43rd AIAA Aerospace Sciences Meeting, Jan. 10–13, 2005, Reno, NV

Analysis of Slope Limiters on Irregular Grids

Marsha Berger*
NYU, 251 Mercer St., NY, NY 10012

Michael J. Aftosmis†
NASA Ames, Moffett Field, CA 94035

Scott M. Murman‡
ELORET Corp., Moffett Field, CA 94035

This paper examines the behavior of flux and slope limiters on non-uniform grids in multiple dimensions. Many slope limiters in standard use do not preserve linear solutions on irregular grids impacting both accuracy and convergence. We rewrite some well-known limiters to highlight their underlying symmetry, and use this form to examine the properties of both traditional and novel limiter formulations on non-uniform meshes. A consistent method of handling stretched meshes is developed which is both linearity preserving for arbitrary mesh stretchings and reduces to common limiters on uniform meshes. In multiple dimensions we analyze the monotonicity region of the gradient vector and show that the multidimensional limiting problem may be cast as the solution of a linear programming problem. For some special cases we present a new directional limiting formulation that preserves linear solutions in multiple dimensions on irregular grids. Computational results using model problems and complex three-dimensional examples are presented, demonstrating accuracy, monotonicity and robustness.

I. Introduction

For many finite-difference or finite-volume computations with discontinuous solutions limiters are a necessary evil. On one hand, they suppress oscillations and maintain monotonicity. On the other hand, they reduce accuracy and can hamper convergence, particularly for multidimensional unstructured grids.[1] Non-smooth limiter formulations lead to limiter chatter, even in near constant portions of the flow field, making it difficult to achieve steady state either with time marching or Newton's method.[2] Overly-compressive limiters in some cases cause staircasing of oblique shocks. These problems occur on both structured and unstructured grids, in both flux and slope-limiter formulations.

Most of the research on limiters has been in one space dimension, where a reliable theory exists.[3] There is substantially less literature on extensions to two or more dimensions, where the existing theory has proven less useful for developing numerical methods. There is almost no discussion on the application of limiters to irregular or non-uniform grids (with some exceptions[4–6]) although nearly all meshes on real geometries fall into this category. Note that ENO and WENO schemes[7] avoid the use of limiters by expending the effort to determine a non-oscillatory gradient.

The focus of this paper is a framework for analyzing limiter formulations on irregular grids, including 1-D cell stretching and general polyhedra in 3-D. This analysis provides conditions under which slope limiters

*berger@cims.nyu.edu. Member AIAA.
†Research Scientist, maftosmis@mail.arc.nasa.gov. Senior Member AIAA.
‡Senior Research Scientist, smurman@mail.arc.nasa.gov. Member AIAA.

Copyright © 2005 by the American Institute of Aeronautics and Astronautics, Inc. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

preserve monotonicity, and exactly reproduce linear solutions (called k-exact for k=1). These conditions are demonstrated as necessary for both solution accuracy, and reducing limiter chatter. With this framework, an understanding of methods currently in practice for irregular grids, along with novel limiter formulations, are developed.

The use of multi-level Cartesian grids with embedded boundaries[8] motivates the development. These grids provide an attractive numerical testbed, since they include regions of uniform grid cells, mesh interfaces with 2:1 mesh stretching, as well as general irregular polyhedra at the embedded boundaries. Examples of monotone, linearity-preserving limiters for all three cell types are included. These limiter examples are used in 3-D numerical experiments to demonstrate the behavior in practical cases.

After a brief review of common flux and slope limiters in one dimension, we develop a symmetric formulation of several common limiters in Section 2. This framework is used to analyze linearity preservation and monotonicity on 1-D stretched-meshes in Section 3. Section 4 extends these ideas to multiple dimensions by formulating slope limiters as the solution of a linear-programming problem. Numerical experiments on 3-D problems demonstrate the accuracy, convergence, and robustness of example limiters in Section 5. We include 2 appendices. Appendix A examines face-based limiters and the common approach of flooring non-physical reconstructed values. Appendix B contains an accuracy study on one-dimensional non-uniform grids to support the main text.

II. One-Dimensional Preliminaries

A. Flux Limiters

The most common framework for looking at limiters was systematized by Sweby[9] using the flux formulation. By analogy with Flux Corrected Transport,[10] one writes a second-order scheme as a first-order scheme plus a limited anti-diffusive flux, shown here for the case of linear advection $u_t + au_x = 0$, $a > 0$, as

$$u_i^{n+1} = u_i^n - \lambda(u_i - u_{i-1}) - (\psi_i F_{i+1/2} - \psi_{i-1} F_{i-1/2})$$

where the anti-diffusive flux is $F_{i+1/2} = \frac{1}{2}\lambda(1-\lambda)(u_{i+1}-u_i)$. Here the ψ_i's are the flux limiters, and $\lambda = \frac{a\Delta t}{\Delta x}$.

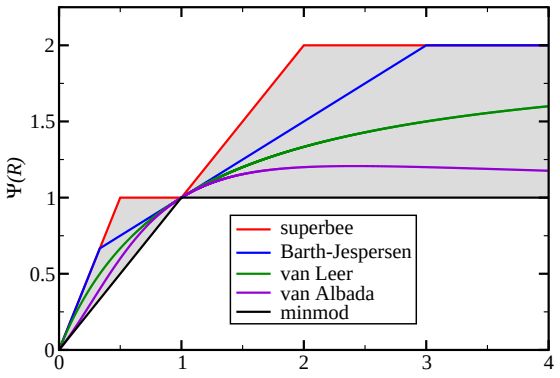


Figure 1. The shaded region is the second order TVD region from.[9] Common flux limiters ψ from (3) are shown lying in the Sweby region.

Define the ratio R of forward to backward differences in the solution (note that R is the inverse of Sweby's r) as

$$R_i = \frac{u_{i+1} - u_i}{u_i - u_{i-1}}. \tag{1}$$

By choosing $\psi_i = \psi(R_i)$ properly, a total-variation-diminishing (TVD) scheme can be achieved. Sweby derives the well-known result

$$0 \le \left(\frac{\psi(R)}{R}, \psi(R)\right) \le 2.$$

This is often expressed graphically for the different possible limiters using the *Sweby diagram* shown in Figure 1: choosing ψ in the shaded region leads to a second-order TVD solution. An important point to note is that for second order accuracy away from extrema, the limiter must satisfy $\psi(1) = 1$. In other words, the method should not do any limiting when the solution is a linear function. Also ψ must be symmetric,

$$\frac{\psi(R)}{R} = \psi(\frac{1}{R}), \tag{2}$$

although this symmetry is not immediately apparent in Figure 1. Later we rewrite the limiters in a more transparently symmetric form. In (3) we list several limiters of interest which satisfy the symmetry relation (2), in order of increasing dissipation for $R \geq 0$. For all limiters, a negative value of R indicates an extremum in the solution at cell i, and all set $\psi(R \leq 0) = 0$. For $R \geq 0$, the limiters are

$$\begin{array}{rrcl} \text{(superbee)} & \psi_{sb}(R) & = & \min[\max\{1, R\}, 2, 2R] \\ \text{(Barth} - \text{Jespersen)} & \psi_{BJ}(R) & = & \frac{1}{2}(R+1)\min\{\min(1, \frac{4R}{R+1}), \min(1, \frac{4}{R+1})\} \\ \text{(van Leer)} & \psi_{VL}(R) & = & \frac{2R}{R+1} \\ \text{(van Albada)} & \psi_{VA}(R) & = & \frac{R^2+R}{R^2+1} \\ \text{(min)} & \psi_{min}(R) & = & \min\{1, R\}. \end{array} \tag{3}$$

In Figure 1, superbee follows the boundary of the TVD region. It is easy to show that while it remains monotone it can actually steepen the gradient. The superbee and Barth-Jespersen limiters are the most compressive, and are known to turn smooth waves into square waves. In multiple dimensions their overly compressive nature may lead to staircasing of discontinuities that are not aligned with the grid. Since superbee is impractical for most real applications, we will not discuss it further.

B. Slope Limiters

Spekreijse[11] shows the equivalence of flux limiting with slope limiting, which is more natural for finite volume schemes in *REP* form:[12] *Reconstruct* the solution from its cell averages, *Evolve* the reconstructed solution from time t_n to t_{n+1}, and *Project* the solution back onto the grid to update the integral cell averages at the new time. We can see this by computing states in the flux form for linear advection, using the one-sided differences from the Fromm scheme. Computing the states at the cell interfaces gives

$$\begin{aligned} u^L_{i+1/2} &= u_i + \frac{1}{2}\psi(R_i)\,(u_i - u_{i-1}) \\ u^R_{i-1/2} &= u_i - \frac{1}{2}\psi(1/R_i)\,(u_{i+1} - u_i) \end{aligned} \tag{4}$$

where $u^L_{i+1/2}$ is the left state at the right edge, etc. Requiring that this be equivalent to reconstructing with a single limited gradient in cell i, and using the central difference $(u_{i+1} - u_{i-1})/2h$, gives

$$\begin{aligned} u^L_{i+1/2} &= u_i + \frac{1}{2}\phi(R_i)\ (u_{i+1} - u_{i-1})/2 \\ u^R_{i-1/2} &= u_i - \frac{1}{2}\phi(1/R_i)(u_{i+1} - u_{i-1})/2 \end{aligned} \tag{5}$$

Thus using (2) we get the relationship between a flux limiter ψ and a slope limiter ϕ

$$\psi(R) = \phi(R)(\frac{R+1}{2}). \tag{6}$$

Slope limiters make the reconstruction step easy, by writing the reconstructed function $U_i(x)$ in cell i as $U_i(x) = u_i + \phi(R_i)\nabla u_i(x - x_i)$, where u_i is the cell average. Note that for any limiter ψ satisfying (2), ϕ satisfies the slope form of the symmetry condition

$$\phi(R) = \phi(\frac{1}{R}). \tag{7}$$

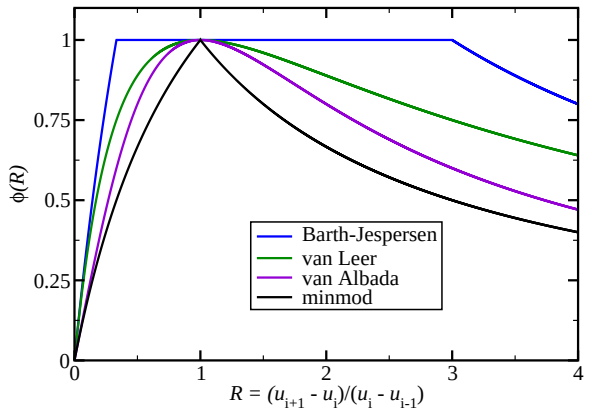


Figure 2. Common limiters in slope form from (8).

