INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:545–569
Published online 10 February 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1183

Adaptive mesh refinement for high-resolution finite element schemes

M. Möller*,† and D. Kuzmin‡

Institute of Applied Mathematics (LS III), University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany

SUMMARY

New *a posteriori* error indicators based on edgewise slope-limiting are presented. The L_2-norm is employed to measure the error of the solution gradient in both global and element sense. A second-order Newton–Cotes formula is utilized in order to decompose the local gradient error from a $\mathscr{P}_1$ finite element solution into a sum of edge contributions. The slope values at edge midpoints are interpolated from the two adjacent vertices. Traditional techniques to recover (superconvergent) nodal gradient values from consistent finite element slopes are reviewed. The deficiencies of standard smoothing procedures—L_2-projection and the Zienkiewicz–Zhu patch recovery—as applied to nonsmooth solutions are illustrated for simple academic configurations. The recovered gradient values are corrected by applying a slope limiter edge-by-edge so as to satisfy geometric constraints. The direct computation of slopes at edge midpoints by means of limited averaging of adjacent gradient values is proposed as an inexpensive alternative. Numerical tests for various solution profiles in one and two space dimensions are presented to demonstrate the potential of this postprocessing procedure as an error indicator. Finally, it is used to perform adaptive mesh refinement for compressible inviscid flow simulations. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: error estimation; gradient recovery; adaptive mesh refinement; convection-dominated problems; high-resolution schemes

1. INTRODUCTION

Progress in computer performance and the improvement of numerical methods for CFD have enabled analysts to simulate more and more challenging problems for which no or at least little *a priori* knowledge of the solution structure is available. This increase of complexity has made it rather difficult to guarantee the reliability of the numerical solution. The recent trend for

*Correspondence to: M. Möller, Institute of Applied Mathematics (LS III), University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany.
†E-mail: matthias.moeller@math.uni-dortmund.de
‡E-mail: kuzmin@math.uni-dortmund.de

Received 12 September 2005
Revised 7 December 2005
Accepted 8 December 2005

Copyright © 2006 John Wiley & Sons, Ltd.

a posteriori error estimation has provided tools with which to verify that the model equation is solved accurately enough and/or to steer mesh adaptation. Starting with the pioneering work of Babuška and Rheinboldt [1] in the late seventies of the last century, theories and methods of *a posteriori* error estimation have been developed extensively [2–4]. It is noteworthy, that most of the research has focused on elliptic and parabolic linear problems in the framework of finite element approximations while for nonlinear hyperbolic partial differential equations the theory of *a posteriori* error estimation and adaptivity has remained in its infancy.

In a series of recent publications [5–9], a family of high-resolution schemes subsumed under the *algebraic flux correction* (AFC) paradigm was developed. In essence, a linear high order discretization, e.g. standard Galerkin scheme, is rendered *local extremum diminishing* (LED) by a conservative elimination of negative off-diagonal entries from the discrete transport operator so as to end up with a nonoscillatory low-order approximation. In order to recover the high accuracy of the original scheme a limited amount of compensating antidiffusion is added in regions of smooth solutions. The interested reader is referred to the aforementioned publications. The promising results obtained for scalar conservation laws as well as for the simulation of compressible inviscid and incompressible viscous flows on fixed unstructured grids have led us to combine flux limiting with an adaptive mesh refinement procedure in order to increase the resolving power of the algorithm. In addition, the computational cost can be drastically reduced if local mesh coarsening is employed in regions where the flow field is almost constant or its variance from node to node is quite small.

A common approach to the adaptive treatment of hyperbolic problems is based in Richardson extrapolation which is used to estimate the truncation error of the numerical scheme and equidistribute the error by means of local grid refinement [10–12]. In its original form, this technique requires the *a priori* knowledge of the order of approximation p. Roache [13] suggested a three grid refinement study to estimate p numerically. For high-resolution schemes based on flux/slope limiters, the approximation order varies locally such that special care must be taken [14]. The drawback of extrapolation-based error indicators is that this approach relies on the local smoothness of the solutions which cannot be guaranteed for hyperbolic systems of equations. Moreover, its entire relation to structured grids compromises the flexibility offered by finite elements which are applicable to unstructured triangular and/or quadrilateral grids.

As an alternative, either a smoothness sensor or the correction factors produced by the flux limiter can be used to steer grid adaptivity [15]. In order to prevent refinement due to microscopic jitters in the solution, also the curvature was taken into account. This approach was adopted to simulate Sod's transient shock tube problem in one dimension by means of finite differences. However, the employed indicator strongly depends on the properties of the limiter and is based solely on the error due to the discretization of convective terms. Hence, it is incapable of detecting insufficient grid resolution in regions where no flux limiting is required.

Recovery-based error estimators were first suggested by Zienkiewicz and Zhu [16], as early as in 1987. The 'simple error estimator for practical engineering analysis' presented for linear elastic problems was motivated by the observation that piecewise continuous finite element solutions generally exhibit discontinuous gradients at the element interfaces. Provided the 'true solution is sufficiently smooth' [17], these jumps in the gradient serve as an indicator for errors in the numerical solution. Several methods for recovering improved gradients have been proposed in the literature. Some of them, including the well-known Zienkiewicz–Zhu patch recovery technique [18, 19], rely on the superconvergence property of the finite element

Copyright © 2006 John Wiley & Sons, Ltd.

method at certain points. Their ease of implementation, robustness, and accuracy in many situations have boosted the popularity of recovery-based adaptive schemes especially in the engineering community. However, problems have been reported [20] applying this methodology to compressible flows using classical finite element or finite volume schemes. In essence, shock waves are typically smeared across several elements and captured as linear approximation with steep gradients. As a consequence, the jumps across element interfaces are very small and the error predicted by the recovery procedure tends to zero at the location of the 'discontinuity' [21]. Hence, mesh refinement is forced in the vicinity of the shock but not at its core. Yet, it is questionable if this phenomenon can be attributed to the error indicator based on gradient reconstruction or to the overly diffusive discretization scheme employed.

The reformulation of the element gradient error in terms of edge contributions allows for the application of slope limiting techniques which have been originally designed for a special treatment of convective terms $\nabla \cdot (\mathbf{v}u)$. Let us replace the velocity vector by the unit vector $\mathbf{e}_i$ in ith spatial direction one after another. Then the task of finding a good approximation to the convective term reduces to that of computing the ith component of the nodal gradient and *vice versa*. Based on our experience with algebraic flux correction schemes we derived two different approaches for the evaluation of edge gradients by means of slope limiting schemes. The gradient values at the edge midpoints can be directly computed as a limited average of consistent slopes adjacent to the corresponding edge. Moreover, standard recovery techniques may be employed to acquire smoothed nodal gradients from which provisional slopes can be interpolated along the edge. A slope limiter is applied edge-by-edge in order to adjust the intermediate values to the natural bounds set up by the constant gradient values from adjacent cells. This idea can be traced back to the concept of *flux corrected transport* (FCT) [5, 6], whereby a flux limiter was designed so as to restrict the high-order solution on the basis of upper and lower bounds stemming from an intermediate *positivity-preserving* (PP) solution.

2. *A POSTERIORI* ERROR INDICATORS

As a model problem, consider a partial differential equation of the following form:

$$\mathscr{L}u = f \quad \text{in } \Omega \tag{1}$$

where the (possibly nonlinear) differential operator $\mathscr{L}$ may consist of both spatial and time derivatives. The variational form of (1) is derived by first multiplying the residual of this equation by a weighting function w and integrating over the domain Ω

$$\int_\Omega w[\mathscr{L}u - f]\,\mathrm{d}\mathbf{x} = 0 \tag{2}$$

Let the exact solution u be approximated by means of finite elements

$$u \approx u_h = \sum_j u_j \varphi_j \tag{3}$$

where φ_j denotes the basis functions spanning the finite-dimensional subspace.

Any solution to Equation (2) represents an approximation to the original problem (1) involving all sorts of numerical errors, such as: integration errors, round-off errors, implementation

Copyright © 2006 John Wiley & Sons, Ltd.

errors (!), algorithmic errors, discretization errors, etc. In what follows, we shall concentrate on *spatial* errors due to the finite element discretization.

The numerical error relates the exact solution u of the continuous problem (1) to the nodal values u_h of the finite element approximation satisfying Equation (2)

$$e = u - u_h \tag{4}$$

It is well known, that obtaining an approximation to e yields a problem as complex as the one for u. Thus the main objective of *a posteriori* error estimation is not to get an approximation of the error defined in (4), but to estimate its magnitude in a suitable norm.

Instead of measuring the error of the solution, for some applications, e.g. convection-dominated flows, it may be useful to consider the error of the gradient. Let

$$\mathbf{e} = \sigma - \sigma_h \tag{5}$$

denote the vector-valued error in the gradient computed directly from the solution as

$$\sigma_h = \nabla u_h = \sum_j u_j \nabla \varphi_j \tag{6}$$

In what follows, we shall refer to σ_h as the low-order gradient. The aim of recovery-based estimators, introduced by Zienkiewicz and Zhu [16], is to replace the exact value σ, which in general is not known, by a smoothed gradient field $\hat{\sigma}_h$ (to be defined below), such that

$$\mathbf{e} \approx \hat{\mathbf{e}} = \hat{\sigma}_h - \sigma_h \tag{7}$$

gives a good approximation to the exact error defined in (5).

