N} \right)
$$

for $\alpha \in \{xx, yy\}$. As in the previous case, we subdivide the sums and the system (4.16) becomes

$$
\begin{pmatrix}
a_{x,x}^C & a_{x,y}^C & a_{x,xy}^C & a_{x,xx}^C & a_{x,yy}^C \\
a_{x,y}^C & a_{y,y}^C & a_{y,xy}^C & a_{y,xx}^C & a_{y,yy}^C \\
a_{x,xy}^C & a_{y,xy}^C & a_{xy,xy}^C & a_{xy,xx}^C & a_{xy,yy}^C \\
a_{x,xx}^C & a_{y,xx}^C & a_{xy,xx}^C & a_{xx,xx}^C & a_{xx,yy}^C \\
a_{x,yy}^C & a_{y,yy}^C & a_{xy,yy}^C & a_{xx,yy}^C & a_{yy,yy}^C
\end{pmatrix}
\cdot
\begin{pmatrix}
\rho_x^C \\ \rho_y^C \\ \rho_{xy}^C \\ \frac{1}{2}\rho_{xx}^C \\ \frac{1}{2}\rho_{yy}^C
\end{pmatrix}
=
\begin{pmatrix}
b_x^C \\ b_y^C \\ b_{xy}^C \\ b_{xx}^C \\ b_{yy}^C
\end{pmatrix},
\tag{4.17}
$$

where

$$
a_{\alpha,\beta}^C = \sum_{N \in \Omega^C} \frac{I_\alpha^{CN}}{V^N} \frac{I_\beta^{CN}}{V^N} \qquad , \qquad b_\alpha^C = \sum_{N \in \Omega^C} \left(\bar{\rho}^N - \bar{\rho}^C\right) \frac{I_\alpha^{CN}}{V^N} \tag{4.18}
$$

for $\alpha, \beta \in \{x, y, xy, xx, yy\}$. The unknown values of derivatives ρ_x^C, ρ_y^C, ρ_{xy}^C, ρ_{xx}^C and ρ_{yy}^C are then computed by solving linear system (4.17) in the usual way. Also, similarly to the case of first derivatives, the integrals I^{CN} can be further simplified for practical computations.

4.2.3. Remarks on Computation of Integrals. In practical computations, it would be very time-consuming to use the integrals as they are defined in the left-hand part of (4.6) and in (4.15). To estimate derivatives in each cell C, one would have to integrate over all neighboring cells $N \in \Omega^C$. Furthermore, we always use the integrals divided by cell volumes. For this reason, we introduce new integrals

$$\begin{aligned} J_x^N &= \frac{1}{V^N} \int_N x \, \mathrm{d}x \, \mathrm{d}y \quad , \quad J_{xx}^N = \frac{1}{V^N} \int_N x^2 \, \mathrm{d}x \, \mathrm{d}y \\ J_y^N &= \frac{1}{V^N} \int_N y \, \mathrm{d}x \, \mathrm{d}y \quad , \quad J_{yy}^N = \frac{1}{V^N} \int_N y^2 \, \mathrm{d}x \, \mathrm{d}y \\ J_{xy}^N &= \frac{1}{V^N} \int_N xy \, \mathrm{d}x \, \mathrm{d}y \end{aligned} \tag{4.19}$$

related to the integrals from (4.6) and (4.15) by

$$\begin{aligned} \frac{I_x^{CN}}{V^N} &= J_x^N - x^C \quad , \quad \frac{I_{xx}^{CN}}{V^N} = J_{xx}^N - 2x^C J_x^N + \left(x^C\right)^2 \\ \frac{I_y^{CN}}{V^N} &= J_y^N - y^C \quad , \quad \frac{I_{yy}^{CN}}{V^N} = J_{yy}^N - 2y^C J_y^N + \left(y^C\right)^2 \\ \frac{I_{xy}^{CN}}{V^N} &= J_{xy}^N - x^C J_y^N - y^C J_x^N + x^C y^C. \end{aligned} \tag{4.20}$$

Note, that the new integrals $J_\alpha^N, \alpha \in \{x, y, xy, xx, yy\}$ depend only on values in cell N, so that they can be computed only once for each cell and stored. Then, to evaluate expression (4.10) or (4.18), we use only multiplication, addition and subtraction. Integrals J_α^N need to be recomputed only if the mesh is changed by node reconnection (or by node repositioning, which is not applicable in our process).