Figure 2 shows the behavior of the same collection of limiters as Figure 1 (except for superbee) but this time in their slope form. The common limiters in slope form, simplifying the expressions for monotone data so that $R \geq 0$, are:

$$\begin{aligned} \phi_{BJ}(R) &= \min\{1, \frac{4}{R+1}, \frac{4R}{R+1}\} \\ \phi_{VL}(R) &= \frac{4R}{(R+1)^2} \\ \phi_{VA}(R) &= \frac{2R}{R^2+1} \\ \phi_{min}(R) &= \min\{\frac{2}{1+R}, \frac{2R}{1+R}\}. \end{aligned} \tag{8}$$

All the limiters should be zero at extrema, when $R \leq 0$.

C. Symmetric Form of Slope Limiters

While slope limiters in the form of Figure 2 are helpful, they are somewhat opaque since R can go off to infinity and the symmetry expressed in (2) and (7) is not obvious. Here we present a more intuitive form which recovers this symmetry and will help in developing and analyzing improved formulations for non-uniform meshes.

In Figure 3 limiters are plotted using the independent variable $f = (u_i - u_{i-1})/(u_{i+1} - u_{i-1})$. As the solution u_i varies from u_{i-1} to u_{i+1}, f varies between 0 and 1. This graphical form more clearly relates the location of u_i to the limiter value. This form has the added benefit that we can more easily see the behavior of the limiter for large R.

To plot the limiters as a function of f, define $w(f) = \phi(R(f))$ and use the symmetry property (7) to get

$$w(f) = \phi(R(f)) = \phi(\frac{1-f}{f}) = \phi(\frac{1}{R}) = \phi(\frac{f}{1-f}) = w(1-f) \tag{9}$$

showing that on uniform grids the graph should be symmetric about $f = \frac{1}{2}$. This is clearly seen in Figure 3. On uniform meshes $f = \frac{1}{2}$ corresponds to linear data for u.

It becomes obvious that it is the neighbor of u_i with the larger jump, either u_{i+1} or u_{i-1} that causes the limiting of u_i's slope, since it is *this* jump that would cause the overshoot at the *opposite* neighbor's face.

In these variables the common limiters become

$$\begin{aligned} w_{BJ}(f) &= \min\{1, 4f, 4(1-f)\} \\ w_{VL}(f) &= 4f(1-f) \\ w_{sin}(f) &= \sin(\pi f) \\ w_{VA}(f) &= \frac{2f(1-f)}{f^2+(1-f)^2} \\ w_{min}(f) &= \min\{2f, 2(1-f)\}. \end{aligned} \tag{10}$$

This symmetric form also makes it easy to devise new limiters. For example, included in Figure 3(a) and equation (10) is a new *sin* limiter. In the symmetric form, it is

$$w_{sin}(1-f) = \sin(\pi - \pi f) = -\sin(-\pi f) = w_{sin}(f). \tag{11}$$

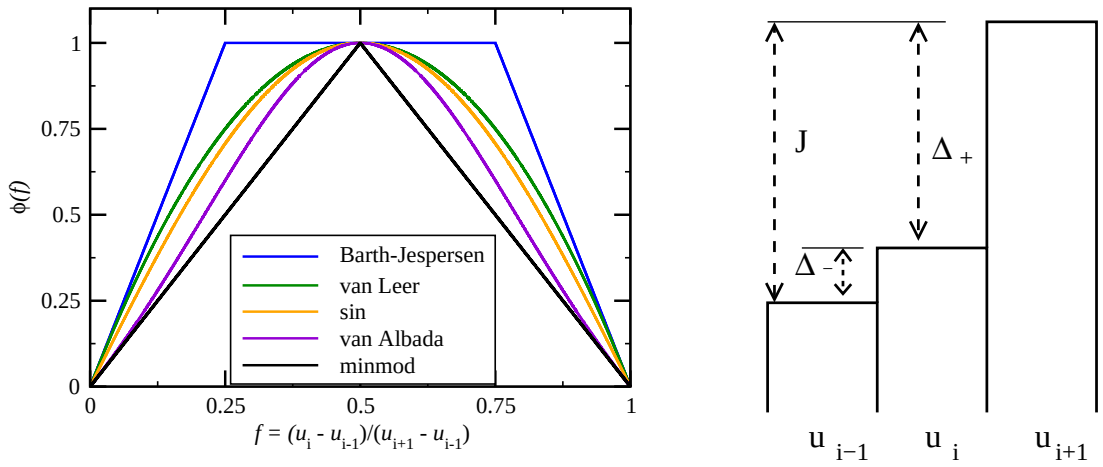


Figure 3. (a) Symmetric form of slope limiters. (b) Cell averages corresponding to the limiters. When u_i is exactly in the middle, the data is linear, $f = .5$, and the limiter is 1. The undivided differences are $\Delta_- = u_i - u_{i-1}$, and $\Delta_+ = u_{i+1} - u_i$. J is the total jump $u_{i+1} - u_{i-1}$.

In the usual notation

$$\phi_{sin} = \sin(\frac{\pi R}{R+1}) = \sin(\frac{\pi(R+1)}{R+1} - \pi R + 1) = \sin(\frac{\pi}{R+1}). \tag{12}$$

which is also symmetric according to (7). This limiter is continuously differentiable, does not limit linear data, satisfies the symmetry property (7), and falls between the van Leer and van Albada limiters in terms of dissipation.

III. Limiters on Non-Uniform Meshes

Monotone solutions on non-uniform grids require that both gradients and limiters sense the mesh irregularity. We use the symmetric form of limiters to highlight pitfalls in the standard formulation and to generalize the limiters of Eq. 10 for non-uniform grids without sacrificing smoothness, monotonicity, or linearity preservation.

A. TVD Condition on Non-Uniform Meshes

The TVD condition on uniform meshes in one dimension provides slope conditions under which the variation in the solution at a given time does not increase at the next time step. These conditions are: at an extremum the gradient is set to zero; when the solution in a cell is reconstructed to each of its cell edges, it should be limited such that its magnitude does not exceed the adjacent neighbor's cell-centered solution. The latter is often stated as the gradient should not exceed twice the forward or backward difference, if they are all the same sign. On uniform grids this is equivalent to the algebraic definition of the Barth-Jespersen limiter in (8), which can be seen by substituting (1) for R and simplifying. This geometric limiting was first proposed by van Leer,[13] but it is not what is known now as the van Leer limiter. In some communities it is known as monotonized central differences. This geometric interpretation also guarantees that if the neighboring cells satisfy physical constraints, for example that density remains positive, then the reconstructed solution will satisfy the same constraints.

In this section we briefly derive the TVD condition for a non-uniform grid in one dimension, and show that the geometric condition still holds. What changes is that the bounding quantities are no longer twice the forward and backward differences. To show this, look at $u_t + au_x = 0$, for $a > 0$. We will use the simple discretization of Forward Euler in time, however the same TVD property holds for the second order in time MUSCL scheme with slightly more complicated inequalities. The upwind finite volume scheme with second order differencing in space on a non-uniform grid is

$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{h_i}[u_{i+1/2}^L - u_{i-1/2}^L]. \tag{13}$$

Here the superscript L indicates the upwind flux is evaluated using the reconstructed state on the left of the interface,

$$u_{i+1/2}^L = u_i + \frac{h_i}{2}S_i, \tag{14}$$

where $S_i = \phi_i D_i$ is the BJ-limited slope at cell i, and D_i is the approximate derivative in that cell.

To show a scheme is TVD, we show first that a monotone increasing solution remains monotone, and second, if u_i^n is a local maximum, then $u_i^{n+1} \leq u_i^n$. The monotone decreasing and local minimum case is analogous.

Given monotonically increasing data $u_{i-1}^n \leq u_i^n \leq u_{i+1}^n$, at the left state we have

$$u_{i-1/2}^L = u_{i-1} + \frac{h_{i-1}}{2}S_{i-1} \leq u_i^n. \tag{15}$$

Similarly, $u_{i+1/2}^L \geq u_i^n$, since the slope at cell i is positive. From (13) this gives

$$u_i^{n+1} \leq u_i^n - \frac{a\Delta t}{h_i}[u_i^n - u_i^n] = u_i^n \tag{16}$$

To bound u_i^{n+1} from below we use the relation $u_i - \frac{h_i}{2}S_i \geq u_{i-1}$ Starting from (13), abbreviating $\lambda_i = \frac{a\Delta t}{h_i}$, and using $S_{i-1} \geq 0$ we get

$$\begin{aligned} u_i^{n+1} &= u_i^n - \lambda_i[u_i^n + \frac{h_i}{2}S_i - (u_{i-1}^n + \frac{h_{i-1}}{2}S_{i-1})] \\ &\geq u_i^n - \lambda_i(u_i^n + u_i^n - u_{i-1}^n) + \lambda_i u_{i-1}^n \\ &= (1 - 2\lambda_i)u_i^n + 2\lambda_i u_{i-1}^n \\ &\geq u_{i-1}^n \end{aligned} \tag{17}$$

Equation (17) holds since the region of stability for this scheme is $\lambda_i \leq 1/2$, so the coefficients are positive. Hence for monotone increasing data the values at time $n+1$ interlace with those at time n so that

$$u_{i-1}^{n+1} \leq u_{i-1}^n \leq u_i^{n+1} \leq u_i^n ... \tag{18}$$

and the solution remains monotone at the next time step.

To show that maxima decrease, let cell i be a local maximum, so that $u_i^n > u_{i-1}^n$ and $u_i^n > u_{i+1}^n$. Then the limited gradient $S_i = 0$. Equation (13) becomes $u_i^{n+1} = u_i^n - \frac{a\Delta t}{h_i}[u_i^n - u_{i-1}^n - \frac{h_{i-1}}{2}S_{i-1}]$, with $S_{i-1} \geq 0$. For the maximum to be non-increasing the terms in brackets must be positive since a is positive, so that $[u_i^n - u_{i-1}^n - \frac{h_{i-1}}{2}S_{i-1}] \geq 0$, or $S_{i-1} \leq \frac{2(u_i - u_{i-1})}{h_{i-1}}$, which we have already seen above. On non-uniform meshes this is not equivalent to the backward difference at cell i, since the mesh spacing is not the average of the two cells' mesh widths. Thus, the TVD condition that the limited slope must satisfy for monotonically increasing data on non-uniform meshes is

$$S_i = \phi_i D_i = \min\left\{\frac{2(u_{i+1} - u_i)}{h_i}, \frac{2(u_i - u_{i-1})}{h_i}\right\} \tag{19}$$

where D_i is the discrete approximation to the gradient ∇u_i.