In general, pointwise error estimates are difficult to obtain, so integral measures are typically employed in the finite element framework. Different norms show different aspects of the error, and for convection-dominated problems, the question of choosing an appropriate norm has not been completely answered. A widely used integral measure is the standard L_2-norm

$$\|\hat{\mathbf{e}}\|_{L_2} = \left(\int_\Omega \hat{\mathbf{e}}^{\mathrm{T}} \hat{\mathbf{e}} \, \mathrm{d}\mathbf{x} \right)^{1/2} \tag{8}$$

Although the above integral measure is defined in the whole domain Ω, its square can be obtained by summing all element contributions over the triangulation $\mathscr{T}_h$ of Ω. Thus

$$\|\hat{\mathbf{e}}\|_{L_2}^2 = \sum_{T \in \mathscr{T}_h} \|\hat{\mathbf{e}}\|_{L_2(T)}^2 \tag{9}$$

where subscript $L_2(T)$ refers to the local L_2-norm computed on element $T \in \mathscr{T}_h$.

Since we employ piecewise linear ($\mathscr{P}_1$) trial functions φ for the approximation of the finite element solution, the discrete gradient σ_h is constant on each element and exhibits discontinuous jumps at element interfaces/vertices. Hence, the improved slopes should be at least piecewise linear so as to provide a better approximation to the exact gradient. To this end, it suffices to specify slope values at all midpoints of edges, i.e. $\mathbf{x}_{ij} := \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$, to obtain a smoothed quantity $\hat{\sigma}_h$ that varies linearly in each $T \in \mathscr{T}_h$ and is allowed to exhibit jumps across interelement boundaries. This approach can be seen as determining the nodal values for a nonconforming approximation of $\hat{\sigma}_h$ by means of linear Crouzeix–Raviart ($\tilde{\mathscr{P}}_1$) finite elements for which the local degrees of freedom are located at edge midpoints. For bilinear

Copyright © 2006 John Wiley & Sons, Ltd.

($\mathcal{Q}_1$) finite elements used on quadrilateral meshes, the gradient approximation can be based on the nonconforming Rannacher–Turek element ($\tilde{\mathcal{Q}}_1$).

In two space dimensions, the following second-order accurate Newton–Cotes quadrature rule may be used to compute the element contributions to the global gradient error

$$\int_T \hat{\mathbf{e}}^{\mathrm{T}} \hat{\mathbf{e}} \, \mathrm{d}\mathbf{x} = \frac{|T|}{3} \sum_{ij} \hat{\mathbf{e}}_{ij}^{\mathrm{T}} \hat{\mathbf{e}}_{ij} \tag{10}$$

where $|T|$ stands for the area of the triangle and the summation extends over the midpoints of the three surrounding edges. Let us single out the contribution of the edge $\vec{ij}$ to Equation (10)

$$\hat{\mathbf{e}}_{ij}^{\mathrm{T}} \hat{\mathbf{e}}_{ij} = [\hat{\sigma}_{ij} - \sigma_{ij}]^{\mathrm{T}} [\hat{\sigma}_{ij} - \sigma_{ij}] \tag{11}$$

Here, $\sigma_{ij} := \sigma_h(\mathbf{x}_{ij})$ is constant on the whole element and $\hat{\sigma}_{ij} := \hat{\sigma}_h(\mathbf{x}_{ij})$ varies linearly along the edge. It remains to devise a procedure for constructing improved gradient values $\hat{\sigma}_{ij}$.

3. GRADIENT RECONSTRUCTION

Our first approach to obtaining a smoothed edge gradient is largely inspired by slope limiting techniques employed in the context of high-resolution finite volume schemes and later carried over to discontinuous Galerkin finite element methods [22]. Various attempts to extend slope limiting to multidimensions can be found in the literature. In essence, the task is to reconstruct the slopes at interelement boundaries where discrete solution values exhibit jumps. However, geometric constraints need to be satisfied in order to guarantee that the numerical solution is free of nonphysical oscillations which would be generated otherwise. To this end, the value of the recovered gradient is taken as a limited average of constant slopes adjacent to edge $\vec{ij}$.

As an alternative, a provisional gradient at the midpoint of edge $\vec{ij}$ can be linearly interpolated from nodal values: $\hat{\sigma}_{ij} = \frac{1}{2}(\hat{\sigma}_i + \hat{\sigma}_j)$. Sophisticated projection or discrete patch recovery techniques can be employed to compute smoothed slopes at the element vertices. However, the resulting edge gradient may violate the natural bounds given by the first-order slopes σ_h of the two adjacent cells. This can be rectified by applying a slope limiter edge-by-edge so as to satisfy geometric constraints.

3.1. Limited gradient averaging

For simplicity, we will discuss the basic ideas of slope-limited finite volume methods in one space dimension. Let the interval $I = \bigcup_{j=1}^{m} I_j$ be partitioned into a set of finite volumes $I_j = (x_{j-1/2}, x_{j+1/2})$ and let $\overline{u}_j$ denote the mean value of some scalar quantity u on cell I_j. The task is to construct a piecewise linear approximate solution

$$\tilde{u}_h(x) = \overline{u}_j + (x - x_j)\sigma_j \quad \forall x \in I_j \tag{12}$$

where σ_j denotes an approximation of the solution gradient on the jth cell. In the simplest case, one-sided or centred slopes have been employed to obtain first- and second-order accurate schemes, respectively. However, oscillations are quite likely to appear in the second case while first-order schemes lead to rather diffusive solution profiles. For a numerical scheme to be nonoscillatory, it should possess certain properties, e.g. be monotone, positivity preserving

Copyright © 2006 John Wiley & Sons, Ltd.

or total variation/local extremum diminishing. This can be accomplished by replacing the approximate slope value σ_j in Equation (12) by its *limited* counterpart.

For the construction of LED and TVD schemes, Jameson utilized limited average operators $\mathscr{L}(a,b)$ which are characterized by the following properties [23]:

P1. $\mathscr{L}(a,b)=\mathscr{L}(b,a)$,
P2. $\mathscr{L}(ca,cb)=c\mathscr{L}(a,b)$,
P3. $\mathscr{L}(a,a)=a$,
P4. $\mathscr{L}(a,b)=0$ if $ab\leqslant 0$.

While the first three conditions are natural properties of an average, P4 is to be enforced by means of limiting. Jameson demonstrated that a variety of standard TVD limiters can be written in such a form. Let the modified sign function be given by

$$\mathscr{S}(a,b)=\frac{\operatorname{sign}(a)+\operatorname{sign}(b)}{2} \tag{13}$$

which equals zero for $ab\leqslant 0$ and returns the common sign of a and b otherwise. Then the most widely used two parameter limiters for TVD schemes can be written as:

1. minmod: $\mathscr{L}(a,b)=\mathscr{S}(a,b)\min\{|a|,|b|\}$.
2. maxmod: $\mathscr{L}(a,b)=\mathscr{S}(a,b)\max\{|a|,|b|\}$.
3. MC: $\mathscr{L}(a,b)=\mathscr{S}(a,b)\min\{\frac{|a+b|}{2},2|a|,2|b|\}$.
4. superbee: $\mathscr{L}(a,b)=\mathscr{S}(a,b)\max\{\min\{2|a|,|b|\},\min\{|a|,2|b|\}\}$.

In light of the above, the limited counterpart of σ_j in Equation (12) can be computed as follows:

$$\sigma_j:=\mathscr{L}\left(\frac{\overline{u}_j-\overline{u}_{j-1}}{\Delta_j},\frac{\overline{u}_{j+1}-\overline{u}_j}{\Delta_j}\right) \tag{14}$$

where the cellwidth of the jth subinterval is denoted by $\Delta_j=x_{j+1/2}-x_{j-1/2}$.

Let us return to our original task that requires the computation of the solution slopes at the midpoint of edge $\vec{ij}$ so as to estimate its contribution (11) to the local error. Let σ_{ij}^{+} and σ_{ij}^{-} denote the piecewise constant gradient values evaluated on the two elements to the left and to the right of edge $\vec{ij}$, respectively. Then, the auxiliary quantities

$$\sigma_{ij}^{\substack{\max\\ \min}}=\max_{\min}\{\sigma_{ij}^{+},\sigma_{ij}^{-}\} \tag{15}$$

provide excellent lower and upper bounds that should be satisfied by any gradient value along the edge. Moreover, each of the limited average operators presented above can be utilized to obtain a usable edge gradient that can be computed efficiently as follows:

$$\hat{\sigma}_{ij}=\mathscr{L}(\sigma_{ij}^{+},\sigma_{ij}^{-}) \tag{16}$$

For all limiter functions $\mathscr{L}$ presented above, the recovered gradient value equals zero if $\sigma_{ij}^{+}\sigma_{ij}^{-}\leqslant 0$ and satisfies the following inequality otherwise:

$$\sigma_{ij}^{\min}\leqslant\hat{\sigma}_{ij}\leqslant\sigma_{ij}^{\max} \tag{17}$$

If the upper and lower bounds have different signs, this indicates that the approximate solution attains a local extremum across the edge. Hence, property P4 of limited average operators

Copyright © 2006 John Wiley & Sons, Ltd.

acts as a discrete analog to the necessary condition in the continuous case which requires the derivative to be zero. Clearly, the recovered gradient (16) depends on the choice of the limiter function to some extent. In the authors' experience, the monotonized centred (MC) limiter seems to be a safe choice as it tries to select the standard average whenever possible without violating the natural bounds (15) provided by the low-order slopes.