Another advantage of storing such normalized integrals is the simplified treatment of ghost cells. Imagine a cell degenerated into a 1D line segment. To compute J_α^N for such cell, we simply take the 1D integral over this segment and divide it by length of this line. Similarly, for a cell degenerated into a single point, we take J_x^N as the x-value in this point, J_y^N as the y-value, etc., so that we avoid division by zero. In the rest of the computation, we do not need to care whether the cell is degenerate or not, since we use only the stored values of J_α^N.

5. Numerical Results. Let us now briefly present results of two numerical tests. The first mesh subdivides the domain $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$ into 32 triangular cells. Fig. 5.1 shows rezoned meshes obtained by our swapping process with first derivatives computed in various ways. We compare the approach with analytical function $f = (x+\frac{1}{2})^3/8 + (x+\frac{1}{2})^2(y+\frac{1}{2})^2/16 + (y+\frac{1}{2})^3/8$ (shown by dashed isolines in the background) and its analytical derivatives to the results based on discrete mean values of f and numerical approximation of its derivatives. Fig. 5.2 shows results on a denser meshes (200 cells) based on piecewise constant and piecewise linear reconstruction of function $f = \cos(\pi x)\cos(\pi y)$.

Here and in all other tests we performed on various meshes, the general behavior of the swapping process is the same: as expected, all cells tend to align either parallel or perpendicular to isolines of the underlying function. This is exactly what we need in ALE simulations: the mesh geometry automatically adapts to some variable of the physical system under study. Note, that the aligning of cells is more dramatic in

the case of piecewise constant reconstruction, since the difference to the underlying function (i.e. the L_2 error of interpolation) is bigger. With the piecewise linear reconstruction, the effect is still strong enough and moreover the cells retain a satisfactory quality.

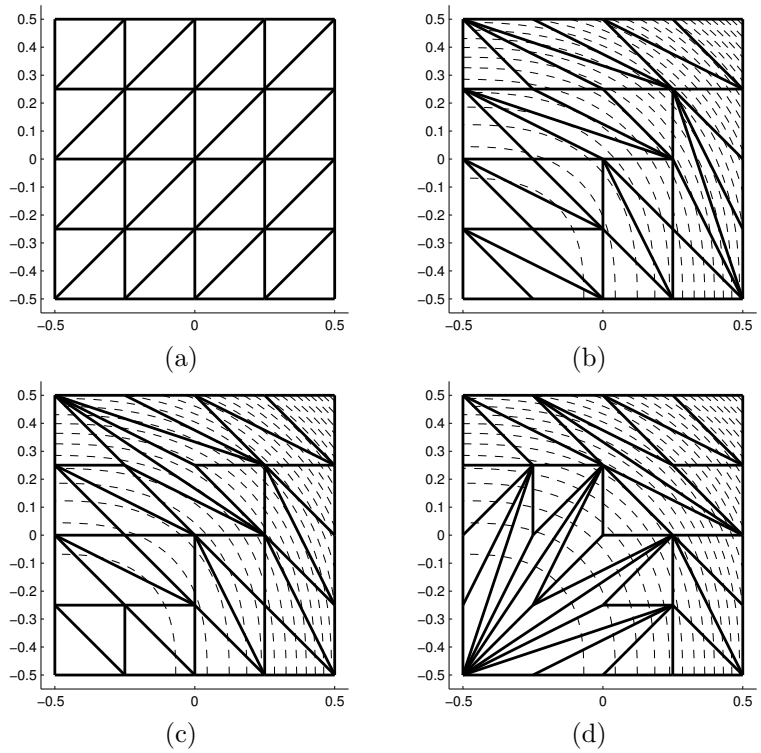

FIG. 5.1. *Test 1. The original mesh (a). Mesh after rezoning based on piecewise constant reconstruction and error estimator with analytical first derivatives (b), unlimited numerical first derivatives (c) or numerical first derivatives restricted by Barth's limiter (d). Isolines of reconstructed function f (dashed).*

6. Conclusion. As we have already mentioned, in ALE simulations it is advantageous if the mesh rezoning strategy is related to physical behavior of the system under study. This is why the proposed node reconnecting algorithm utilizes the values of a selected state variable, for example density, and minimizes the error of its piecewise constant or piecewise linear interpolation. First, we tested the idea on an analytical underlying function with known analytical derivatives. In all cases, cells of a sufficiently dense mesh aligned according to isolines of that function, which means that in this way the mesh geometry can keep track with behavior of the physical system. However, in practical ALE simulations we know only discrete values of the state variable (mostly mean values in the cells) and so we need to interpolate them and approximate the derivatives. This can be done for example by minimization of the next term in Taylor expansion, as it was suggested above. Numerical results of this discrete implementation are in good correspondence to results with analytical function, which proves that the proposed method is practically applicable.

