Available online at www.sciencedirect.com

Journal of Computational Physics 188 (2003) 462–471

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

An efficient linearity-and-bound-preserving remapping method

Milan Kucharik [a], Mikhail Shashkov [b,*], Burton Wendroff [b]

[a] *Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague,*
Břehová 7, 115 19 Prague 1, Czech Republic
[b] *Theoretical Division, T-7, Los Alamos National Laboratory, MS-B284, Los Alamos, NM 87545, USA*

Received 4 October 2002; received in revised form 26 February 2003; accepted 12 March 2003

Abstract

In this paper we describe an efficient, local-bound-preserving conservative interpolation (remapping) algorithm, which is exact for a global linear function (linearity-preserving). The algorithm is based on reconstruction, approximate integration and mass re-distribution. We demonstrate our new algorithm on a series of numerical examples.

Keywords: Conservative interpolation; Remapping; ALE methods

1. Introduction – statement of the problem

Consider that we have a mesh that is a tessellation of some region Ω by a collection of cells $C_i,\ i = 1, \ldots, L$, that is, the cells have disjoint connected interiors and their union is Ω. This we will call the old mesh.

Next, suppose there is a new tessellation of the region with a different collection of cells $\tilde{C}_i,\ i = 1, \ldots, L$, such that each new cell $\tilde{C}_i$ is obtained by a *small* displacement of the vertices of the old cell C_i by some displacement vector field δ. We emphasize here that the old and new meshes have the same number of cells and the same connectivity.

For example, in the context of an arbitrary Lagrangian–Eulerian method the old mesh would be the result of the Lagrangian step and the new mesh would be the result of mesh modification (rezoning) (see, for example, [5]).

The situation is that the only data that the computation has at hand are, for each cell C_i, the mass for some density function $\rho(\mathbf{x})$, said mass being

[*] Corresponding author. Tel.: +1-505-667-4400; fax: +1-505-665-5757.
E-mail addresses: kucharik@karkulka.fjfi.cvut.cz (M. Kucharik), shashkov@lanl.gov (M. Shashkov), bbw@lanl.gov (B. Wendroff).

0021-9991/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00187-6

$$m_i = \int_{C_i} \rho(\mathbf{x})\,\mathrm{d}V$$

with corresponding mean value

$$\bar{\rho}_i = \frac{m_i}{|C_i|},$$

where $|C_i| > 0$ is the volume (area in 2D) of C_i.

The conservative interpolation (remapping) problem is to compute the masses $\tilde{m}_i$ and mean values of the new grid, but since the density distribution is unknown we can only approximate them. We will denote these approximate values by $\tilde{m}_i^*$.

This approximation will have to satisfy at least the following two constraints:

(I) Total mass should be conserved.

(II) Exactness for a global linear function. If $\rho(\mathbf{x})$ happens to be a linear function on Ω then the new masses $\tilde{m}_i^*$ should exactly agree with the masses of that function for the new cells.

A possible procedure for accomplishing this has two stages. First, one constructs a new distribution on the old mesh $\hat{\rho}(x)$ such that

$$\int_{C_i} \hat{\rho}(\mathbf{x})\,\mathrm{d}V = m_i$$

and if $\rho(\mathbf{x})$ is a global linear function then $\hat{\rho}(\mathbf{x}) \equiv \rho(\mathbf{x})$. Usually $\hat{\rho}(\mathbf{x})$ is a discontinuous piece-wise linear (linear on each cell C_i) function, whose local gradients have been limited in some way to satisfy additional conditions such as weak monotonicity.

For the second stage the reconstructed function $\hat{\rho}(\mathbf{x})$ is exactly integrated over the new cells. This requires finding the intersection of each new cell with the old ones. Finding the intersections is easy in one dimension, doable but computationally very expensive in two dimensions, and not practical in three dimensions. We will call any such two stage method "exact".

The goal of this paper is to present a new efficient, conservative, local-bound-preserving algorithm for remapping which does not require finding intersections between the cells of the old and new mesh, and is exact for a global linear function.

In the first stage of our new remapping method we also use a discontinuous piecewise linear $\hat{\rho}(\mathbf{x})$, but without gradient limiters.