B. Least Squares Gradient on Non-Uniform Meshes

On non-uniform grids the first question is what slope to actually limit. After all, there are many generalizations to a uniform grid central difference gradient. On irregular or unstructured meshes, a common procedure that extends to multiple dimensions is to approximate the gradient at cell i using a least squares fit to the solution in neighboring cells.[14] Least squares yields first-order slope estimates which are sufficient for second-order accuracy overall. Additionally, even on distorted meshes, least squares gradients remain exact for linear data. Limiting, however, can destroy this property.

In one space dimension the least squares gradient is especially simple. Let x_{i-1}, x_i and x_{i+1} be the centers of three adjacent cells on an irregular mesh with approximate solution u_{i-1}, u_i and u_{i+1}. The least squares problem is to find the slope D_L in cell i to minimize the residual of

$$\begin{bmatrix} x_{i+1} - x_i \\ x_{i-1} - x_i \end{bmatrix} [D_{Li}] = \begin{bmatrix} u_{i+1} - u_i \\ u_{i-1} - u_i \end{bmatrix}. \tag{20}$$

Forming the normal equations and solving for D_{Li} results in the approximation

$$D_{Li} = \frac{h_+^2}{h_+^2 + h_-^2} D_+ u_i + \frac{h_-^2}{h_+^2 + h_-^2} D_- u_i \tag{21}$$

where $h_+ = x_{i+1} - x_i$, $h_- = x_i - x_{i-1}$, $D_+ u_i = (u_{i+1} - u_i)/h_+$, etc. Note that the least squares gradient (21) is a convex combination of the forward and backward differences computed on the irregular mesh. If the mesh is uniform, $h_+ = h_-$, the weights become 1/2 each, and we recover the familiar second-order centered first derivative stencil

$$D_c u_i = \frac{(D_+ u_i + D_- u_i)}{2} = \frac{u_{i+1} - u_{i-1}}{2h} \tag{22}$$

On a non-uniform mesh a second order accurate gradient approximation is of course possible, but it does not minimize the least squares error of the linear fit. The second order non-uniform approximation is a linear combination with different weights, $D_i = \frac{h_+}{h_+ + h_-} D_+ u_i + \frac{h_-}{h_+ + h_-} D_- u_i$. However in multiple dimensions this approach is much more expensive.

C. Limiting for Linear Data

If mesh stretching is not accounted for in the limiter formulations, unexpected limiting can reduce the solution accuracy. For many limiters in common use today, (with notable exceptions[4,15]), even linear solutions will be limited on a non-uniform mesh, leading to a loss of accuracy.

Let (1) be applied to linear data with slope s on a mesh with constant stretching ratio a, so that $(x_i - x_{i-1}) = a \cdot (x_{i+1} - x_i)$. We get

$$R = \frac{u_{i+1} - u_i}{u_i - u_{i-1}} = \frac{s \cdot (x_{i+1} - x_i)}{s \cdot (x_i - x_{i-1})} = a. \tag{23}$$

On a uniform grid this ratio is unity, but on a stretched mesh it isn't, so all limiters except Barth-Jespersen will activate and fail to preserve the linear data. For example, if $a = 1.2$, the *min* limiter using (8) gives $\phi = .90$. Luckily the van Leer limiter gives $\phi = .99$, and van Albada gives $\phi = .98$, which are less severe for this modest stretching ratio. For an abrupt mesh refinement in space with $h_{i+1} = 2 \cdot h_i$, $R = \frac{3}{4}$ and the uncorrected limiters are $\phi_{min} = 6/7$, $\phi_{VA} = 0.96$, and $\phi_{VL} = 0.98$.

It is tempting to try modifying the limiter definitions for non-uniform grids by re-writing them using scaled differences, or gradients. For example one could simply re-define $R = \frac{u_{i+1} - u_i}{u_i - u_{i-1}}$ to be

$$R' = \frac{D_+}{D_-}. \tag{24}$$

Substituting into the van Leer limiter (8) gives

$$\phi_{VL}(R') = \frac{D_+ D_-}{D_c^2} \tag{25}$$

on uniform meshes using (22), or simply left as

$$\phi_{VL}(R') = \frac{4(D_+ D_-)}{(D_+ + D_-)^2} \quad \text{or} \quad \phi_{VL}(R') = \frac{D_+ D_-}{D_L^2} \tag{26}$$

for non-uniform meshes, where D_L is the least squares gradient (21).

Since all the gradient approximations will exactly reconstruct a linear function, both forms in (26) are now linearity preserving and won't limit a linear solution on an irregular grid. Unfortunately it turns out that none of these limiters lie entirely inside the TVD region on a general irregular mesh. To be able to find a generalization that stays within the TVD bounds (19), we will examine them more closely using the symmetric form of the previous section.

D. Symmetric Form for Non-Uniform Meshes

The goal of this section is to develop a smooth limiter that remains TVD on irregular grids. Since none of the obvious generalizations above are TVD, a more delicate approach is needed. To understand the difficulties, we look at the TVD limit (19), as well as the generalization of the *min* limiter, which is simple to derive. A comparison of these two, which bracket the smooth limiters in the uniform case, illustrates the difficulty with the non-uniform case. We then propose several limiter formulations that lie in-between the *min* and the TVD limit, all of which simplify to van Leer on uniform grids.

Define the mesh stretching ratios on either side of u_i with

$$a = h_{i-1}/hi \tag{27}$$

$$b = h_{i+1}/hi \tag{28}$$

so that

$$\frac{h_+}{h_-} = \frac{\frac{(h_i+h_{i+1})}{2}}{\frac{(h_i+h_{i-1})}{2}} = \frac{(1+b)}{(1+a)} \tag{29}$$

We will also need expressions for D_+, D_-, and D_L, the least squares gradient (21), since the amount of limiting needed will depend on which gradient is being limited. The expressions will involve the jump $J = u_{i+1} - u_{i-1}$, and the fraction $f = (u_i - u_{i-1})/J$, giving

$$D_- = \frac{u_i - u_{i-1}}{h_-} = \frac{2fJ}{(1+a)h_i} \tag{30}$$

$$D_+ = \frac{u_{i+1} - u_i}{h_+} = \frac{2(1-f)J}{(1+b)h_i} \tag{31}$$

$$D_L = \frac{(1+b)(1-f) + (1+a)f}{(1+b)^2 + (1+a)^2} \; \frac{2J}{h_i}. \tag{32}$$

We can now derive the symmetric form for the TVD limit and the *min* limiter on a non-uniform grid, assuming monotone increasing data. For the TVD limit, (19) requires that the limited gradient be less than a maximum value $\phi_{tvd} D_L$ where

$$\phi_{tvd} D_L = \min\left\{ \frac{2(u_i - u_{i-1})}{h_i}, \frac{2(u_{i+1} - u_i)}{h_i} \right\} = \min\{f, (1-f)\} \, \frac{2J}{h_i}. \tag{33}$$

This has its peak when $f = 1/2$, where it takes the value

$$\phi_{tvd} D_L = \frac{J}{h_i}. \tag{34}$$

For the *min* limiter, which has a straightforward generalization to non-uniform meshes,

$$\phi_{min} D_L = \min\left\{ \frac{(u_i - u_{i-1})}{h_-}, \frac{(u_{i+1} - u_i)}{h_+} \right\} = \min\left\{ \frac{(1-f)}{(1+b)}, \frac{f}{(1+a)} \right\} \frac{2J}{h_i}. \tag{35}$$

This function has a peak when the terms are equal, (e.g. the data is linear and $D_- = D_+$), which occurs when $f = f_p$, where

$$f_p = \frac{(1+a)}{2+a+b}, \tag{36}$$

and the limited slope takes the value

$$\phi_{min} D_L(f_p) = \frac{2}{(2+a+b)} \frac{J}{h_i}. \tag{37}$$

Since a and b are positive, this peak value is less than that of the TVD peak value (34), and is 1/2 of the peak on a uniform grid, as expected. As the mesh ratios a and b approach 0, the *min* peak approaches the TVD peak. For non-unit mesh ratios, note that the peaks are not located at the same place.

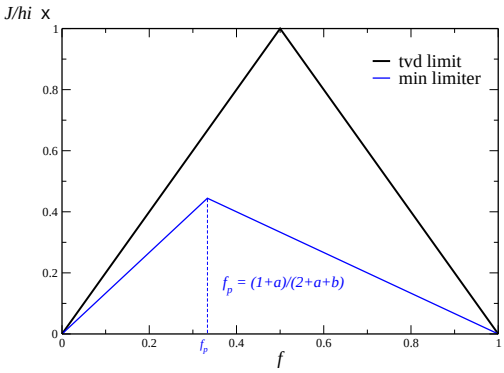


Figure 4. The TVD slope limit and the min limiter are shown here on non-uniform meshes with mesh ratios $a = .5$ and $b = 2$ defined in (27) and (28) corresponding to a mesh stretching of 2. The limiters are plotted as a function of $f = (u_i - u_{i-1})/(u_{i+1} - u_{i-1})$ using the symmetric form.

For both the TVD and *min* limiters, (33) and (35) are piecewise linear functions of f, and go to zero as f approaches 0 and 1. This is shown graphically in figure 4 for a mesh with a constant stretching ratio of 2, corresponding to $a = .5$, $b = 2$.

All of the other limiters lie between the TVD and *min* limiters. A limiter should also go through the *min* limiter peak, since this implies a linear function will not be limited on the irregular grid. (The TVD peak by contrast shows that the gradient can be magnified and still be TVD.)

Figure 4 shows the problem with smooth limiters on non-uniform grids. As the mesh stretching a and b go to 0, the *min* and TVD limiters approach each other, leaving no room to squeeze in a van Leer-like quadratic of any form. Instead, the exponent must be variable and be flexible enough to approach a linear function in the limit. This function should start at 0 when $f = 0$, and its gradient should be between the TVD and *min* function gradient. It should interpolate the *min* peak, at f_p, the condition for preserving a linear function.

If $a = 0$ and b are large, the *min* limiter will lie very close to the TVD limit on the left, but have lots of room on the right. Thus figure 4 shows that we will need two different functions for the generalization, one to the left of the peak, $f \le f_p$, and one to the right, $f \ge f_p$. They should be differentiable, and so the final interpolation condition for our function is that it have a zero derivative at the peak, so both functions can match smoothly there. When the mesh is uniform, we would like the limiter to recover the form of a standard smooth limiter like the van Leer quadratic (8).