3.2. Nodal gradient recovery

The idea of using recovery techniques to obtain improved gradient/stress values exhibits quite a long tradition in finite elements (cf. Oden *et al.* [24, 25] or Hinton and Campbell [26]). To the authors' best knowledge, Cantin *et al.* [27] were the first to consider the so-called *averaging projection* schemes which have also been utilized by Zienkiewicz and Zhu [16] in their original paper on recovery-based error estimation. Let the smoothed gradient be approximated by

$$\hat{\sigma} = \sum_j \hat{\sigma}_j \phi_j \tag{18}$$

where the coefficients $\hat{\sigma}_j$ are obtained by solving the discrete problem

$$\int_\Omega \phi_i (\hat{\sigma} - \sigma_h)\, \mathrm{d}\mathbf{x} = 0 \tag{19}$$

Note that the element shape functions used to construct the basis functions ϕ_i onto which σ_h is projected may be different from those employed in the finite element approximation (3). A detailed analysis of projection-based error estimators by Ainsworth *et al.* [17] reveals that the corresponding polynomial degrees should satisfy $\deg \phi \geqslant \deg \varphi$. Like in an earlier publication by Oden and Brauchli [24], they conclude that the recovery procedure utilized in Reference [16], which corresponds to choosing $\phi \equiv \varphi$ in the equations above, 'is not only effective, but also the most economical' one. Substitution of Equation (18) into (19) yields

$$\sum_j \left[\int_\Omega \phi_i \phi_j \, \mathrm{d}\mathbf{x} \right] \hat{\sigma}_j - \sum_j \left[\int_\Omega \phi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x} \right] u_j = 0 \quad \forall i \tag{20}$$

Thus, the smoothed gradient can be recovered by solving the linear algebraic system

$$M_C \hat{\sigma} = \mathbf{C} u \tag{21}$$

where $M_C = \{m_{ij}\}$ denotes the consistent mass matrix and $\mathbf{C} = \{\mathbf{c}_{ij}\}$ is the matrix of discretized spatial derivatives. The coefficients of these matrices are given by

$$m_{ij} = \int_\Omega \phi_i \phi_j \, \mathrm{d}\mathbf{x}, \quad \mathbf{c}_{ij} = \int_\Omega \phi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x} \tag{22}$$

For a fixed mesh, the coefficients m_{ij} and $\mathbf{c}_{ij}$ remain unchanged throughout the simulation and, consequently, need to be evaluated just once during the initialization step and each time the grid has been modified. If $\phi \equiv \varphi$, the coefficients defined in (22) coincide with the matrix entries of the finite element approximation and, hence, are available at no additional cost.

An edge-by-edge assembly of the right-hand side is also feasible

$$(\mathbf{C}u)_i = \sum_{j \neq i} \mathbf{c}_{ij} (u_j - u_i) \tag{23}$$

Copyright © 2006 John Wiley & Sons, Ltd.

since $\mathbf{C}$ features the zero row sum property $\sum_j \mathbf{c}_{ij} = 0$ as long as the sum of the basis functions equals one. Note that for $\phi \equiv \varphi$, the algebraic system (21) can also be obtained by applying the standard Galerkin approximation to the weak form of the continuous problem $\sigma = \nabla u$. Thus, projection schemes of the form (18)–(19) are called *variational recovery* [28] and can be applied repeatedly so as to determine an approximation to a higher-order derivative.

The solution to the algebraic system (21) can be computed iteratively by successive approximation preconditioned by the lumped mass matrix $M_L = \{m_i\}$, where $m_i = \sum_j m_{ij}$

$$\hat{\sigma}^{(m+1)} = \hat{\sigma}^{(m)} + M_L^{-1}[\mathbf{C}u - M_C\hat{\sigma}^{(m)}], \quad m = 0, 1, 2, \ldots \tag{24}$$

If mass lumping is applied directly to Equation (21), the values of the projected gradient can be determined at each node from the explicit formula

$$\hat{\sigma}_i = \frac{1}{m_i} \sum_{j \neq i} \mathbf{c}_{ij}(u_j - u_i) \tag{25}$$

Over the years, a more accurate patch recovery technique was introduced by Zienkiewicz and Zhu [18, 19], which relies on the superconvergence property of the finite element solution at some exceptional, yet *a priori* known, points. Let the smoothed gradient be represented in terms of a polynomial expansion of the form

$$\hat{\sigma} = p(\mathbf{x})\,\mathbf{a} \tag{26}$$

where for two space dimensions $p(\mathbf{x}) = [1, x, y, x^2, \ldots, x^k, x^{k-1}y, \ldots, xy^{k-1}, y^k]$ contains the monomials of degree k at most. Since each vertex is surrounded by a patch of elements sharing this node, the vector of coefficients $\mathbf{a} = [a_1, a_2, \ldots, a_m]^{\mathrm{T}}$ with $m = (k+1)(k+2)/2$ can be computed from a discrete least square fit to the set $\mathscr{S}_i$ of sampling points $\mathbf{x}_j$ [18]. From that it follows, that the coefficient vector $\mathbf{a}$ is the solution to the linear system

$$M_p\mathbf{a} = \mathbf{f} \tag{27}$$

where the local matrix M_p and the right-hand side vector $\mathbf{f}$ are given by

$$M_p = \sum_{j \in \mathscr{S}_i} p(\mathbf{x}_j)^{\mathrm{T}} p(\mathbf{x}_j), \quad \mathbf{f} = \sum_{j \in \mathscr{S}_i} p(\mathbf{x}_j)^{\mathrm{T}} \sigma_h(\mathbf{x}_j) \tag{28}$$

For linear triangles, $\mathbf{a} = [a_1, a_2, a_3]^{\mathrm{T}}$, $p = [1, x, y]$ and the gradient is sampled at the centroid $\mathbf{x}_j$ of each triangle in the patch. In this case the lumped L_2-projection yields almost the same results on uniform grids but only patch recovery retains its superconvergence property if the grid gets distorted which in general makes (26) superior to (25). However, the solvability of the linear system (27) strongly depends on the relation rank $M_p = m$.

Since the advent of the *superconvergent patch recovery* (SPR) technique [18] its super- and even ultraconvergence property has been analysed extensively in the literature [29–31]. This paved the way to the development of the so-called *polynomial preserving* (PPR) gradient recovery schemes [32]. While in SPR methods a $\mathscr{P}_k$-polynomial is best fitted to σ_h directly, PPR schemes compute the nodal quantity $p \in \mathscr{P}_{k+1}$ as a polynomial approximation to u_h and apply the derivative operator afterwards ($\hat{\sigma}_h = \nabla p$). In order to ensure the solvability of the linear system (27), the patch of surrounding elements needs to be enlarged recursively. Recently, Zhang *et al.* introduced a 'meshless' gradient recovery method [33] in which the

Copyright © 2006 John Wiley & Sons, Ltd.

idea of element patches is abandoned in favour of spherical patches which are expanded adaptively so as to satisfy a solvability condition.

The ease of implementation, generality and ability to produce quite accurate estimators boosted the popularity of recovery-based techniques especially in the engineering community. However, any of the above-mentioned strategies to compute a high-order gradient from the finite element solution is quite likely to fail either for steep gradients or in case the solution exhibits jumps as it is often the case in compressible flow computations featuring shock waves and contact discontinuities. This can be attributed to the fact, that the consistent L_2-projection scheme tends to produce nonphysical oscillations in the vicinity of jumps whereas both its lumped counterpart and the patch recovery are overly diffusive. This drawback of the standard procedures can be rectified by combining both imperfect methods as explained below.

3.3. Edgewise slope limiting

No matter if patch recovery or averaging projection schemes are employed, the nodal gradient values result from an averaging process over an *unsettled* number of surrounding element gradients which may strongly vary in magnitude and even possess different signs. Thus, it is very difficult to find admissible upper and lower bounds to be imposed on the recovered *nodal* gradient. Let us recall, that in order to compute the element gradient error (10) we have to sum the contributions (11) of the adjacent *edges*. In the interior, each edge can be associated with exactly *two* triangles sharing it. Hence, the auxiliary quantities defined in (15) constitute natural upper and lower bounds for the final edge gradient, such that

$$\sigma_{ij}^{\min} \leqslant \hat{\sigma}_{ij} \leqslant \sigma_{ij}^{\max} \tag{29}$$

In the first step, provisional edge gradient values are *predicted* at the midpoint of edges which are located in the overlap of two element patches. Hence, the intermediate edge slopes can easily be computed by linear interpolation of nodal values resulting from any of the above nodal recovery schemes, i.e. $\hat{\sigma}_{ij} = \frac{1}{2}(\hat{\sigma}_i + \hat{\sigma}_j)$. As an alternative, patch recovery can be used to obtain the midpoint gradient values directly. In the next step, the provisional edge gradients are *corrected* subject to the upper and lower bounds (15)

$$\hat{\sigma}_{ij}^{*} = \max\{\sigma_{ij}^{\min}, \min\{\hat{\sigma}_{ij}, \sigma_{ij}^{\max}\}\} \tag{30}$$

The edgewise slope-limiting procedure is illustrated in Figure 1 for an interior edge. In one space dimension, Zienkiewicz and Zhu observed, that 'the recovered derivative nodal values [computed by either lumped L_2-projection or patch recovery] for linear elements are located between the discontinuity of the finite element solution' [18]. The task of our edgewise slope limiter is to enforce this property in the sense of a multidimensional extension. The generality of this concept allows for the application of any nodal gradient recovery procedure proposed in the literature or even a combination thereof.