In the second stage, the new mass is written in flux form, that is, the mass of each new cell will be set to the mass of the corresponding old cell plus terms which define exchange of mass with nearest neighbors (fluxes). Using the flux form trivially guarantees conservation of total mass. In contrast to Dukowicz and Baumgardner [3], who use exact integration to compute these fluxes, we use a quadrature, which does not require finding the intersections of new and old cells.

It may happen that the mean densities in the new cells will be out of local bounds, for example, they can be negative even if the original means were all positive. The cause of this can be: either use of unlimited gradients in the reconstruction, approximate integration, or both.

To fix out of bound mean densities we introduce a third stage – repair. In this stage masses of new cells are re-distributed in such a way that the modified new means are in the range of local bounds, and they are changed as little as possible compared with those obtained as a result of the second stage.

In the next two sections we will describe the algorithm in some detail and will give some numerical examples. In those examples for the purpose of comparison we also present some results using a particular exact method, that is, the method of limited reconstruction of Barth and Jespersen [1].

A summary of the numerical results and some discussion about the extension of the algorithm to 3D are given in Section 4.

2. The remapping algorithm

The method we are describing is by no means restricted to a logically rectangular (structured quadrilateral) grid or to 2D, but we explain it in those terms.

The grid points or vertices of a logically rectangular grid have the vertices $P_{i,j} = (x_{i,j}, y_{i,j})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. The cells $C_{i+\frac{1}{2},j+\frac{1}{2}}$ are the quadrilaterals formed by the four vertices $P_{i,j}$, $P_{i+1,j}$, $P_{i+1,j+1}$, $P_{i,j+1}$ (see Fig. 1).

2.1. Reconstruction – first stage

The first task is to construct the function $\hat{\rho}(x,y)$. We define a piecewise linear function $\hat{\rho}(x,y)$ on the old grid (with gradient (g_x, g_y), to be defined) as

$$\hat{\rho}(x,y) = \hat{\rho}_{i+\frac{1}{2},j+\frac{1}{2}} = \bar{\rho}_{i+\frac{1}{2},j+\frac{1}{2}} + g^x_{i+\frac{1}{2},j+\frac{1}{2}}\left(x - \bar{x}_{i+\frac{1}{2},j+\frac{1}{2}}\right) + g^y_{i+\frac{1}{2},j+\frac{1}{2}}\left(y - \bar{y}_{i+\frac{1}{2},j+\frac{1}{2}}\right), \quad (x,y) \in C_{i+\frac{1}{2},j+\frac{1}{2}},$$

where the centroid $(\bar{x}_{i+\frac{1}{2},j+\frac{1}{2}}, \bar{y}_{i+\frac{1}{2},j+\frac{1}{2}})$ is

$$\bar{x}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{\left|C_{i+\frac{1}{2},j+\frac{1}{2}}\right|}\int_{C_{i+\frac{1}{2},j+\frac{1}{2}}} x\,\mathrm{d}V,$$

$$\bar{y}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{\left|C_{i+\frac{1}{2},j+\frac{1}{2}}\right|}\int_{C_{i+\frac{1}{2},j+\frac{1}{2}}} y\,\mathrm{d}V,$$

and these can be computed exactly for polygons, as can the integral of any linear function (see, for example [4]). The gradient $(g^x_{i+\frac{1}{2},j+\frac{1}{2}}, g^y_{i+\frac{1}{2},j+\frac{1}{2}})$ can be obtained from any difference approximation to the gradient of $\rho(x,y)$ using neighboring old mean values as long as it is exact if the old means are the mean values of a global linear function (for examples of such approximation see [4]).

Clearly such reconstruction will be exact for global linear functions.

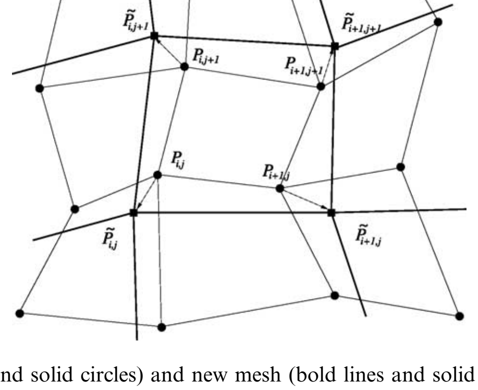

Fig. 1. Old (thin lines and solid circles) and new mesh (bold lines and solid squares); swept regions.