Several functions can be derived that have these properties. The one we use in the numerical experiments is

$$\phi_1 D_L = \begin{cases} f\left[1 - \frac{a}{1+a}\left(\frac{f}{f_p}\right)^{\frac{1}{a}}\right] \frac{2J}{h_i} & f \le f_p \\ (1-f)\left[1 - \frac{b}{1+b}\left(\frac{1-f}{f_p}\right)^{\frac{1}{b}}\right] \frac{2J}{h_i} & f \ge f_p \end{cases} \tag{38}$$

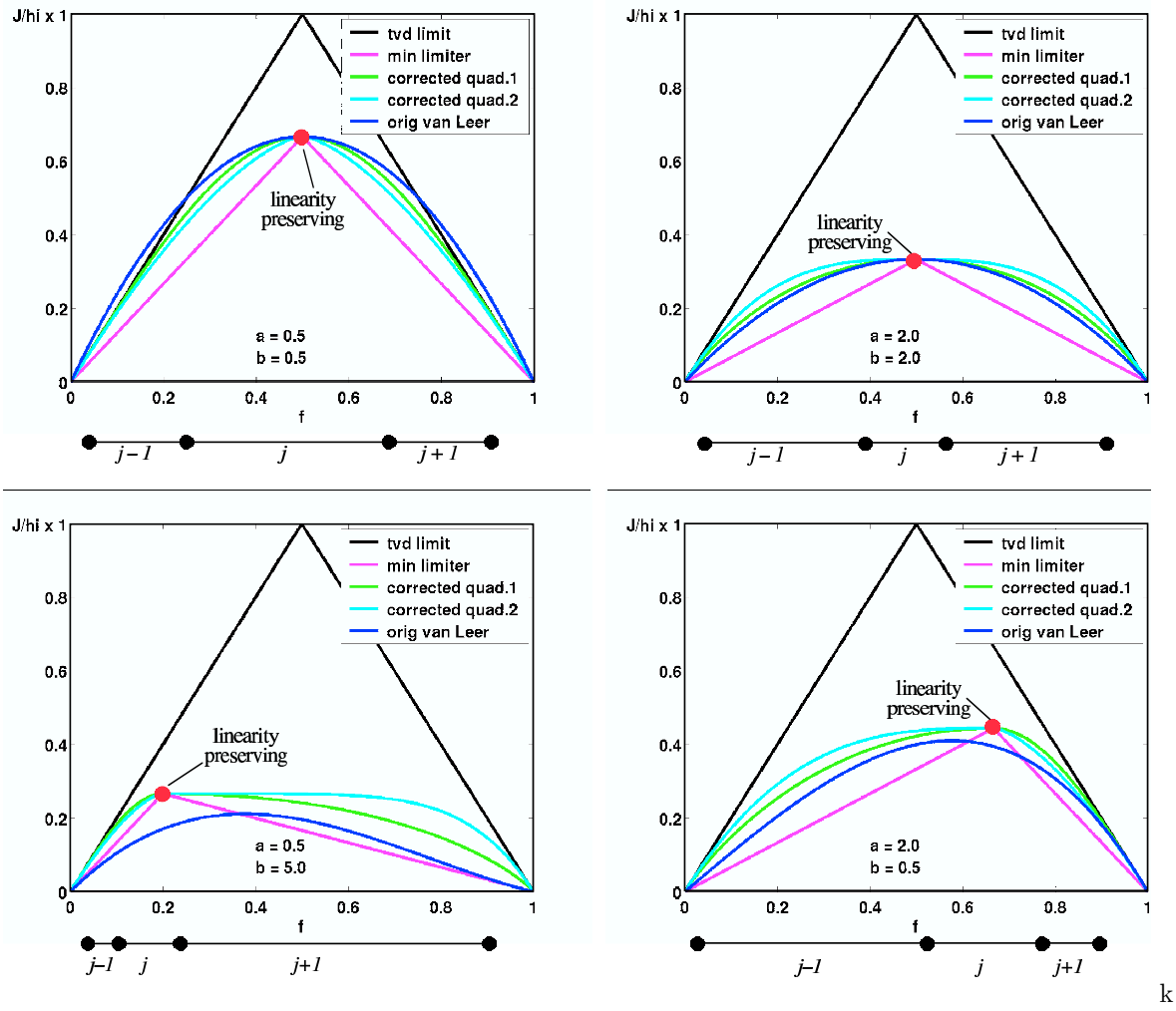


Figure 5. Two monotonic, linearity-preserving generalizations of a quadratic limiter that lie between the TVD and min limiter are shown, corresponding to (38) and (39). Each figure shows the mesh stretching ratios used. The values $a = .5, b = 2$ correspond to a constant stretching ratio of 2. The values $a = 2, b = 2$ correspond to a cell with a coarser cell on each side (analogously $a = b = .5$ corresponds to a finer cell on each side). The last figure shows a more extreme ratio with $b = 5$. Both corrected limiters become van Leer's quadratic limiter on a uniform mesh.

Another possibility is

$$\phi_2 D_L = \begin{cases} \frac{f_p}{1+a}[1-(1-\frac{f}{f_p})^{1+a}]\ \frac{2J}{h_i} & f \le f_p \\ \frac{f_p}{1+b}[1-(1-\frac{1-f}{f_p})^{1+b}]\ \frac{2J}{h_i} & f \ge f_p \end{cases} \tag{39}$$

On a uniform mesh, where $a = b = 1$, and $f_p = 1/2$, these both recover the usual van Leer quadratic form.

In figure 5 we plot these limiters for a variety of mesh ratios a and b. As the mesh stretching varies, the location of the peak moves, so that linear functions will not be limited. With the variable exponent on the modified limiters, they always lie beneath the TVD limit. In the next section we will show computational experiments in one dimension examining their accuracy and monotonicity.

E. One-Dimensional Experiments

We demonstrate graphically the monotonicity of the new limiter (38) and the lack of monotonicity of van Leer without modification. In Appendix B we also include a study of the accuracy of the limiters on non-uniform grids.

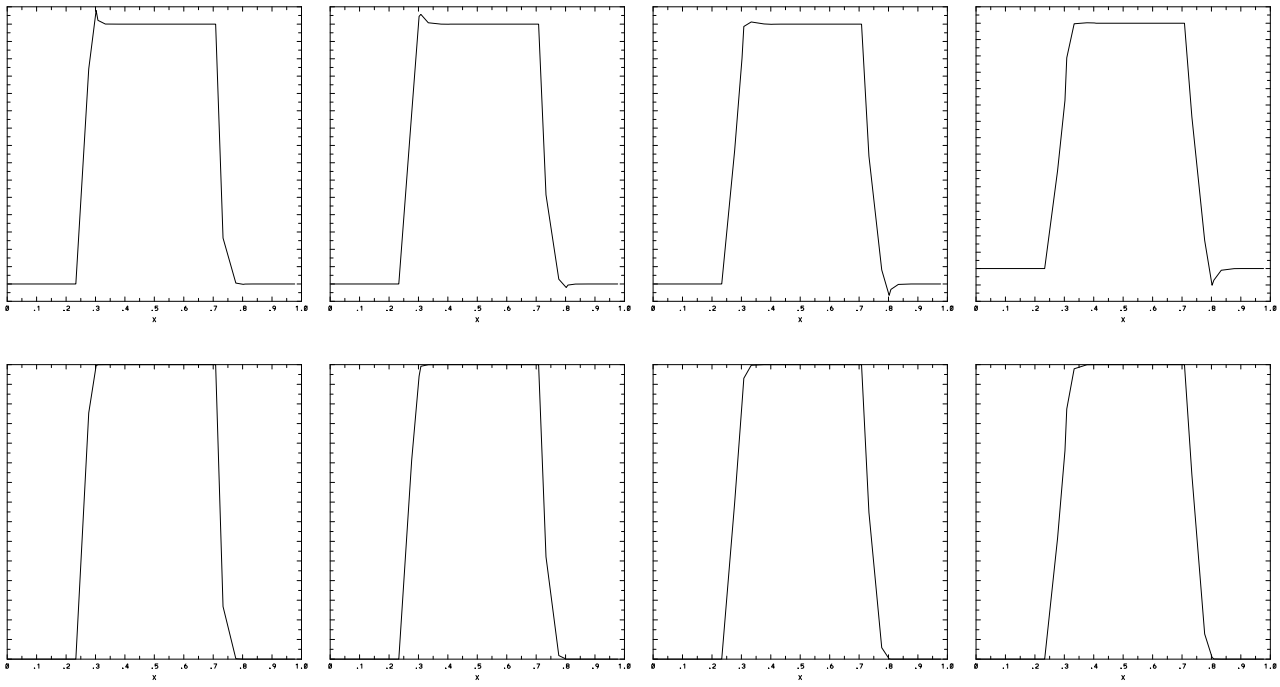


Figure 6. Results of advecting a square wave using the original non-monotone van Leer limiter (top row), and the modified limiter (38) (bottom row), using 40 points in the unit interval. Snapshots are at 2, 4, 6 and 8 steps. Details of the experiment are given in Appendix B.

The irregular mesh consists of repeating blocks of 4 cells with mesh widths h, $2h$, $10h$ and $11h$. The time step is chosen based on the smallest cell size, using a CFL of .9 and a 3-stage TVD Runge-Kutta scheme. The initial conditions are a square wave. The sequence of plots in figure 6 show that the overshoots appear and disappear, depending on whether the next cell is larger or smaller. By contrast, the new limiter preserves monotonicity at all steps.

IV. Multidimensional Limiting

The one-dimensional formulations of the previous section are only a first step. In multiple dimensions, the gradient is now a vector whose components may be limited componentwise in each direction or with a single scalar. In addition, multiple dimensions introduces the possibility of mesh skewing along with mesh stretching, adding complexity to the issues of monotonicity and linearity preservation.