Let us consider the situation when the upper and lower bounds (15) have different signs. This indicates that the approximate solution attains a local minimum/maximum across the edge. In the continuous case, the necessary condition of an extremum requires that the corresponding derivatives be equal to zero. For the recovered gradient to satisfy a discrete analog,

Copyright © 2006 John Wiley & Sons, Ltd.

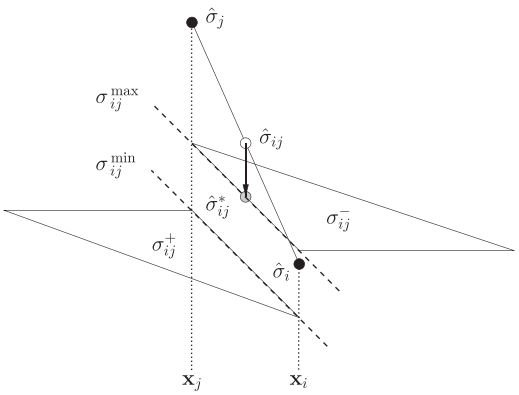

Figure 1. Edgewise slope-limited gradient recovery for internal edges.

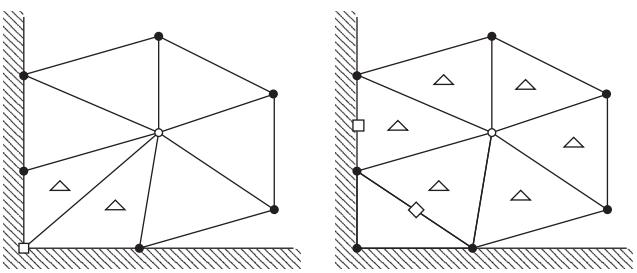

Figure 2. Boundary treatment: nodal vs. edgewise recovery.

it makes sense to modify Equation (30) as follows:

$$\hat{\sigma}_{ij}^{*} = \mathscr{S}(\sigma_{ij}^{\min}, \sigma_{ij}^{\max}) \, |\max\{\sigma_{ij}^{\min}, \min\{\hat{\sigma}_{ij}, \sigma_{ij}^{\max}\}\}| \tag{31}$$

This adjustment corresponds to property P4 of limited average operators (see above).

Boundary treatment: In the framework of *nodal* recovery procedures, the gradient values at boundary vertices can be reconstructed from specially designed boundary patches. However for a corner node, say '□', an insufficient number of elements denoted by '△' can render the matrix of system (27) singular as illustrated in Figure 2 (left). Zienkiewicz and Zhu recommend *always* recovering the nodal values at the boundary from an interior patch recovery point '○' [18] which for unstructured triangulations is far from being unique.

In the context of our edge-based formulation, it is natural to adopt the constant slope value from the adjacent element, say '△', also at the midpoint '□' of a boundary edge. As an alternative, the gradient value of a boundary edge can be recovered from the *unique* patch assembly point '○' opposite to it as depicted in Figure 2 (right).

The same applies to an interior edge, say '◇', that belongs to a triangle with three boundary nodes. Then the edge slope value can be recovered from the uniquely defined patch assembly point '○' connected to the adjacent 'interior' triangle.

Copyright © 2006 John Wiley & Sons, Ltd.

3.4. Summary of reconstruction schemes

Let us summarize the different approaches for recovering an improved gradient value $\hat{\sigma}_{ij}$ at the midpoint of edge $\vec{ij}$ and introduce abbreviated forms for each of there techniques.

Improved gradient values $\hat{\sigma}_{\mathscr{L}}$ at edge midpoints can be directly computed from formula (16) as the limited average of the constant slopes $\sigma_{ij}^{\pm}$ to the left and to the right of edge $\vec{ij}$.

As an alternative, traditional recovery procedures can be employed to predict (nodal) gradient values $\hat{\sigma}_i$ from which slopes at midpoints of edges can be computed by linear interpolation $\hat{\sigma}_{ij} = \frac{1}{2}(\hat{\sigma}_i + \hat{\sigma}_j)$. In what follows, $\hat{\sigma}_{M_C}$, $\hat{\sigma}_{M_L}$ and $\hat{\sigma}_{ZZ}$ will denote gradient values which result from consistent (24) or lumped (25) L_2-projection schemes and the ZZ patch recovery technique (26), respectively. Their corrected counterparts which satisfy the inequalities in (29) are indicated by superscript $*$ and computed by the slope limiter (30).

4. MESH ADAPTIVITY

For CFD problems, the flow pattern is governed by the propagation and interaction of localized structures which dominate the error to a large extent. This observation suggests adaptive mesh refinement as a useful tool for the treatment of hyperbolic conservation laws. First, disturbances are propagated along characteristics with finite speed such that adaptive mesh refinement is most likely to improve the solution locally without affecting its global behaviour. Second, the nonlinear nature of the equations at hand gives rise to the formation of discontinuities which call for the use of nonoscillatory approximations. It follows, that local grid refinement would improve the resolution of shocks significantly. These peculiarities of hyperbolic problems must be accounted for in the process of mesh adaptation, since they affect both the selection of elements to be refined and coarsened and the choice of refinement strategies.

4.1. Adaptation strategy

In adaptive solution procedures for steady state simulations of hyperbolic systems of equations, one typically starts with a moderately coarse grid on which an initial solution can be computed efficiently. Nevertheless, the mesh needs to be fine enough to capture all essential flow features so as to enable the error indicator to detect 'imperfect' regions. As pointed out in the introduction, the misfortune experienced with applying the standard Zienkiewicz–Zhu error estimator to an adaptive shock wave simulation [21] may be attributed to the hapless interplay of overly diffusive spatial discretization schemes applied on insufficiently fine triangulations. In other words, the artificial dissipation introduced by the numerical method overstrained the resolution facility of the employed coarse grid and misled the error indicator.

For all examples presented in this article, the GiD mesh generator [34] which is based on the advancing front concept is used to create the coarse grid for geometric multigrid schemes [35]. A hierarchical mesh data structure for the initial grid is generated by successive subdivision of each triangle of the coarse grid into four subelements of equal size. To speed up the steady state convergence of the initial solution, nested iterations [36] are used. The idea is to recursively compute a provisional solution on a coarser mesh and interpolate it to the next finer level so as to obtain a reasonable initial guess. This procedure, also referred to as full multigrid (FMG) [37], has proven to be quite expedient for the simulation of steady compressible flows.

Copyright © 2006 John Wiley & Sons, Ltd.

The relative error of the density has been employed to monitor steady state convergence [38]. Following Reference [39], the flow solver is stopped if this error falls below the square root of the prescribed tolerance, that is, intermediate solutions are required to be only 'half-converged'. The (more expensive) computation of a fully converged solution is required on the final grid.

In the next step, cells are flagged for local refinement or coarsening according to some adaptation parameters. A common practice is to prescribe the tolerance for the relative percentage error of the solution and the gradient, respectively,

$$\eta := \frac{\|\mathbf{e}\|_{L_2}}{\|\nabla u\|_{L_2}} \leqslant \eta_{\text{ref}} \tag{32}$$

Since neither the exact slope values nor the true error are known, the best approximation available is utilized instead. From (9) it follows that the global L_2-norm can be decomposed into element contributions. Moreover, let us assume that the relative error is equally distributed between cells. Then the condition $\eta \leqslant \eta_{\text{ref}}$ (cf. (32)) can be rewritten as follows:

$$\|\hat{\mathbf{e}}\|_{L_2(T)} \leqslant \eta_{\text{ref}} \left[\frac{\|\sigma_h\|_{L_2}^2 + \|\hat{\mathbf{e}}\|_{L_2}^2}{|\mathscr{T}_h|} \right]^{1/2} =: e_{\text{ref}} \tag{33}$$

where $|\mathscr{T}_h|$ denotes the number of elements which constitute the current triangulation $\mathscr{T}_h$.

A similar estimate in terms of the solution error is formulated in Reference [16]. Given the ratio

$$\xi_T = \frac{\|\hat{\mathbf{e}}\|_{L_2(T)}}{e_{\text{ref}}} \tag{34}$$

an element T is marked for refinement if $\xi_T > 1$. At the same time, the cells to be coarsened can be determined by inverting all inequalities and replacing η_{ref} by some $\eta_{\text{crs}} \ll \eta_{\text{ref}}$.