2.2. Flux form and swept region quadrature – stage two

2.2.1. Flux form

We now look for a finite difference equation defining the new masses $\tilde{m}^*_{i+\frac{1}{2},j+\frac{1}{2}}$ in terms of the old masses $m_{i+\frac{1}{2},j+\frac{1}{2}}$, and mass exchange with neighboring cells,

$$\tilde{m}^*_{i+\frac{1}{2},j+\frac{1}{2}} = m_{i+\frac{1}{2},j+\frac{1}{2}} + \mathscr{F}_{i+1,j+\frac{1}{2}} - \mathscr{F}_{i,j+\frac{1}{2}} + \mathscr{F}_{i+\frac{1}{2},j+1} - \mathscr{F}_{i+\frac{1}{2},j}. \tag{1}$$

Here $\mathscr{F}$ are edge fluxes, which are defined following ideas from [4] and will be described in next subsection.

2.2.2. Swept regions

Consider an oriented edge $F_{i,j+\frac{1}{2}} = \{P_{i,j+1}, P_{i,j}\}$. This edge is common to cells $C_{i+\frac{1}{2},j+\frac{1}{2}}$ and $C_{i-\frac{1}{2},j+\frac{1}{2}}$. The displacement vector field δ moves $P_{i,j}$ to its new position $\tilde{P}_{i,j}$ and $P_{i,j+1}$ to $\tilde{P}_{i,j+1}$, thereby forming the oriented quadrilateral $\delta F_{i,j+\frac{1}{2}} = (P_{i,j}, \tilde{P}_{i,j}, \tilde{P}_{i,j+1}, P_{i,j+1})$ (see Fig. 1). It is important to note that this quadrilateral can have self-intersections. This quadrilateral was called a fluxing area in [3], swept region in [4], but the notion seems to have been introduced earlier by Collela [2]. Similarly, for an oriented edge $F_{i+\frac{1}{2},j} = \{P_{i,j}, P_{i+1,j}\}$ we define the swept region as $\delta F_{i+\frac{1}{2},j} = (P_{i,j}, P_{i+1,j}, \tilde{P}_{i+1,j}, \tilde{P}_{i,j})$.

We next summarize the development in [4]. First of all, the critical function of the swept region is its *signed* area $|\delta F_{i,j+\frac{1}{2}}|$, which in turn depends on the ordering of its vertices (orientation of the boundary). It is introduced by expressing the area in terms of a line integral as follows:

$$\left|\delta F_{i,j+\frac{1}{2}}\right| = \int_{\delta F_{i,j+\frac{1}{2}}} \mathrm{d}V = \oint_{\partial\left(\delta F_{i,j+\frac{1}{2}}\right)} x\,\mathrm{d}y. \tag{2}$$

Because of the chosen orientation, for situation shown in Fig. 1, $|\delta F_{i,j+\frac{1}{2}}| < 0$.

As has been shown in [4], we can introduce *signed* integration of any polynomial function over a polygon by reducing it to a line integral similar to (2). The sign, of course, will depend on the chosen orientation. In the rest of the paper each time we integrate a linear function over a polygon we always will assume that it is a signed integration.

We define fluxes $\mathscr{F}_{i,j+\frac{1}{2}}$ in (1) as follows

$$\mathscr{F}_{i,j+\frac{1}{2}} = \int_{\delta F_{i,j+\frac{1}{2}}} \hat{\rho}_{i,j+\frac{1}{2}}(x,y)\,\mathrm{d}V,$$

where

$$\hat{\rho}_{i,j+\frac{1}{2}}(x,y) = \begin{cases} \hat{\rho}_{i+\frac{1}{2},j+\frac{1}{2}}(x,y), & |\delta F_{i,j+\frac{1}{2}}| \geqslant 0, \\ \hat{\rho}_{i-\frac{1}{2},j+\frac{1}{2}}(x,y), & |\delta F_{i,j+\frac{1}{2}}| < 0, \end{cases}.$$

In words, in the swept region the integrand is taken entirely from the cell on one side or the other of the edge, depending on the sign of the swept area. In the case shown in Fig. 1, $|\delta F_{i,j+\frac{1}{2}}| < 0$ and we use the reconstructed linear function belonging to the left side of the edge ($\hat{\rho}_{i-\frac{1}{2},j+\frac{1}{2}}(x,y)$) because most of the swept region lies inside cell $C_{i-\frac{1}{2},j+\frac{1}{2}}$.