Figure 7 shows simple two-dimensional examples illustrating problems that can arise. As previously pointed out,[16] on the three irregular meshes shown, linear data may be inadvertently limited, sacrificing k-exactness and producing excessive dissipation. In each case, consider linear data such that the centroidal values A and B are equal, (thus the gradient is normal to the line connecting them). The face centroid C, on the face between A and B, will then be an extremum when compared to the cell averages. Since

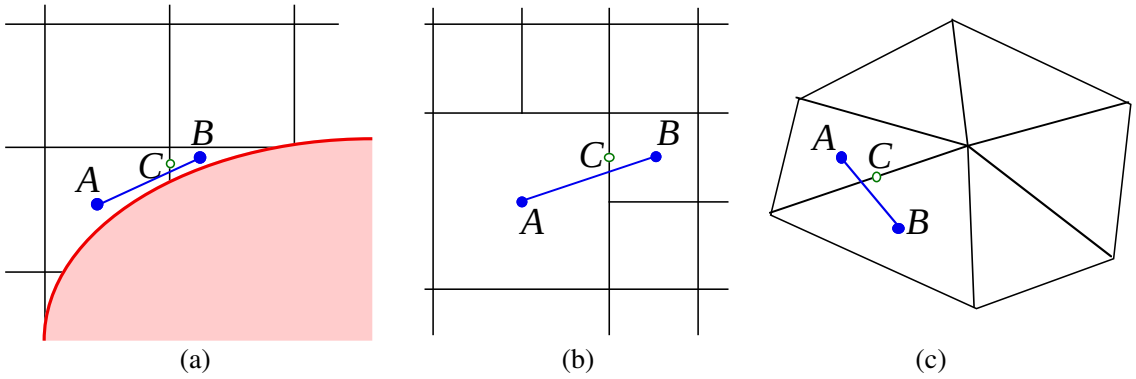


Figure 7. In all three examples, linear data is not preserved by standard limiters. Let the line between cell centers A and B be a level line of the solution. The flux evaluation and solution limiting occur at point C, which is not on the line. Point C will look like an overshoot, and the *exact* gradient which was reconstructed at A and B will be limited.

C is an extremum, the limiter will clip the gradients of both A and B. Without careful attention in the implementation, this will happen regardless of which limiter is used, whenever the reconstructed point is not *exactly* co-linear with A and B. Scalar limiters exacerbate this problem by clipping all components of the gradient vector.

A. Directional Limiting and Gradient Rotation

Most finite volume implementations on unstructured grids use a scalar form of the limiter

$$u^L = u_i + \phi_i \vec{r} \nabla u_i. \tag{40}$$

Here ϕ is a scalar slope limiter in cell i and $\vec{r}$ is the position vector from the cell centroid to the face. In contrast, most structured techniques limit gradients on a direction-by-direction basis. Faces in x limit u_x; faces in y limit u_y, etc. This component-wise limiting can be expressed as a matrix limiter where $\bar{\bar{\phi}}$ is the diagonal matrix $\bar{\bar{\phi}} = \text{Diag}[\phi_x, \phi_y, \phi_z]$, and (40) becomes

$$u^L = u_i + \vec{r}\,\bar{\bar{\phi}}\,\nabla u_i. \tag{41}$$

Scalar limiters, while popular, are far more severe than their vector counterparts, since limiting triggered by *any* face of a polyhedral cell degrades the gradient in *all* directions. Previous results[1] show that even for smooth flows, the scalar implementation is dramatically more dissipative than the directional form, making it imperative to seek vector formulations.

While directional limiting is straightforward on cells with orthogonal faces, it is more subtle on general meshes. The underlying difficulty stems from competing requirements of a face-by-face implementation that still guarantee positivity. Figure 8 illustrates the situation for both cell types. The frames on the left of Figure 8(a) trace the evolution of the gradient on a Cartesian cell, while the frames on the right examine the process for a cell with non-orthogonal faces (in this case simply a triangle).

Following the evolution on the orthogonal cell in 8(a), frame 8(a.1) shows the monotonicity boundary imposed by face *1*. The directional implementation removes the component of ∇u normal to this face (shown in red) resulting in the modified gradient $\nabla u'$. In frame 8(a.2), the gradient is further reduced by removal of the component of $\nabla u'$ that extends beyond the monotonicity boundary established by face 2. Notice that since faces 1 and 2 are orthogonal, the limiting in frame 8(a.2) retards the gradient while still respecting the limit boundary from face 1. The limited gradient, $\bar{\bar{\phi}}\nabla u$ resulting from this face-based implementation

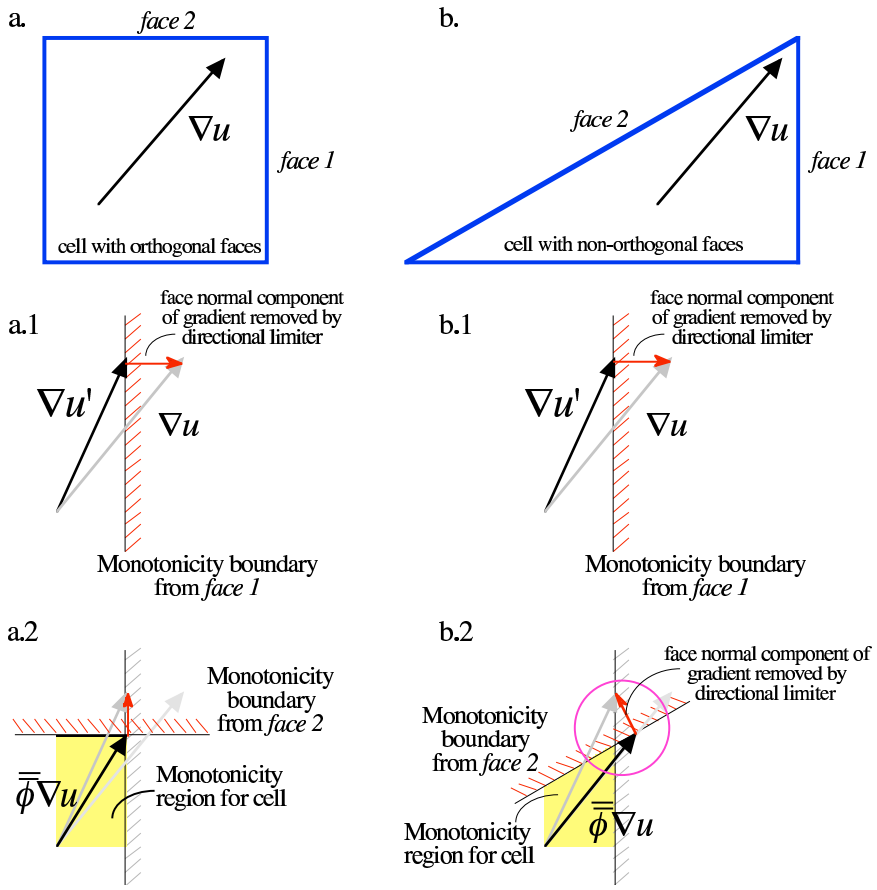


Figure 8. Illustration of the directional limiting process on a cell with orthogonal (a) and non-orthogonal (b) faces. The "monotonicity region" (shaded in yellow) is more properly the "region of allowable gradient" according to the chosen limiter. The final limited gradient must fall within in the yellow region.

sits on the perimeter of the monotonicity region (shaded yellow) in Figure 8(a.2). Since the boundaries in this figure are imposed by one's chosen limiter they are not *strictly* monotonicity boundaries. Similarly, the "monotonicity region" is more properly the "region of allowable gradient" according to the chosen limiter.

By contrast with the Cartesian example, examine the result of this process on the general polyhedral cell on the right of Figure 8. In frame 8(b.1) face 1 limits ∇u to $\nabla u'$ by again removing the component normal to the monotonicity boundary. In frame 8(b.2) we impose the corresponding limit from face 2. This time, however, removal of the face normal component of the gradient produces a limited gradient $\bar{\bar{\phi}}\nabla u$ which violates the earlier boundary established by face 1, and the final limited gradient unfortunately lies *outside* the monotonicity region for the cell. Inspection of the geometry Figure 8 makes it clear that this will be a problem anytime adjacent limit boundaries form an acute angle with each other.

The reduction in dissipation and improved reconstruction properties offered by directional limiting[1] gives strong motivation to successfully implement them on general polygonal meshes. However, as the sketches in Figure 8 demonstrate, this implementation is delicate. In essence, each new face must respect the boundary of the monotonicity region established by all other faces. This can be posed as a clipping operation, an implicit system of equations, or a linear programming problem. For each cell gradient, we seek the point on the boundary of the monotonicity region which preserves as much of the unlimited gradient as possible. Additional sweeps over the cell faces would be required to determine the limiter value which simultaneously

respects the monotonicity boundaries imposed by all faces of general polyhedral cells. However on cells with perpendicular faces, this system decouples and orthogonality ensures that limiting of one face will not exceed a constraint boundary that has already been imposed.

B. Multidimensional Limiting as an *LP* Problem

The directional limiting problem outlined in the preceding section has an interesting formalization. The sketches in Figure 8 highlight several important observations. The region of allowable gradient is bounded by either the limit boundaries imposed by the cell faces, or by a sign constraint (which simply states that the limiter does not change the sign of the gradient in any direction). Note also that any limited gradient which is inside the monotonicity region will produce a monotone solution, however, we seek one which is "closest to" the unlimited gradient. In other words, we seek the point on the boundary of the closed polygon (polyhedron in higher dimensions) which minimizes in some measure the reduction of the original gradient vector.

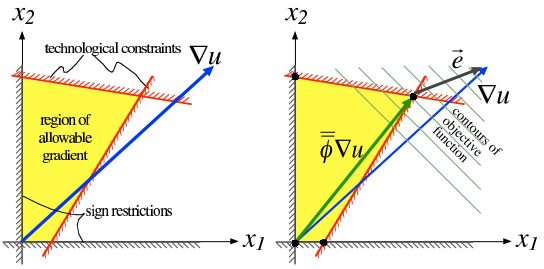


Figure 9. Directional limiting recast in *LP* framework. The limiter on each cell face imposes technological constraints as shown on the left. On the right, the final limited gradient is the optimal solution of the *LP* problem seeking minimize reduction of the gradient vector over the monotonicity region.

Figure 9 sketches this reinterpretation. Examining this figure, it is clear that the multidimensional limiting problem is exactly the description of a problem in linear programming (LP). This observation opens up a wealth of literature with both exact and inexact possible solution techniques.

The *technological constraints* of the LP system are the limit boundaries imposed by any chosen limiter on each cell face. On the i^{th} face, in three spatial dimensions, this limit boundary can be expressed as

$$a_{ix}u_x + a_{iy}u_y + a_{iz}u_z \begin{pmatrix} \geq \\ \leq \end{pmatrix} b_i, \quad i = 1, ..., m. \tag{42}$$

This is a linear equation in the components of the gradient vector, where m is the number of cell faces. In addition, we have the *sign constraints* saying that the limiter $\bar{\bar{\phi}}$ can't change the sign of the gradient,

$$\phi_x \geq 0 \quad \phi_y \geq 0 \quad \phi_z \geq 0 \tag{43}$$

The objective function of the *LP* is to minimize the reduction in the gradient. In Figure 9, $\vec{e}$ is the error vector - the amount of the gradient that must be removed so that $\nabla u - \vec{e} = \bar{\bar{\phi}}\nabla u$. We seek to preserve as much of the gradient as possible, thus minimizing $\vec{e}$.

$$\text{minimize } \|e\|_1 := |e_x| + |e_y| + |e_z| = u_x(1-\phi_x) + u_y(1-\phi_y) + u_z(1-\phi_z) \tag{44}$$

By using the $L1$ norm to measure the magnitude of the reduction in the gradient the problem stays linear.