A suitable refinement algorithm (see below) is employed to locally adapt the mesh in regions of underresolved wave patterns and the current solution is projected onto the refined mesh. If mesh coarsening should be performed in regions of uniform flow the use of a conservative projection scheme such as the one presented in Reference [40] is recommended at least for transient problems. Once the adapted mesh has been created, the solution is marched to the stationary limit and the whole process starts again until the final mesh has been reached. To this end, one or more 'mesh convergence' criteria need to be defined. The simplest approach for steady state problems is to prescribe the maximum number of refinement levels *a priori*. If a quantitative analysis of the error is available it is advisable to stop the adaptation process if the global error has reached some tolerance. As an alternative, the (relative) changes of the solutions on two successive grids can be used to terminate the simulation if the difference is sufficiently small.

4.2. Grid refinement techniques

The grid refinement strategy follows the algorithm for regular local mesh refinement proposed by Bank *et al.* [41]. In a loop over elements, cells which are marked for refinement by the error indicator are subdivided into four similar triangles. This so-called '*red*' *refinement* is applied iteratively so as to eliminate adjacent cells with two or three hanging nodes. In order to restore global regularity of the triangulation '*green*' *refinement* is applied afterwards

Copyright © 2006 John Wiley & Sons, Ltd.

to all 1-irregular [42] elements, that is, the midpoints of bisected edges are connected to the opposite vertices. Prior to the next adaptation cycle, all edges which have been introduced due to green refinement can/should be removed to retain the shape regularity of the successively refined grids. The regular local mesh refinement algorithm for triangles and quadrilaterals is dealt with in Reference [41] including a detailed description of efficient data structures. The red–green refinement strategy has been employed for all simulation results presented in this publication.

An alternative class of conforming mesh refinement algorithms for triangular/tetrahedral grids is based on edge bisection and was originally introduced by Rivara [43]. For each element flagged for refinement, a new node is inserted at the midpoint of the longest edge and connected to the opposite vertex. The bisection process continues recursively for all adjacent triangles sharing a hanging node with the refined element until all nonconforming vertices have been eliminated. A nonrecursive variant has been suggested in Reference [44] where the longest-edge propagation path is computed *a priori* and a backward algorithm is employed to perform edge bisection. Some *geometric* properties of these methods can be found in Reference [45]. In fact, the classical longest-edge bisection approach is probably not the best choice as far as algebraic flux correction (AFC) [8, 9] is concerned which may be explained as follows. For each element that needs to be refined due to accuracy reasons, the propagation path which may extend far away from the originating triangle depends *solely* on the mesh geometry and does not account for the solution behaviour. Recall that our AFC methodology [8, 9] entirely rests on an edge-based formulation. The amount of artificial dissipation that outlasts the flux limiting procedure depends on the *interplay* of internodal fluxes which are proportional to the edgewise solution difference multiplied by some (anti-)diffusion coefficient.

If the solution variation along the longest edge is small or even negligible then bisection will hardly facilitate the task of the flux limiter and neither will it improve the resolution appreciably. Instead, it is worthwhile to refine the edge with the largest solution variation or the largest antidiffusive flux. Consider the situation in which this flux into one node, say i, cannot be balanced by diffusive fluxes from neighbouring nodes so that its magnitude needs to be drastically reduced. In this case, edge $\vec{ij}$ should be bisected, unless this would entail a reduction of the correction factor for node j. In a forthcoming paper, this *algebraic* edge partition approach which is tailored to the peculiarities of AFC schemes [8] will be embedded into a more theoretical framework and its numerical performance will be analysed.

4.3. Grid coarsening techniques

Mesh coarsening is applied in regions of sufficiently uniform flow where the relative gradient error (33) is below some prescribed tolerance η_{crs}. For all simulations presented in this article, the vertex removal procedure described in Reference [40] has been employed. In essence, edge-swapping is performed repeatedly so as to 'isolate' the vertex to be erased. The iteration continues until the corresponding node is connected to just three triangles and can be safely removed. If the vertex to be deleted resides on the boundary it is first 'moved' into the interior by introducing an artificial boundary element before the standard procedure can be applied.

In the context of algebraic flux correction methods which rely on an edge-based data structure, edge collapse techniques [46] may be a promising alternative. This technique has been extensively discussed in computer graphics literature (cf. [47]). The basic idea is to

Copyright © 2006 John Wiley & Sons, Ltd.

contract edges and consolidate the two adjacent nodes. In general, different strategies [48] exist for positioning the newly created vertex which can be combined to improve the geometric quality of the resulting mesh. For an AFC scheme, the new vertices should be positioned so as to improve the algebraic properties of the discretization.

4.4. Grid improvement techniques

Typically, edge-swapping is employed as a postprocessing step to improve the mesh quality with respect to some geometric measures, e.g. the normalized shape regularity functional [49]

$$q(T) = 4\sqrt{3}|T| \Big/ \sum_{ij} \ell_{ij}^2 \tag{35}$$

Here, ℓ_{ij} denotes the length of the edge from node i to node j. This is where algebraic aspects come into play. For interior edges, the sum of opposite angles should not exceed π so as to guarantee that the coefficient matrix resulting from a piecewise linear Galerkin finite element approximation of the Laplacian operator is an M-matrix [50]. For boundary edges the opposite angle is required to be less than $\pi/2$. Edge-swapping can be equipped with algebraic quality measures of such kind so as to guarantee that all off-diagonal entries of the discrete diffusion operator remain nonnegative.

In addition, 'smart' Laplacian or optimization-based mesh smoothing [51] can be driven by algebraic quality measures in order to 'minimize' the amount of required artificial diffusion. The knowledge about suboptimal regions of the computational mesh is already 'hidden' in the matrix coefficients, and only needs to be retrieved. In other words, the flux limiter not only prevents the birth and growth of oscillations on a given (suboptimal) mesh but also provides valuable information for further mesh improvement.

4.5. Summary of the algorithm

Starting from an initial coarse grid that is supposed to be fine enough to capture essential flow features, the algorithmic steps of our adaptive AFC schemes are as follows:

1. Generate the required adjacency lists for nodes and elements and initialize the edge-based data structure. Assemble the constant coefficient matrices resulting from the Galerkin discretization of the variational problem (2) which resemble the ones in (22).
2. Compute the numerical solution on the current mesh
   - Enforce the positivity constraint by means of algebraic flux correction [8].
   - Solve the resulting nonlinear system using an iterative defect correction procedure.
   - Compute the relative error of some indicator variable, i.e. density, to check if the solution has 'half-converged'. Otherwise, continue flux/defect correction.
3. Evalute the consistent gradient (6) and recover improved slope values for each edge either as a limited average of constant slopes (16) or by means of edgewise slope limiting (30)/(31) applied to the average of smoothed nodal gradients (cf. (24)–(25) or (26)–(27)).
4. Assemble the L_2-norm of the element gradient error (10) from the edge contributions (11) and refine/coarsen all triangles according to conditions (33)–(34).

Copyright © 2006 John Wiley & Sons, Ltd.

5. *Optionally*: Apply edge-swapping or other grid improving methods in order to increase the mesh quality with respect to geometric or algebraic quality measures.
6. If the final/'converged' grid has been constructed, proceed to steps 1–2 until the fully converged solution is obtained. Otherwise, continue with steps 1–6.

5. NUMERICAL EXAMPLES

In order to demonstrate the behaviour of the edgewise slope-limited recovery procedure let us start with the investigation of one-dimensional profiles. Each of the following academic examples is designed so as to illustrate the deficiencies of standard recovery procedures *per se*. In contrast, the edgewise slope limited recovery outperforms its linear counterparts.

In the second part of this section, the new error indicator is applied to compressible inviscid flows at different Mach numbers. In previous publications, the authors presented numerical results for some of these benchmarks computed on fixed meshes in order to illustrate the performance of modern high-resolution finite element schemes based on the *algebraic flux correction* (AFC) paradigm. In this paper, emphasis is placed on grid adaptivity so that for all simulations the same TVD type algorithm is utilized in conjunction with the moderately diffusive CDS-limiter ($\Phi(\theta) = \min(1, 2\theta)$) applied to the characteristic variables. In contrast to schemes of FCT type, the amount of artificial diffusion remaining after an upwind-biased flux limiter has been employed does not depend on the size of the time step. For a detailed comparison of flux limiting schemes of TVD and FCT type, the interested reader is referred to References [6–8]. Since we only consider *steady state flows* which call for a fully implicit time discretization, i.e. the unconditionally stable backward Euler method, the time step should be taken as large as possible in order to rapidly reach a converged solution. Grid adaptivity only needs to be performed each time the flow has 'halfway' converged which does not increase the overall computational costs considerably. For all benchmarks, the error indicator is applied to the density which serves as a key variable [52]. A nonconservative projection scheme may be employed in order to transfer the old solution to the newly generated grid.