Similar formulas are used for the other family of edges.

Formula (1) can be considered as a conservative quadrature for computing the new mass $\tilde{m}_{i+\frac{1}{2},j+\frac{1}{2}}$. In fact, in the situation shown in Fig. 1 the new cell $\tilde{C}_{i+\frac{1}{2},j+\frac{1}{2}}$ is the union of the old cell $C_{i+\frac{1}{2},j+\frac{1}{2}}$ and the four swept regions corresponding to its edges. Formula (1) means that the new mass is approximated by the integral over the old cell (which gives $m_{i+\frac{1}{2},j+\frac{1}{2}}$) plus integrals over the swept regions, where the choice of which piece of the reconstructed function to use is based on the sign of the volume of the corresponding swept area.

2.2.3. Accuracy of swept region quadrature

Since our reconstruction is exact for a global linear function, the swept region quadrature will also be exact for such a function because reconstruction is the same in all swept regions. In general, smoothness of the underlying function, typical assumptions about regularity of the grid, and exactness of the quadrature for a global linear function imply second-order accuracy [4]. For non-smooth functions we are less concerned about order of accuracy than we are about the quality of the remapped function and its absolute accuracy. This is addressed in Section 3.

2.3. Conservative mass re-distribution – stage three

The quantity which one is remapping may have specific physical meaning. For example, the given means might be derived from a concentration lying between zero and one, and for the remapped means to be physically correct they have to be in the same range – this we call *global bounds*. There are other requirements related to improving monotonicity of the remapped discrete function. For example, one can require that the value in the new cell lies between maximum and minimum old values of the corresponding old cell and its nearest neighbors – this we call *local bounds*. At this point our algorithm may create values out of range specified by local or global bounds. This is especially true for non-smooth ρ. In this section we describe a procedure based on mass re-distribution that locally adjusts the out of range values to be within the bounds.

First, choose the *bound-determining neighborhood* $\mathscr{C}_i$ for each cell C_i (we return to the notation from Section 1 and use just one index). For example, this neighborhood might consist of cell C_i itself and all of its nearest neighbors – in the case of a logically rectangular grid it will be the 3 × 3 patch with center in cell $(i+\frac{1}{2}, j+\frac{1}{2})$. The lower and upper bounds (in which the new means are allowed to be), $\rho_i^{\min}$, $\rho_i^{\max}$, are

$$\rho_i^{\min} = \min_{j\in\mathscr{C}_i} \overline{\rho}_j, \quad \rho_i^{\max} = \max_{j\in\mathscr{C}_i} \overline{\rho}_j.$$

We will use the notation $\mathscr{C}_i$ also for the set of indices of cells in the neighborhood.

In our repair procedure we first check if the new mean value, $\overline{\tilde{\rho}}_i$, is within its range, if so we do nothing. This will always be the case for a global linear function, and this implies that the algorithm continues to be exact for this function. If the new mean value is out of range we repair it. Below we give the algorithm for the case when, for some i, $\overline{\tilde{\rho}}_i < \rho_i^{\min}$. The case when $\overline{\tilde{\rho}}_i > \rho_i^{\max}$ is similar.

If

$$\overline{\tilde{\rho}}_i < \rho_i^{\min},$$

then

$$\delta m_i^{\text{needed}} = \left(\rho_i^{\min} - \overline{\tilde{\rho}}_i\right)|\tilde{C}_i|$$

is how much mass we need to add to this cell to bring the new value up to its lower bound. Because our method has to be conservative, this needed mass has to be taken from neighboring cells. Here we start the search in the bound-determining neighborhood. First, we need to check how much total mass we can take from all cells in the neighborhood without violating their lower bounds. To do this for each cell in the neighborhood, that is, for each $j \in \mathscr{C}_i$, we compute how much mass can safely be taken from this cell, which is

$$\delta m_j^{\text{avail}} = \max\left((\overline{\tilde{\rho}}_j) - \rho_j^{\min})|\tilde{C}_j|, 0\right),$$

so that the total mass available is

$$\delta m_{\text{total}}^{\text{avail}} = \sum_{j\in\mathscr{C}_i} \delta m_j^{\text{avail}}.$$