Since (42) - (44) form a linear system of equations, the fundamental theorem of *LP* tells us that the optimal value of the objective function must be at a corner of the region bounded by the constraints (42) and (43). The sketch in the right frame of Figure 9 makes this clear as well with isoclines of the objective function overlaying the region of allowable gradient, with the optimal solution at the upper right corner of the monotonicity region.

This formulation gives important insight into multidimensional limiting. In the special case of Cartesian constraint polygons, the constraint equations (42) decouple and thus the constraints from each face can be

applied independently and in any order. This was implied by our earlier discussion of Figure 8 but is now obvious. Scalar limiting is also an interesting special case. In one dimension it is the only choice, and yields optimum solutions. In higher dimensions, however, a scalar limiter holds the direction of the gradient fixed, thus the ratios of the coefficients in equation (42) are predetermined and only one of the constraints will be binding. This is attractive since, again, the constraints can be applied one at a time. However, since the vector is only free to vary along its length, the resulting limited gradient will be potentially far from the optimum allowed by the *LP* problem, producing a monotone but excessively dissipative solution. For non-Cartesian monotonicity polygons optimal solutions clearly require a coupled technique to determine which of the feasible solutions to the *LP* problem is optimal.

C. An *LP* Formulation for Mesh Interfaces

As noted in the previous section, if the constraint polygon is Cartesian, the constraint equations (42) uncouple, and the limit boundaries may be imposed one-by-one as in the directional process sketched in Figure 8a (left side) and still yield the solution to the *LP* problem. This seemingly narrow subclass of problems is of particular interest on Cartesian and structured meshes. Not only can it be used in all the regular volume hexahedra away from refinement interfaces, but with only a slight variation, it can be applied at Cartesian mesh interfaces as well.

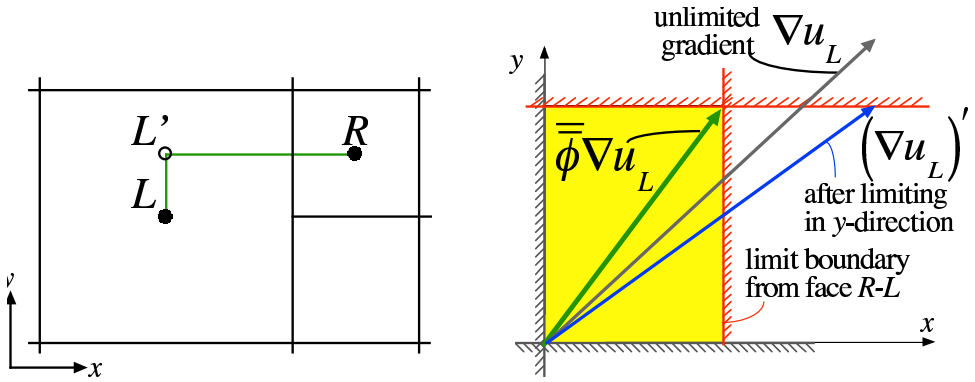


Figure 10. On the left is a typical mesh refinement interface in the x direction. On the right is its associated monotonicity polygon.

The LP formulation can be used to analyze a simple approach that maintains linearity at mesh interfaces. The left side of Figure 10 shows Cartesian cells on either side of a mesh refinement interface in 2D. The constraint polygon for cell L is sketched in the frame on the right. The geometric picture is shown in Figure 10. In the figure, the original gradient ∇u_L has been limited in the y direction by the north and south neighbors, (i.e. $(\nabla u_L)'$ satisfies the technological constraints at the top of the monotonicity region). The remaining question is how to limit the x gradient, since the large and small cell centroids are not coordinate aligned in this direction. We can accomplish this by using the already limited y gradient to safely *recenter* the data in cell L to L',

$$u_{L'} = u_L + (\nabla u_L)' \cdot \overrightarrow{d_{L'L}}. \tag{45}$$

After recentering, L' may be thought of as a *virtual cell center* serving as a left state in the limiter evaluation on the face between L and R. One can then apply any of the compact two-point limiters (*min* or Barth-Jespersen) since they require no assumptions about what is happening to the left of cell L. For example, applying the former on this face simply compares the x component of ∇u_L with the one-sided slope between R and L',

$$\phi_x u_x = \min\left(u_x, \frac{u_R - u_{L'}}{\|d_{RL'}\|}\right). \tag{46}$$

This recentering allows us to implement a directional, vector limiter instead of the more common scalar form. Alternatively one can define a cell to the left of cell L and use any of the stretched mesh formulas from eqs.(38) or (39) to guarantee a positive state for the Riemann problem at the face.

This recentering procedure has a direct interpretation using the monotonicity polygon. In Fig. 10, once the gradient satisfies the y constraints, it is retarded in the x direction only until the monotonicity boundary coming from the interface is satisfied. To do this, we slide the tip of the gradient vector along the upper boundary of the constraint polygon until it is at the northwest corner of the monotonicity region, leaving the y gradient alone. Algebraically, this is equivalent to comparing the directional derivative $(\nabla u_L)' \cdot \overrightarrow{d_{LR}}$ in the direction between the centroids with the one-sided difference $u_R - u_L$. If the former exceeds the latter, the necessary change is attributed completely to the x limiter. Algebraically, this gives

$$\bar{\bar{\phi}} \nabla u_L \cdot \bar{n} = \frac{u_R - u_L}{d_{RL}} \tag{47}$$

$$\phi_x u_x \frac{d_{RL\prime}}{d_{RL}} + u_y \frac{d_{L\prime L}}{d_{RL}} = \frac{u_R - u_L}{d_{RL}} \tag{48}$$

$$\phi_x u_x = \frac{(u_R - u_L - d_{L\prime L} u_y)}{d_{RL\prime}}, \tag{49}$$

and (49) is equivalent to the recentering equation (46) above.

D. An *LP* Interpretation of Scalar Limiting

As noted in section IV.B, scalar limiting can be viewed as a second special case of the *LP* formulation of multidimensional limiting. In the scalar case, the direction of the gradient vector is held fixed, and the only feasible solution is the intersection of the gradient vector with the constraint polygon. Since the constraint polygons are convex, this implies only one intersection, and thus only one of the constraint equations (42) is binding. For edge or face-based data structures this is extremely attractive. Ultimately only one face sets the limiter's value, and this final value is unique regardless of the order in which the constraints are applied.

For general unstructured meshes, scalar limiting obviously adds excessive dissipation since it reduces the gradient vector more than is optimal. However the simplicity of implementation makes it worth another look. In a Cartesian mesh with embedded boundaries there are only $O(N^2)$ cut cells in a volume mesh with $O(N^3)$ total cells. Motivated by these kinds of counting arguments we include results in the next section using a linearly-exact form of scalar limiting for this lower dimensional subset of mesh cells. Referring to the sketch in figure 7a, a linearly-exact scalar *min* limiter may be implemented by comparing the projection of the gradients in cells A and B with the directional derivative $\frac{u_B - u_A}{||\vec{d}_{AB}||}$. If the projection of either cell's gradient exceeds this directional derivative, then all components of that cells' gradient are reduced to remove the excess. Note that this is not the standard implementation of scalar limiting. Normally the solution is reconstructed to the face centroid, which is not generally on the line connecting the cell centroids.

V. Computational Experiments

Numerical experiments designed to support the discussion in the previous sections are presented. The computations use a parallel, multi-level Cartesian solver for the Euler flow equations.[17] The Onera M6 wing computed at Mach 0.5, and $\alpha = 3.06^\circ$ provides an attractive test case. Since strong shocks do not form at these conditions, limiters are not required to maintain positivity, allowing direct comparison of results with and without limiters. Further, the inviscid flow is irrotational and hence the theoretical value for the drag is zero. In numerical simulations, the sharp trailing edge of the wing produces a small but finite drag value.

Figure 11 shows the Cartesian computational mesh and sample pressure contours for the Onera M6 calculated without limiters ($\phi = 1$ everywhere). The mesh contains 301K cells, with roughly 21K cut cells adjacent to the embedded boundary, and 88K interface faces with 2:1 mesh refinement, for roughly 110K interface cells.

The use of a Cartesian mesh allows us to isolate the effects of different limiter formulations. In the first experiment, the standard limiter applied in the regular hexahedra is varied, while holding the limiter in the interface and cut cells fixed. In the interface cells the recentering *min* formulation (45) is applied, and in the cut cells the limiter is forced to $\phi = 0$ to isolate the effects of the field limiters on convergence. Table 1 presents the computed drag using each of the limiters presented in (8). The computed results show an increase in drag corresponding to the increasing dissipation associated with the limiter. The unlimited calculation provides the minimum drag possible for this combination of solver and mesh. The convergence properties (L1 norm of density residual) of the standard limiters are presented in figure 12. Five levels of a full multigrid scheme are used. Since the coarse levels are first order, the startup behavior for the first four levels are identical. All simulations show good convergence, with the exception of the calculation using the Barth-Jespersen limiter, which stalls after converging roughly three orders of magnitude. This behavior with BJ is well documented.[1,2]

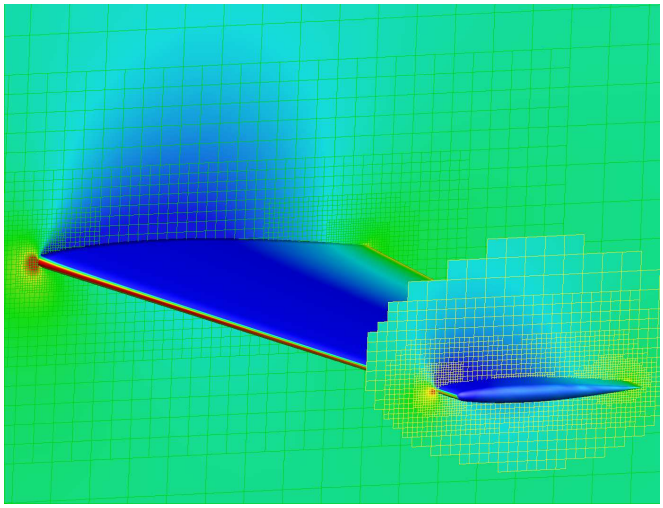


Figure 11. Onera M6 mesh and sample pressure contours calculated without limiters ($M_\infty = 0.5$, $\alpha = 3.06°$).