5.1. One-dimensional profiles

Example 1

Let us start with the classical *hat function* given by

$$u(x) = \begin{cases} 1 - r_0^{-1}|x - x_0|, & x \in (0.3, 0.7) \\ 0, & x \in [0, 0.3] \cup [0.7, 1] \end{cases} \tag{36}$$

where $x_0 = 0.5$ and $r_0 = 0.2$ as depicted in Figure 3. Obviously, the exact gradient (not displayed) exhibits three discontinuities at $x \in \{0.3, 0.5, 0.7\}$ and is constant elsewhere. The values of the finite element gradient σ_h which serve as upper/lower bounds are denoted by dots in all plots of Figure 4. It can be clearly seen from diagram (a) that the high-order gradient breaches the admissible bounds in the vicinity of the discontinuities and thus suffers from nonphysical oscillations. As depicted in (b), the gradient resulting from either the lumped L_2-projection or the discrete patch recovery, which yield indistinguishable results on uniform meshes, is completely free of under- and overshoots. Obviously, it stays within the bounds

Copyright © 2006 John Wiley & Sons, Ltd.

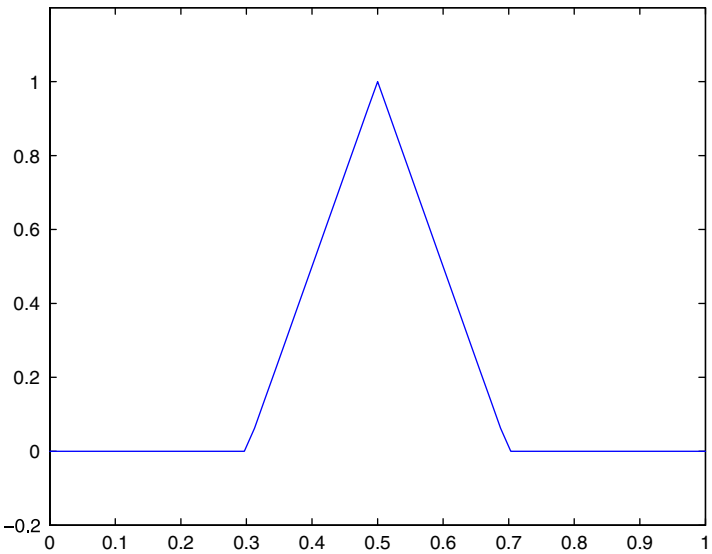

Figure 3. One-dimensional hat function.

from the outset (see also remark (ii) in Reference [18]) but is less accurate. In contrast, edgewise slope-limiting (c) combines the advantages of both techniques: high accuracy *and* bounded results. Finally, the improved gradient $\hat{\sigma}_{\mathscr{L}}$ computed directly by means of limited averaging (16) of constant slope values σ_h is depicted in Figure 4(d). Remarkably, the results recovered by the *monotonized centred* (MC) limiter very much resemble the edgewise slope limited gradient $\hat{\sigma}^*_{MC}$ in both accuracy and the fact, that no undershoots and overshoots take place.

Example 2
Our next example deals with the *normal distribution function*

$$u(x) = (4\pi\varepsilon)^{-1} \mathrm{e}^{-(x-x_0)^2/4\varepsilon} \quad \text{in } [0,1] \tag{37}$$

where again $x_0 = 0.5$. As can be seen from Figure 5, the solution profile (a) is smooth but features strong gradients (b), two inflection points and a local extremum. The parameter $\varepsilon = 0.005$ is chosen such that the extrema of the gradient are located at $x \in \{0.4, 0.6\}$, where the curvature of u changes its sign. Initially, the interval $[0,1]$ is uniformly discretized with linear finite elements of size $h = 0.1$. In order to study the nodal rate of convergence, regular subdivision is applied until the mesh size reaches $h = 0.0004$.

The convergence of the consistent finite element gradient σ_h ('•') and its recovered counterparts are illustrated in Figure 6. From left to right, the absolute error has been measured at the boundary $x = 0$, at the local maximum $x = 0.4$ and at the point $x = 0.7$ located in a smooth region. Since σ_h exhibits discontinuous jumps across element boundaries, we have always chosen the value giving the maximum absolute error. At the boundary, only the discretely recovered $\hat{\sigma}_{ZZ}$ exhibits superconvergence while the convergence rate of all other schemes degenerates to $O(h)$. Consistent L_2-projection yields a slightly smaller error as compared to the

Copyright © 2006 John Wiley & Sons, Ltd.

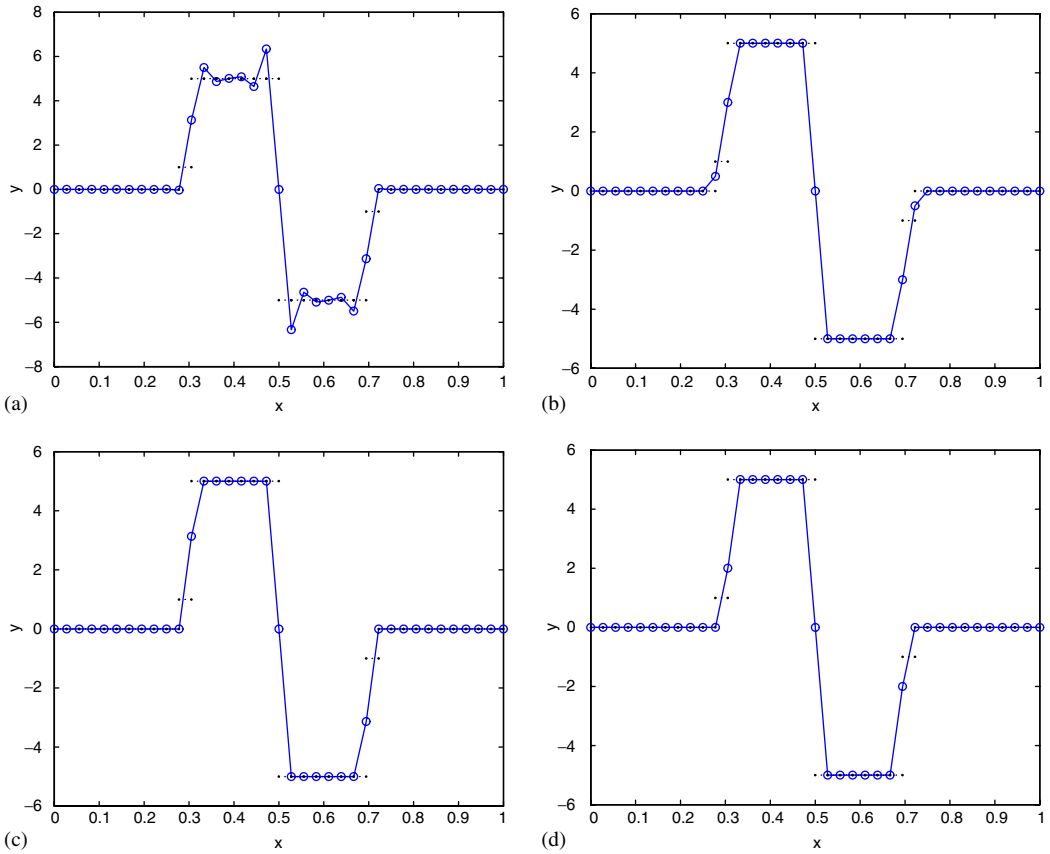

Figure 4. One-dimensional hat function: Recovered gradients: (a) $\hat{\sigma}_{M_C}$; (b) $\hat{\sigma}_{M_L}/\hat{\sigma}_{ZZ}$; (c) $\hat{\sigma}^*_{M_C}$; and (d) $\hat{\sigma}_{\mathscr{L}}$.

rest of linearly converging methods. Due to the lack of appropriate bounds, no slope limiting is performed at the boundary so that $\hat{\sigma}^*_{M_C} := \hat{\sigma}_{M_C}$.

From Figure 6 (middle) we observe, that all schemes are at least superconvergent with only negligible differences in terms of the absolute error at local extrema. Unfortunately, the ultraconvergence of the consistent L_2-projection does not carry over to its slope-limited counterpart. Indeed, *peak clipping* is a well known phenomenon in the context of limiting procedures [53] which can be attributed to the fact that the upper and lower bounds (15) are too restrictive to preserve the accuracy of the original high-order scheme.

The nodal rate of convergence for smooth gradients is depicted in Figure 6 (right). Obviously, σ_h converges only linearly whereas the gradients resulting from discrete patch recovery, lumped L_2-projection and MC-limited averaging of constant slopes exhibit $O(h^2)$ convergence rates. Furthermore, the ultraconvergence of $\hat{\sigma}_{M_C}$ carries over to its slope limited counterpart $\hat{\sigma}^*_{M_C}$. It is noteworthy, that limited averaging of constant slopes yields results competitive to those produced by discrete patch recovery for interior edges. This observation implies that

Copyright © 2006 John Wiley & Sons, Ltd.