If there is enough available mass in neighboring cells to provide the mass needed for cell $\tilde{C}_i$, that is, if

$$\delta m_i^{\text{needed}} \leqslant \delta m_{\text{total}}^{\text{avail}},$$

then the mass and corresponding density in cell $\tilde{C}_i$ are brought up to their lower bounds. That is, we set

$$\tilde{m}_i' = \rho_i^{\min}|\tilde{C}_i|, \quad \bar{\tilde{\rho}}_i' = \rho_i^{\min}.$$

All other masses in the neighborhood are decreased proportionally to the mass available in the corresponding cell, that is,

$$\tilde{m}_j' = \tilde{m}_j - \frac{\delta m_j^{\text{avail}}}{\delta m_{\text{total}}^{\text{avail}}} \delta m_i^{\text{needed}},$$

and clearly the total mass of cell $\tilde{C}_i$ and its neighbors remains unchanged. For the densities $\bar{\tilde{\rho}}_j'$ this implies

$$\bar{\tilde{\rho}}_j' = \left(1 - \frac{\delta m_i^{\text{needed}}}{\delta m_{\text{total}}^{\text{avail}}}\right)\bar{\tilde{\rho}}_j + \frac{\delta m_i^{\text{needed}}}{\delta m_{\text{total}}^{\text{avail}}} \rho_j^{\min},$$

This completes repair of cell $\tilde{C}_i$.

If

$$\delta m_i^{\text{needed}} > \delta m^{\text{avail}},$$

that is, not enough mass is available in neighboring cells to provide the needed mass, then the neighborhood is extended and the process is repeated.

We now prove that this process will terminate successfully in a finite number of steps if the following assumption holds:

$$\tilde{C}_i \cap C_j \neq \emptyset \quad \text{if and only if } j \in \mathscr{C}_i, \tag{3}$$

that is, $\tilde{C}_i$ is completely covered by the bound-determining neighborhood of cell C_i.

If the bound-determining neighborhood is specified then this condition can be considered as an assumption on the displacement field. In other words, allowable displacements have to be compatible with the definition of bound-determining neighborhood.

Let us now assume that there is at least one new cell that violates its lower bound. Define the following quantities:

$$\Delta M^- = \sum_{i:\bar{\tilde{\rho}}_i < \rho_i^{\min}} \left(\rho_i^{\min} - \bar{\tilde{\rho}}_i\right)|\tilde{C}_i|,$$

$$\Delta M^+ = \sum_{i:\bar{\tilde{\rho}}_i > \rho_i^{\min}} \left(\bar{\tilde{\rho}}_i - \rho_i^{\min}\right)|\tilde{C}_i|.$$

Then $\Delta M^- > 0$ is the total amount of mass that is needed to bring all new densities that are lower than their lower bounds up to their lower bounds, $\rho_i^{\min}$, and ΔM^+ is the total amount of mass that can safely be taken from other cells. For repair to be successful we need to prove that

$$\Delta M^+ \geqslant \Delta M^-.$$

In fact, the total mass M can be expressed as follows:

$$\begin{aligned} M &= \sum_i \bar{\tilde{\rho}}_i |\tilde{C}_i| = \sum_i \rho_i^{\min} |\tilde{C}_i| + \sum_{i:\bar{\tilde{\rho}}_i > \rho_i^{\min}} \left(\bar{\tilde{\rho}}_i - \rho_i^{\min}\right) |\tilde{C}_i| - \sum_{i:\bar{\tilde{\rho}}_i < \rho_i^{\min}} \left(\rho_i^{\min} - \bar{\tilde{\rho}}_i\right) |\tilde{C}_i| \\ &= \sum_i \rho_i^{\min} |\tilde{C}_i| + \Delta M^+ - \Delta M^-. \end{aligned}$$

Therefore, using assumption (3) and the definition of $\rho_i^{\min}$

$$\begin{aligned} \Delta M^+ - \Delta M^- &= M - \sum_i \rho_i^{\min} |\tilde{C}_i| = M - \sum_i \sum_{j \in \mathscr{C}_i} \rho_i^{\min} |\tilde{C}_i \cap C_j| \geqslant M - \sum_i \sum_{j \in \mathscr{C}_i} \bar{\rho}_j |\tilde{C}_i \cap C_j| \\ &= M - \sum_i \sum_j \bar{\rho}_j |\tilde{C}_i \cap C_j| = M - \sum_j \bar{\rho}_j \sum_i |\tilde{C}_i \cap C_j| = M - \sum_j \bar{\rho}_j |C_j| = M - M = 0. \end{aligned}$$

That is, $\Delta M^+ \geqslant \Delta M^-$ and it is always possible to successfully complete repair by extending the repair stencil.