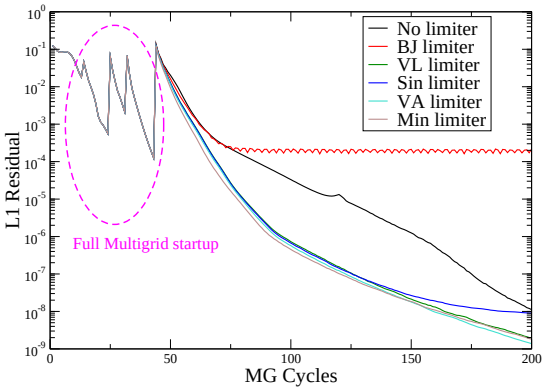


Figure 12. Convergence in the L1 norm of the density residual for subsonic OneraM6 wing example ($M_\infty = 0.5$, $\alpha = 3.06°$).

Table 1. Computed drag for different limiter formulations in the regular hexahedra, in order of increasing dissipation.

Limiter Type	Drag Coefficient $\times 10^3$
No limiter	3.24
BJ limiter	4.56
VL limiter	5.41
Sin limiter	5.59
VA limiter	6.15
Min limiter	8.15

A similar comparison is presented to illustrate the behavior of the recentering procedure presented in section IV.C. The subsonic Onera M6 wing is computed using the scalar BJ limiter, along with the recentering procedure implemented using the BJ and *min* limiters. The scalar BJ reconstructs to the face centroid, and reduces the value if it exceeds the neighboring centroid value. The regular hexahedra use the van Leer limiter, while the cut cells are again forced to $\phi = 0$. Recall that BJ and *min* are the only compact two-point limiters in (8). Implementation of the recentering approach for the stretched mesh form of the limiters (38) is left for future work. Table 2 presents the computed drag varying only the interface limiters. As before, the increase in drag correlates with an increase in numerical dissipation. The recentering procedure reduces the dissipation over the scalar procedure. Simply changing the implementation from scalar to recentered

directional in the interface cells reduces the drag by nearly 14 counts. The recentering procedure improves the convergence history as well, as shown in figure 13. The convergence of the scalar implementation stalls after 3 orders, while the recentered implementation has good convergence properties.

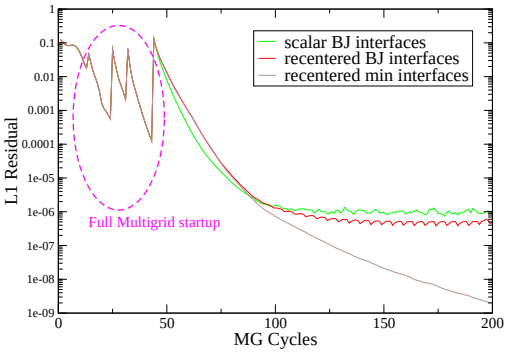


Figure 13. Convergence in the L1 norm of the density residual subsonic OneraM6 wing example ($M_\infty = 0.5$, $\alpha = 3.06^\circ$).

Table 2. Computed drag for different limiters in the 2:1 interface cells.

Limiter Type	Drag Coefficient $\times 10^3$
No limiter	3.24
Recentered BJ	5.24
Recentered Min	5.41
Scalar BJ	6.60

The recentering procedure is effective at interfaces since the limiting of the transverse directions can be done first. For general polyhedra this approach can not be applied directly. Table 3 presents instead the computed drag using the scalar *min* limiter in the cut cells, compared against unlimited and first-order cut cell results. Using scalar *min* in the cut-cell polyhedra stalls convergence in the density residual, similar to the results using scalar BJ at the interfaces. The discussion above demonstrates that the van Leer limiter in the regular hexahedra, and the recentered *min* limiter in the interface cells both converge. Thus, the limiter chatter is caused by the scalar *min* implementation in the cut cells, even though this limiter is monotone and linearity-preserving. Note that the scalar limiter does not anchor the limited gradient vector at a corner of the monotonicity polygon, unlike the recentered *min* procedure used in the interface cells, which does. One hypothesis is that the gradient vector is rotating excessively from iteration to iteration, generating low-level chatter. An optimal solution to the LP problem in these cells would tie the gradient to a corner of the positivity polygon, and possibly reduce the noise. This hypothesis is currently being tested by solving the full *LP* problem in the cut cells.

Table 3. Computed drag using different limiters in the cut cells.

Limiter Type	Drag Coefficient $\times 10^3$
No limiter	3.24
Scalar Min	4.72
1^{st} order cut cells	5.41

In the final example, we demonstrate the performance and robustness of the limiters and flow solver for a practical application. We use the van Leer limiter in the regular hexahedra, the recentered *min* at the interfaces, and the scalar *min* in the cut cells. These were applied in the same fashion as in the Onera example without any tweaking. We compute the flow around the space shuttle orbiter at a free stream Mach number of 18.5, and angle of attack 15 degrees. The computational mesh contains 1.8M cells. This example used a 3 level multigrid W cycle. Gradients were evaluated at every stage of a five stage Runge-Kutta scheme. Figure 14 shows flow features that include strong shocks, regions of separated flow, and rapid expansions. The density and pressure at the back of the shuttle (in the engine bells and OMS pods) gets as low as 10^{-7} times the non-dimensional free stream quantity. The pressure on the orbiter's nose is nearly 450 times that in the free stream, a range of nearly eight orders of magnitude. Figure 15 shows the convergence history of the residual and the forces for this case. As before, convergence stalls after 3 orders of magnitude.

VI. Conclusions

We have reviewed and analyzed common forms of slope limiters, highlighting their shortcomings for practical application. The shortcomings are the failure of traditional implementations to preserve linear solutions in the presence of mesh stretching or other irregularities, and a lack of a clear extension to multiple dimensions. The main contributions of this paper are the analysis of limiters in one dimension on non-uniform grids, and an LP formulation with which to analyze the general multidimensional case. Future work will use this formulation to test directional limiters that maintain monotonicity and do not need to be followed by a positivity test. The performance of all the limiters were demonstrated in three dimensional applications using a Cartesian embedded-boundary method.

VII. Acknowledgments

We thank Jonathan Goodman for insightful discussions and a careful reading of the manuscript. Dan Goodman is thanked for suggesting the functional form (39). Thanks also to Tom Pulliam for testing some of the 1d stretched mesh limiters. Marsha Berger was supported by AFOSR grant F19620-00-0099 and DOE grants DE-FG02-00ER25053 and DE-FC02-01ER25472. Scott Murman was supported by NASA Ames Research Center (contract NAS2-00062) during this work.

References

[1]Aftosmis, M., Gaitonde, D., and Tavares, T. S., "On the accuracy, stability and monotonicity of various reconstruction algorithms for unstructured meshes," *AIAA-94-0415*, January 1994.

[2]Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *J. Comp. Phys.*, Vol. 118, 1995, pp. 120–130.

[3]Harten, A., "High resolution schemes for hyperbolic conservation laws," *J. Comp. Phys.*, Vol. 49, 1983, pp. 357–393.

[4]Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 2: Artificial Diffusion and Discrete Shock Structure," Submitted to *Intl. J. Comp. Fluid Dynamics*, 1994.

[5]Rider, W. J. and Kothe, D. B., "Constrained Minimization for Monotonic Reconstruction," *AIAA-97–2036*, 1997.

[6]Hubbard, M., "Multidimensional Slope Limiters for MUSCL-Type Finite Volume Schemes on Unstructured Grids," *J. Comp. Phys.*, Vol. 155, 1999, pp. 54–74.

[7]Liu, X., Osher, S., and Chan, T., "Weighted essentially non-oscillatory schemes," *J. Comp. Phys.*, Vol. 115, 1994, pp. 200–212.

[8]Aftosmis, M., Melton, J., and Berger, M., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, 1998.

[9]Sweby, P., "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws," *SIAM J. Num. Anal.*, Vol. 21, 1984.

[10]Zalesak, S. T., "Fully multidimensional flux-corrected transport algorithms for fluids," *J. Comp. Phys.*, Vol. 31, 1979, pp. 335–362.

[11]Spekreijse, S., "Multigrid Discretization of Monotone Second-Order Discretizations of Hyperbolic Conservation Laws," *Math. Comp.*, Vol. 49, 1987.

[12]Goodman, J. and LeVeque, R., "A Geometric Approach to High Resolution TVD Schemes," *SIAM J. Num. Anal.*, Vol. 25, 1988.

[13]van Leer, B., "Toward the ultimate conservative difference scheme. V. A second order sequel to Godunov method," *J. Comp. Phys.*, Vol. 32, 1979, pp. 101–136.

[14]Barth, T., "Numerical Methods and Error Estimation for Conservation Laws on Structured and Unstructured Meshes," *Von Karman Institute Computational Fluid Dynamics Lectures Notes*, 2003.

[15]Kroner, D., Noelle, S., and Rokyta, M., "Convergence of higher order upwind finite volume schemes on unstructure grids for scalar conservation laws in several space dimensions," *Numerische Mathematik*, Vol. 71, 1995, pp. 527–560.

[16]Barth, T. and Jespersen, D., "The design and application of upwind schemes on unstructured meshes," *AIAA-89-0366*, 1989.

[17]Aftosmis, M., Berger, M., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," *AIAA-2000-0808*, January 2000.

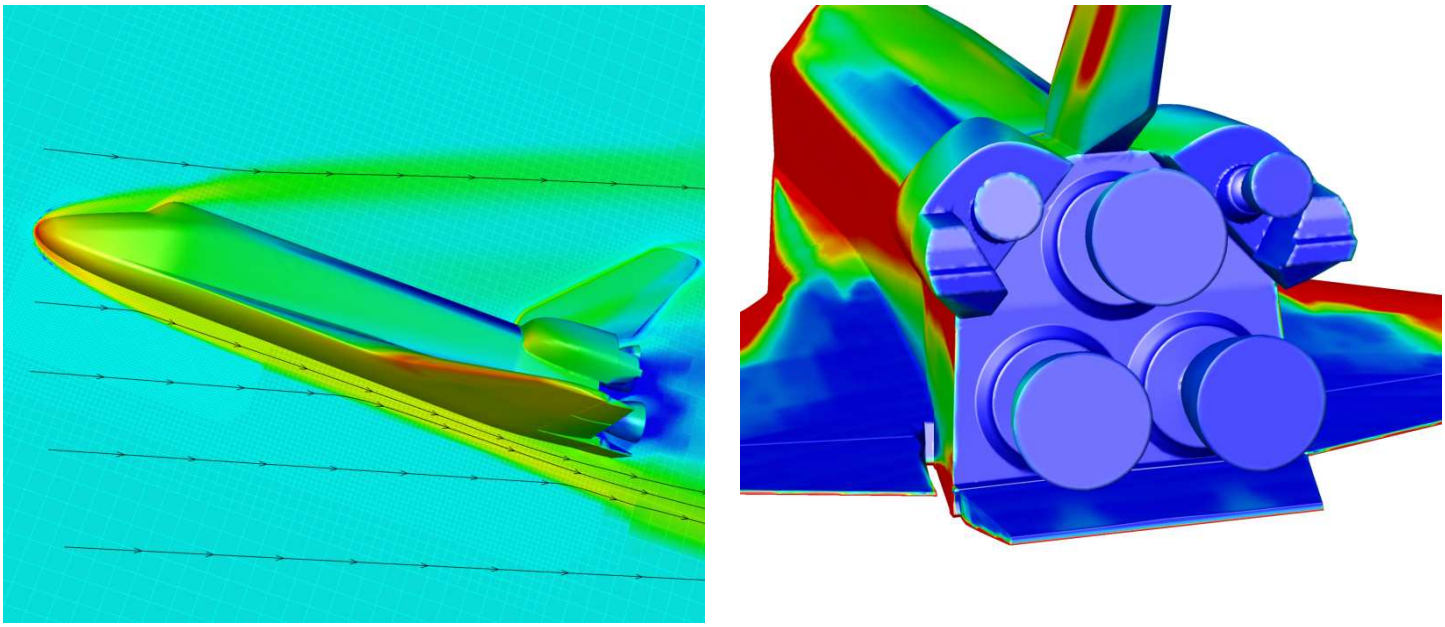


Figure 14. Orbiter calculation using 1.8M cell mesh, Mach 18.5 at 15 degrees angle of attack. The contours show the log of pressure. The left figure also shows some streamlines of the flow in black. Both the complex geometry and complicated physics of the flow field are apparent.