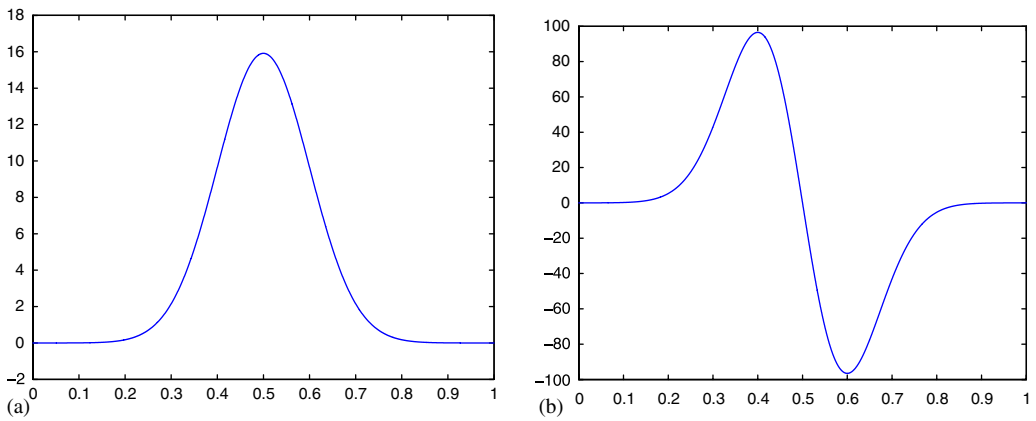

Figure 5. One-dimensional Gaussian hill: (a) solution profile; and (b) exact gradient.

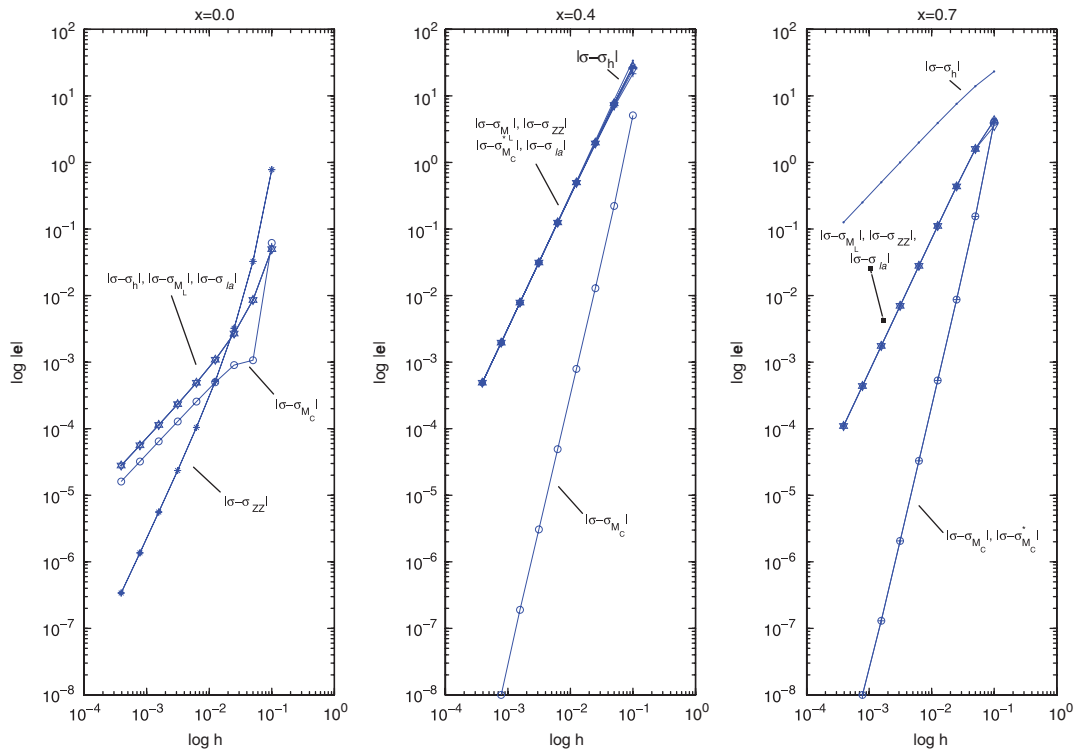

Figure 6. Gaussian hill: Nodal rate of convergence of σ_h, $\hat{\sigma}$ and $\hat{\sigma}^*$.

Copyright © 2006 John Wiley & Sons, Ltd.

Table I. Compression corner: comparison of error indicators for $\eta_{\text{ref}} = 1\%$, $\eta_{\text{crs}} = 0.1\%$.

	Grid1	Grid2	Grid3	Grid4	Grid5
$\hat{\sigma}_{ZZ}$	1612	1877	2869	5000	9329
$\hat{\sigma}^*_{ZZ}$	1612	1830	2743	4815	8963
$\hat{\sigma}_{M_L}$	1612	1874	2831	4950	9242
$\hat{\sigma}^*_{M_L}$	1612	1827	2738	4781	8888
$\hat{\sigma}_{M_C}$	1612	1699	2448	4238	7918
$\hat{\sigma}^*_{M_C}$	1612	1705	2451	4221	7783

limited averaging techniques constitute a useful tool for cost effective gradient reconstruction procedures which may be extended to the reconstruction of higher order derivatives.

5.2. 15° *Converging channel*

Let us proceed to the numerical treatment of the compressible Euler equations and employ the presented error indicator to govern an adaptive mesh refinement/coarsening procedure. As a first benchmark we consider a supersonic flow through a two-dimensional channel. The right half of the bottom wall is sloped at 15° giving rise to the formation of an oblique shock. For $M_\infty = 2.5$ the inclination angle $\beta = 36.94°$ and the downstream Mach number $M = 1.87$ can be easily computed as explained in any textbook on oblique shock theory. A detailed description of this so-called compression corner benchmark including numerical solutions computed by the *Wind-US* code is available in the *CFD Verification and Validation Database* of the NPARC Alliance [54].

The initial coarse grid1 of 1612 linear triangles is presented in Figure 7(a). The grid refinement/coarsening procedure has been called each time the relative changes of the 'halfway' converged solution reached the square root of $\varepsilon = 10^{-7}$. The adapted grids resulting from 4 iteration cycles are depicted in the diagrams (b)–(e). Here, the edgewise slope-limited gradient values $\hat{\sigma}^*_{M_C}$ have been employed to steer the adaptation process with $\eta_{\text{ref}} = 1\%$ and $\eta_{\text{crs}} = 0.1\%$. The resulting Mach number distribution computed on the final grid5 is shown in Figure 7(e). The thin shock wave is captured with an impressive accuracy and the inclination angle perfectly matches the theoretical value. The adapted numerical solution is even superior to the one presented in Reference [9] which was computed on a boundary-fitted uniform mesh of 128×128 bilinear elements by the less diffusive FEM–FCT algorithm making use of a smaller time step.

The regular triangulations resulting from other error indicators 'look' quite similar to the ones presented above and, hence, are not shown here. However, some difference in terms of triangles can be observed from Table I. The number of finite elements that constitute the finest grid for $\hat{\sigma}_{ZZ}$ exceeds that for $\hat{\sigma}^*_{M_C}$ by as much as 20%. This moderate improvement may be attributed to the very simple structure of the considered benchmark.

5.3. 5° *Converging channel*

Our next example is taken from Reference [39] and deals with multiple shock reflections. A supersonic flow at $M_\infty = 2$ enters a converging channel with the bottom wall sloped at 5° from the inlet. The initial triangulation was generated from a uniform mesh of 60×16

Copyright © 2006 John Wiley & Sons, Ltd.

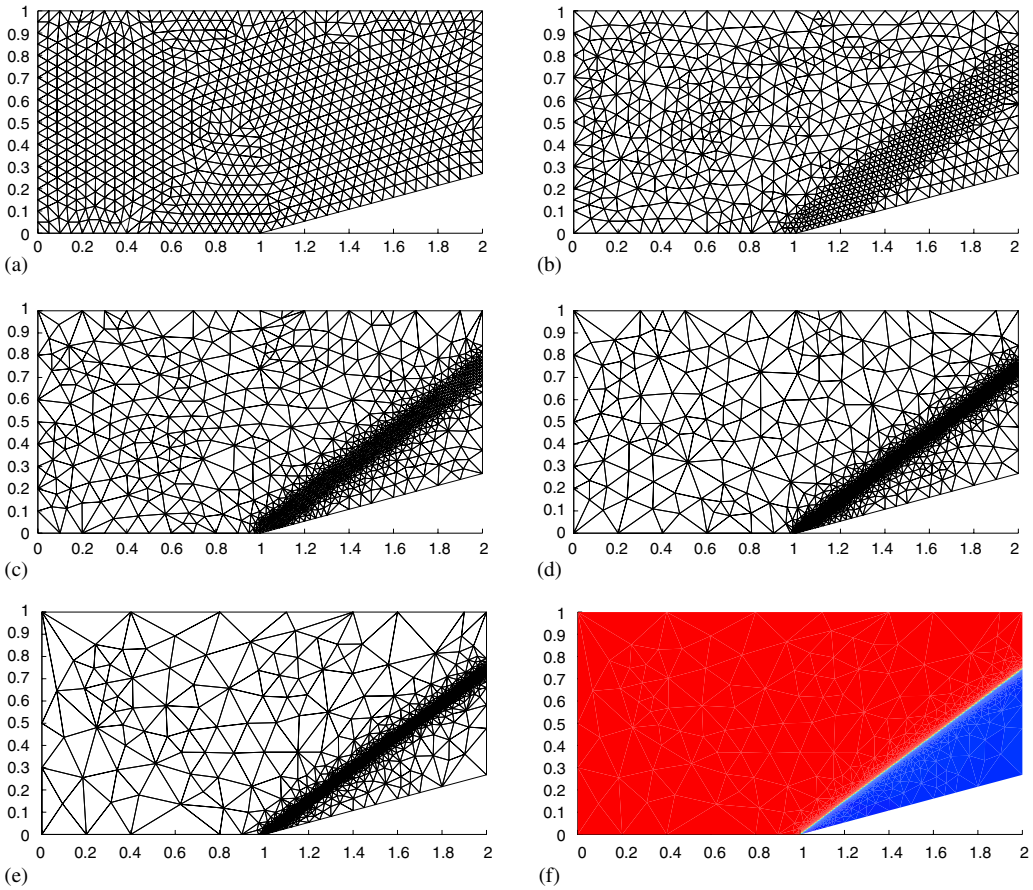

Figure 7. 15° Compression corner at $M_\infty = 2.5$: (a) coarse grid1; (b) grid2; (c) grid3; (d) grid4; (e) grid5; and (f) Mach number.