It is worth repeating here that the complete algorithm (including repair) is exact for global linear functions.

3. Tests

Our algorithm can be indicated by the key words "unlimited, swept, repair", so we will call it the USR algorithm. For comparison purposes we also present results for the following similar algorithms: USN, meaning unlimited reconstruction, swept region approximate integration, and no repair; and as a reference we use BE, meaning limited reconstruction of Barth and Jespersen and exact integration.

To test the behavior of our new method we performed a series of remaps on a sequence of grids for several given functions. The initial grid is uniform in the unit square $[0,1] \times [0,1]$, while successive grids are alternately a small random displacement of the uniform grid and a return to the uniform grid. We always perform an even number of remaps, thereby returning to the original uniform grid. Each successive problem consists of a remapping from the remapped data on the previous grid to the next grid in the sequence. This process allow us to investigate the accumulation of the error, which is defined as the difference between the original and final means. When refining the grid we increase the number of remappings such that the total displacement stays approximately the same.

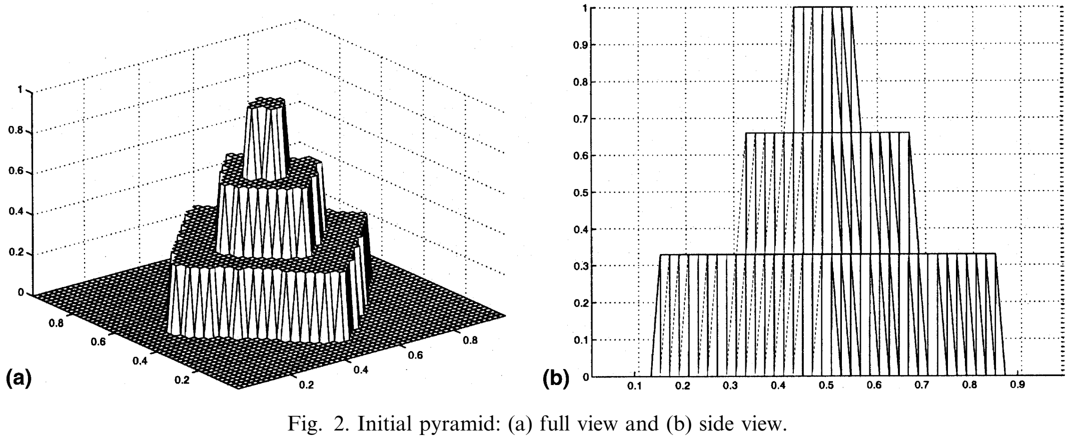

Fig. 2. Initial pyramid: (a) full view and (b) side view.

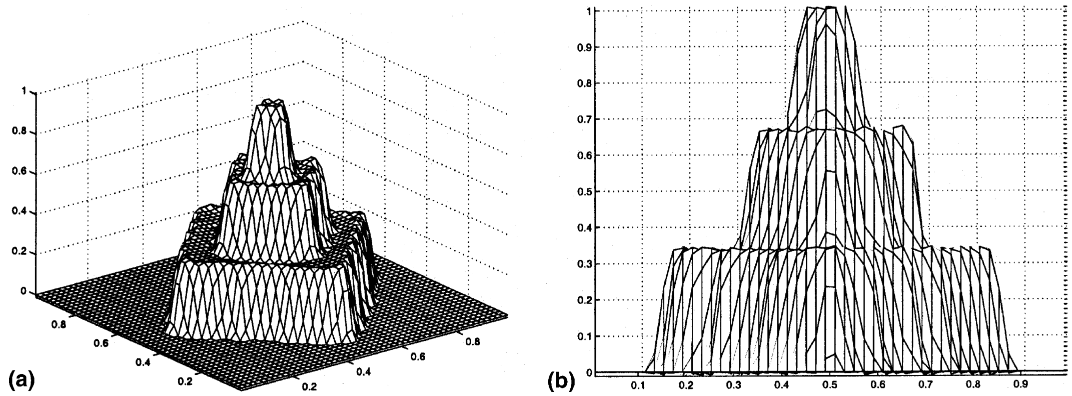

Fig. 3. Unlimited, swept, no repair – (USN): (a) full view and (b) side view.