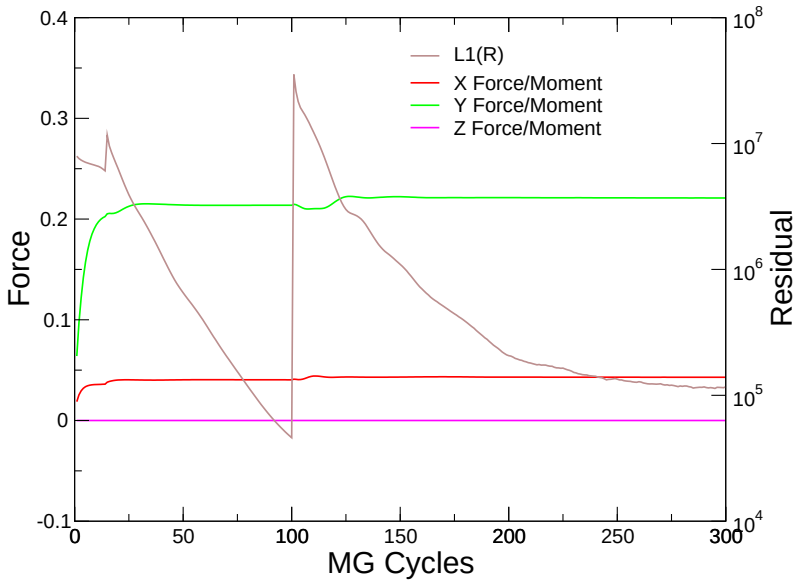


Figure 15. Convergence rate of L1 residual for orbiter calculation. Also shown are the computed forces at each iteration.

Appendix A: Face-Based Limiting

In this appendix we show why face-based limiters is ineffective. We also show that face-based limiting is equivalent to a common technique of maintaining positivity "on-the-fly", which helps explain why this practice does not deliver as much robustness as one would hope.

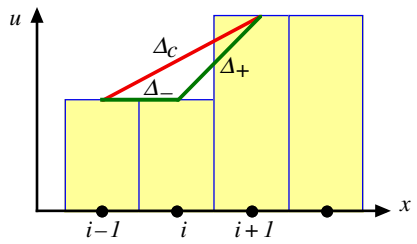


Figure 16. Initial piecewise constant solution for counter-example showing that face-based limiting (or thresholding of reconstructed data) does not preserve monotonicity. All gradients are zero except at cell i.

Face-based limiting was proposed over a decade ago in hopes of reducing dissipation, but does not work well in practice.[1] This approach applies a limiter face-by-face on each cell. Each face limits the reconstructed value as suggested by the limiter, but without actually modifying the gradient of the cell. At another face the original gradient may still be used. Many solvers have adopted a variant of the face-based approach in search of greater robustness. It is common practice to set a floor on the reconstructed values of density, pressure or internal energy to avoid taking the square root of a negative number in the Riemann solver. This situation can arise for example when the limiting is driven by one set of data, and the reconstruction is evaluated at another set of values. When the non-positive state is discovered, it is too late to modify the gradient. These cases are typically treated by thresholding the reconstruction to prevent the non-physical state. In essence, this is face-based limiting: the reconstruction is modified, and the other faces sharing the cell are unaware of the modification.

Let u be a step function as shown in Figure 16 on a uniform grid. For linear advection, $u_t + u_x = 0$, the update at cell i using a second-order upwind difference in space (so the upwind state is on the left) and face-based limiting gives

$$\begin{aligned} u_i^{n+1} &= u_i^n - \frac{\Delta t}{h}(u^L_{i+1/2} - u^L_{i-1/2}) \\ &= u_i^n - \frac{\Delta t}{h}(u^L_{i+1/2} - u^n_{i-1}) \\ &= u_i^n - \lambda((u_i^n + h/2\,\phi_R\, D_{ci}) - u^n_{i-1}), \end{aligned} \tag{50}$$

where ϕ_R is the value of the limiter on the right face of cell i, and D_{ci} is the central difference in cell i. Note that the forward difference $D_+u_i = \frac{u_{i+1}-u_i}{h} > D_{ci}$, so the reconstruction at the right edge will not exceed the value u_{i+1} and will not need to be limited using BJ. Thus $\phi_R = 1$ so

$$\begin{aligned} u_i^{n+1} &= u_i^n - \lambda(u_i + h/2\frac{(u_{i+1} - u_{i-1})}{2h} - u^n_{i-1}) \\ &= u_i^n - \lambda(u_i - u_{i+1}/4 + u_{i-1}/4 - u_{i-1}) \\ &= u_i^n - \lambda(1/4\,u_{i+1} + u_i - 5/4\,u_{i-1}) \\ &= u_i^n - \lambda(u_{i+1}/4 - u_{i-1}/4) \qquad \text{since} \quad u_{i-1} = u_i \\ &< u_i^n \qquad \text{since} \quad u_i < u_{i+1} \end{aligned} \tag{51}$$

In other words, there will be an undershoot at u_i at the next time step. As this example shows, the face with the largest change in the solution is responsible for limiting sufficiently for the other face. While it occurs naturally for symmetric limiters satisfying (7), this symmetry property is violated in the face-based implementation.

Appendix B: Accuracy Study on One-Dimensional Non-Uniform Grids

We demonstrate the accuracy of the one-dimensional limiters (38) for non-uniform grids by advecting a Gaussian periodically on the unit interval. A three stage TVD Runge-Kutta scheme is used to advance the solution until time $t = 1$. The gradient is approximated using the least squares approach (21).

Several types of irregular meshes are employed. A randomly perturbed irregular mesh is generated by choosing N uniform random variables between 0 and 1 for the mesh widths, then scaling them all so that the length of the interval is 1. Also, mesh widths less than 10% of the corresponding uniform mesh width are thrown away, so that a reasonable time step with a CFL number of .9 can be maintained. A block cyclic mesh (Block 1 in the table) is generated using repeating blocks of mesh widths h_1, h_2, h_3, h_4. In the first case, the mesh widths are equal to $dx, 2dx, 3dx, 4dx$ for an appropriately chosen dx that discretizes the unit interval. The more extreme cyclic mesh (Block 2) uses mesh widths equal to $dx, 2dx, 10dx, 11dx$, where this time $dx = 1/(24 \cdot \#pts)$. This is the mesh used in the calculations presented in figure 6.

Table 4. Relative error in the L1 norm for various limiters applied to scalar advection on a non-uniform mesh. The errors increase with increasing mesh irregularity as well as dissipativity. The corrected van Leer limiter is more accurate than the original formulation in all cases.

Limiter Type	Uniform	Random	Block 1	Block 2
No limiter (20 pts)	12.19 %	13.56 %	13.27 %	19.47%
No limiter (40 pts)	2.96 %	3.51 %	3.07 %	4.48 %
No limiter (80 pts)	.69 %	.81 %	.73 %	.68 %
No limiter (160 pts)	.17 %	.20 %	.18 %	.12 %
BJ limiter (20 pts)	11.49 %	14.12 %	14.38 %	23.94 %
BJ limiter (40 pts)	3.00 %	4.40 %	3.96 %	7.72 %
BJ limiter (80 pts)	.86 %	1.21 %	1.14 %	1.85 %
BJ limiter (160 pts)	.24 %	.32 %	.31 %	.48 %
VL limiter (20 pts)	13.27 %	16.21 %	17.22 %	27.95 %
VL limiter (40 pts)	3.83 %	5.35 %	5.76 %	11.24 %
VL limiter (80 pts)	1.11 %	1.64 %	1.86 %	4.60 %
VL limiter (160 pts)	.31 %	.52 %	.69 %	2.10 %
Corrected quad. limiter (20 pts)	13.27 %	15.97 %	16.20 %	24.35 %
Corrected quad. limiter (40 pts)	3.83 %	5.30 %	4.98 %	8.33 %
Corrected quad. limiter (80 pts)	1.11 %	1.63 %	1.57 %	2.37 %
Corrected quad. limiter (160 pts)	.31 %	.44 %	.42 %	.64 %
Min limiter (20 pts)	24.68 %	25.74 %	25.20 %	34.04 %
Min limiter (40 pts)	9.43 %	9.58 %	9.39 %	13.47 %
Min limiter (80 pts)	3.42 %	3.12 %	3.06 %	4.23 %
Min limiter (160 pts)	1.18 %	1.02 %	1.01 %	1.51 %

The Gaussian $y = e^{-25(x-.5)^2}$ is advected for one period. Table 4 compares the error and convergence rates for three limiters, the original van Leer, our revised formulation using (38), and the *min* limiter, as well as a computation that does not use limiters. For comparison, we also include the error of a uniform grid computation with the same number of points. As expected for this smooth solution, the unlimited solution is the most accurate, and has the best convergence rate. On slightly perturbed meshes the new formulation is slightly better. On extreme meshes, it is much better. Thus, there is no penalty for using the new formulation, in some cases it is much better, and the new formulation is preserves monotonicity. The *min* limiter is included in this tableto bracket the error, since we expect that the more dissipative the limiter, the greater the error. The BJ limiter, since it limits the least, is more accurate than van Leer.