There is no doubt that a good mesh rezoning strategy can be achieved as a combination of node reconnection with node repositioning (see e.g. [3]), but we believe

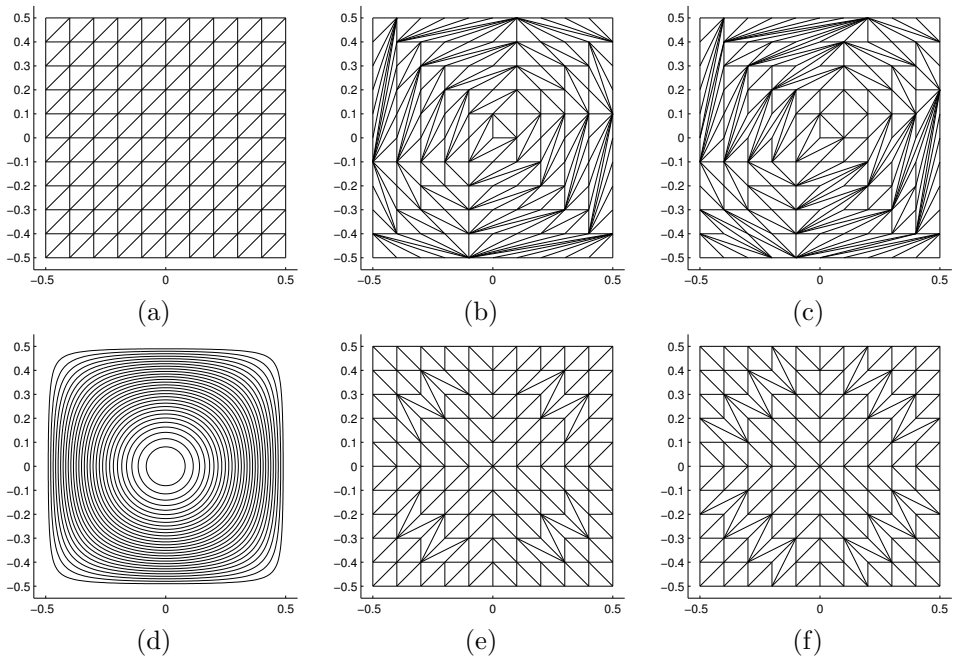

FIG. 5.2. *Test 2. The original mesh (a). Isolines of underlying function f (d). Mesh rezoned by piecewise constant reconstruction using error estimator with analytical (b) or unlimited numerical first derivatives (c). Mesh rezoned by piecewise linear reconstruction with error estimator based on analytical (e) or numerical (f) first and second order derivatives.*

that it is first necessary to understand and manage both approaches in detail in order to develop their fully controlled combination. Our belief is supported by preliminary results of the ongoing research.

Acknowledgments. The authors would like to thank M. Shashkov for fruitful cooperation and P. Knupp for valuable discussions and comments. P.V. thanks R. Liska for his support and advice during the entire work on this project.

REFERENCES

[1] C. W. HIRT, A. A. AMSDEN AND J. L. COOK, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, J. Comp. Phys., 14 (1974), pp. 227–253.

[2] M. KUCHAŘÍK, M. SHASHKOV AND B. WENDROFF, *An efficient linearity-and-bound-preserving remapping method*, J. Comp. Phys., 188 (2003), pp. 462–471.

[3] P. VÁCHAL, R. V. GARIMELLA AND M. J. SHASHKOV, *Untangling of 2D meshes in ALE simulations*, J. Comp. Phys., 196 (2004), pp. 627–644.

[4] GARIMELLA, R., *MSTK - A Flexible Infrastructure Library for Developing Mesh Based Applications*, In: Proceedings of 13th International Meshing Roundtable, Williamsburg, VA, Sep. 2004.

[5] L. G. MARGOLIN AND M. J. SHASHKOV, *Second-order-sign-preserving remapping on general grids*, Tech. Report LA-UR-02-525, Los Alamos National Laboratory, 2002.

[6] M. J. SHASHKOV, *Conservative finite-difference methods on general grids*, CRC Press, Boca Raton, FL, 1995.

[7] M. KUCHAŘÍK, R. V. GARIMELLA, M. J. SHASHKOV AND B. WENDROFF, *Efficient algorithm for local-bound-preserving remapping in ALE methods*, In: Feistauer M., Dolejsi V., Knobloch, P., Najzar, K. (eds.): Numerical Mathematics and Advanced Applications. ENUMATH 2003. pp. 815–824. Springer Verlag, Berlin - Heidelberg - New York, 2004.