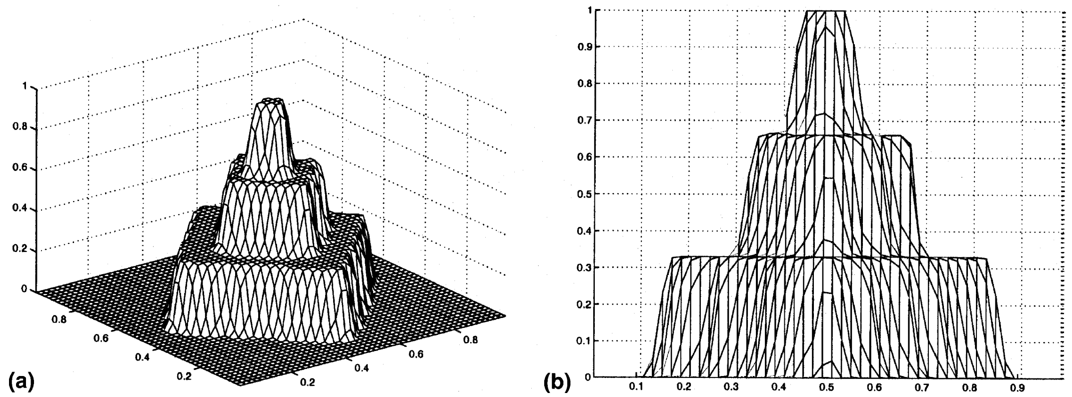

Fig. 4. Unlimited, swept, repair – USR: (a) full view and (b) side view.

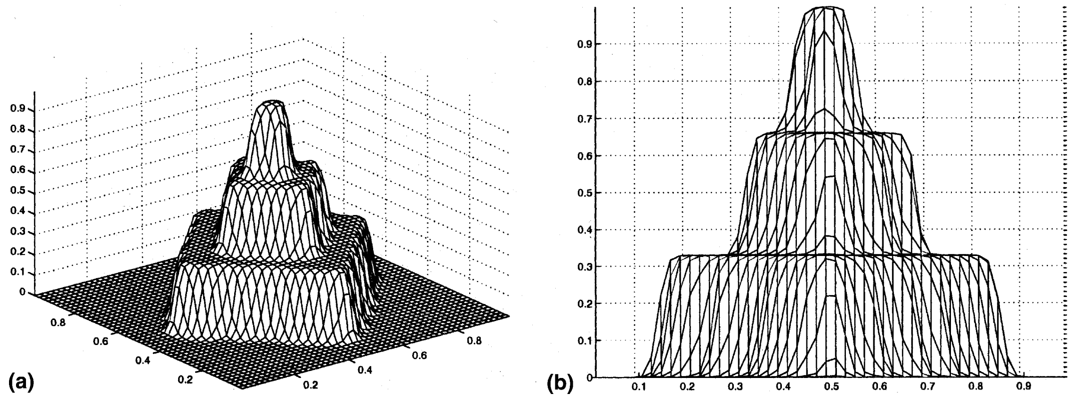

Fig. 5. BE exact method: (a) full view and (b) side view.

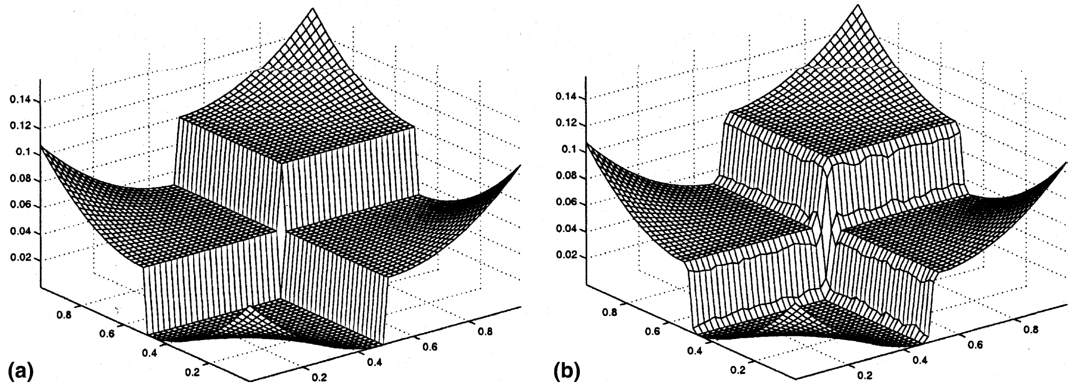

Fig. 6. Piecewise quadratic: (a) initial and (b) after 10 maps.

First, we remapped a smooth quartic $(x-.5)^4+(y-.5)^4$ using the new USR algorithm, with grids 50×50 with 10 remaps, 100×100 with 20 remaps, 200×200 with 40 remaps. The corresponding errors in the maximum norm are 4.10e − 5, 1.19e − 5, 3.56e − 6. In the L_1 norm the errors are 5.38e − 6, 1.48e − 6, 3.89e − 7. This numerically shows second-order convergence for this smooth function in both norms for small random grid displacements.

The next test remaps a cubical pyramid, Fig. 2, on a 50×50 grid, using the three methods indicated above. This function varies between 0 and 1. Specifically, let $d(x,y)=|x-.5|+|y-.5|$. If $d(x,y)>.38$ $\rho=0$, if $.2<d(x,y)\leqslant .38$ $\rho=.33$, if $.08<d(x,y)\leqslant .2$ $\rho=.66$, if $d(x,y)\leqslant .08$ $\rho=1$. Note that without any limiters or mass re-distribution, i.e., the USN method, some values will go out of range and there will be oscillations, which can be seen in Fig. 3. The minimal value actually is −1.17e − 2 and the maximum is 1.01 The L_1 norm of the error in this case is 1.02e − 2.

Now we can see in Fig. 4 that this problem is fixed by our method (USR), which differs from USN only by adding repair. The L_1 norm of the error in this case is 9.39e − 3. Only about 1.5% of the cells actually required repair.

For comparison, the result of the exact method (BE) is shown in Fig. 5. The L_1 norm of the error in this case is 1.14e − 2. As expected, the values are within bounds, but also the shape appears to have been rounded off more than with USR.

We also used our USR algorithm for a discontinuous quadratic pagoda shape, Figs. 6(a) and (b). The L_1 norm of the error in this case is 6.00e − 4, while BE for this problem had an error of 7.91e − 4.

4. Conclusion

The conclusion from these runs is that our algorithm has the smallest L_1 error and is less dissipative. Many other tests were performed on these and other problems with more remappings and different movement of the grids. All confirm that the error with USR is most often smallest or else differs very little from the smallest; the ability to achieve this without the expense of computing intersections of the old and new cells is a clear advantage of the new method.

Concerning an extension of our algorithm to 3D, the main technical difficulty is that faces of cells may not be flat. The simplest example of such a cell is a distorted brick (which is an extension of quadrilateral to 3D, see [6]). For such a cell the definition of the faces is not unique. However, for our algorithm this is not that important, as long as a consistent definition of face is given. For example, face can be defined as a union of two triangles. Now the cells are polyhedra, and all integrals needed in the algorithm can be computed exactly.

Acknowledgements

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/BES Program in the Applied Mathematical Sciences and the Laboratory Directed Research and Development program (LDRD). M. Shashkov also acknowledges the partial support of DOE's Accelerated Strategic Computing Initiative (ASCI). The authors thank L. Margolin, B. Swartz, R. Liska and M. Berndt for fruitful discussions and constructive comments. The authors thank the referees for their comments that led to significant improvements of the original manuscript.

References

[1] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: C. Rohde, D. Kroner, M. Ohlberger (Eds.), An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws, Berlin, Lecture Notes in Computational Science and Engineering, Springer, Berlin, 1997.

[2] P. Collela, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 171–200.

[3] J. Dukowicz, J. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (2000) 318–335.

[4] L. Margolin, M. Shashkov, Second-order sign-preserving remapping on general grids, J. Comput. Phys. 184 (2003) 266–298.

[5] L. Margolin, P. Knupp, M. Shashkov, Reference jacobian optimization-based rezone strategies for arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 176 (2002) 93–128.

[6] O.C. Zienkiewicz, R.L. Taylor, in: The Finite Element Method, fifth ed., The Basis, vol. 1, Butterworth–Heinemann, Oxford, 2000, pp. 134–135.