quadrilaterals by dividing each element into two triangles. Figure 8 shows the coarse grid as well as a sequence of three refinement/coarsening steps. Note how the finest region confines itself more and more to the vicinity of the shock as the adaptation proceeds. At the same time, a multiply reflected shock wave confines five zones of essentially uniform flow in which the mesh becomes increasingly coarsened. The normalized density distribution computed on the finest grid demonstrates the precise separation of five uniform zones as depicted in Figure 8(e). The agreement of density and Mach number for the exact and the numerical solution presented in Table II is amazing.

The crisp resolution of the reflected shock wave can also be realized from the density values on a slice through the grid presented in Figure 9. For unstructured meshes, a straight line along $y = 0.6$ is quite unlikely to match any of the grid points. In order to draw a fair comparison between different levels of refinement, the coordinates of all intersection points of edges and the prescribed cutline ($y = 0.6$) have been computed. In a second step, the density

Copyright © 2006 John Wiley & Sons, Ltd.

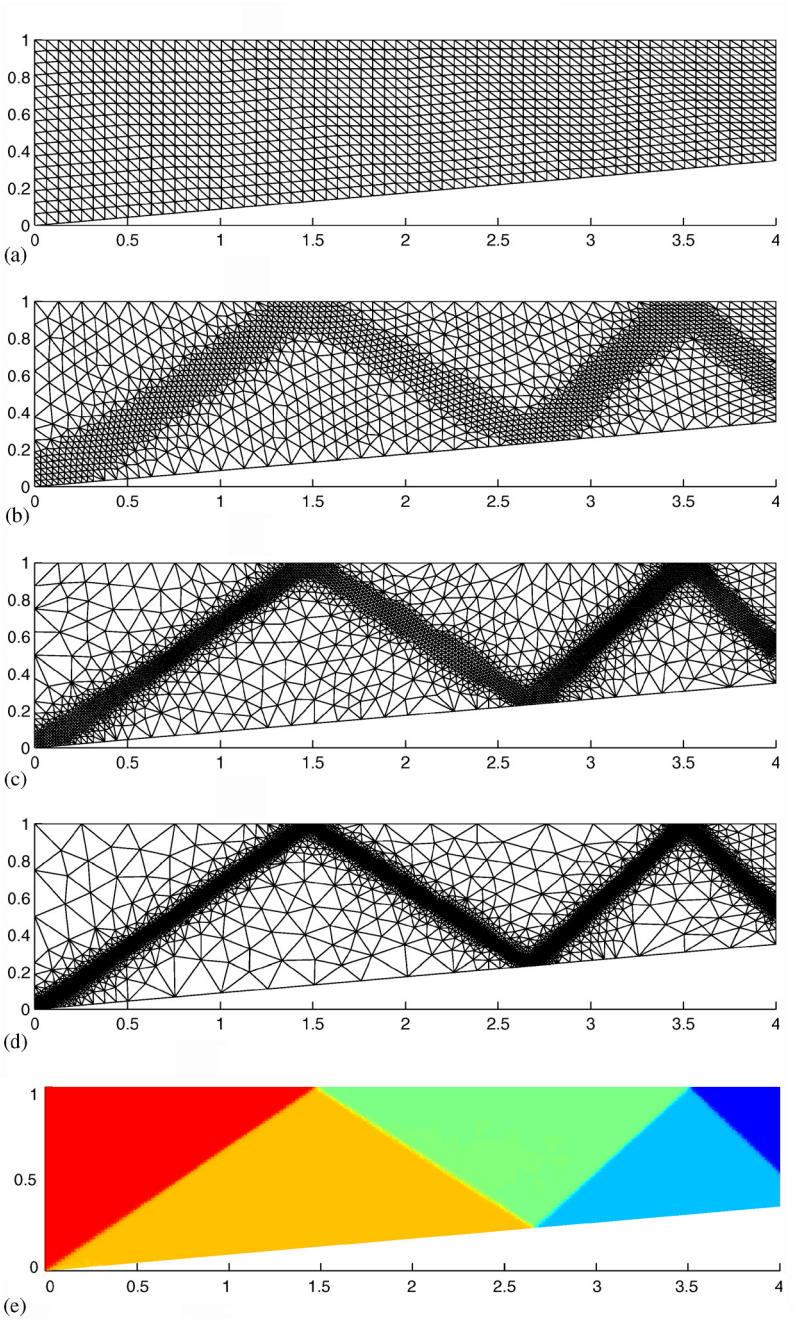

Figure 8. 5° Converging channel at $M_\infty = 2$: (a) coarse grid, 2048 cells; adapted grids: (b) 3503 cells; (c) 7194 cells; finest grid: (d) 15 664 cells; and (e) density distribution.

Copyright © 2006 John Wiley & Sons, Ltd.

Table II. Converging channel: solution values.

	Exact	Computed
ρ_{I}	1.000	1.000
ρ_{II}	1.216	1.216
ρ_{III}	1.463	1.462
ρ_{IV}	1.747	1.747
ρ_{V}	2.081	2.079
M_{I}	2.000	2.000
M_{II}	1.821	1.821
M_{III}	1.649	1.651
M_{IV}	1.478	1.479
M_{V}	1.302	1.304

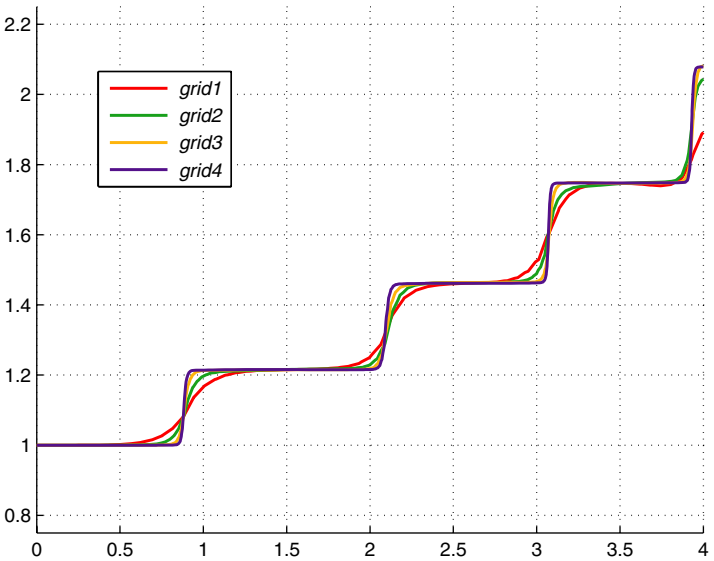

Figure 9. Density cutlines at $y = 0.6$.

values on the slice have been recovered by means of linear interpolation from adjacent nodes. No additional error has been introduced during the visualization procedure, since $\mathscr{P}_1$-finite elements have been employed to approximate the solution which, consequently, varies linearly along edges.

It can be clearly seen that the correct solution values in the interior are already obtained on the coarsest grid. However, artificial diffusion passing through the flux limiter smears the shock wave across several elements and yields underpredicted density values at the outflow. Both the steepness of the 'cascade' and the correctness of the boundary values get greatly improved as the adaptation process continues.

Copyright © 2006 John Wiley & Sons, Ltd.

6. CONCLUSIONS AND OUTLOOK

In this paper we focused on the reliability of gradient recovery procedures applied to discontinuous solutions. The local L_2-error of the solution gradient has been decomposed into a sum of edge contributions. Standard variational and discrete recovery techniques have been revisited within an edge-based formulation and their applicability to solutions with steep gradients has been questioned. Geometric constraints to be imposed on the high-order slopes have been derived. In particular, the values of the consistent low-order finite element gradient provided natural upper and lower bounds. A slope limiter was invoked edge-by-edge so as to correct the provisional gradient values resulting from linear interpolation of nodal data. Limited averaging procedures inspired by high-resolution finite volume schemes were presented as a promising alternative. They were utilized to compute smoothed gradient values at the midpoints of edges directly from the adjacent consistent slopes. Moreover, the treatment of boundary nodes/edges was addressed.

The performance of the new error indicators was demonstrated in one and two space dimensions. Algebraic flux correction schemes [8] have been successfully equipped with adaptivity. The highly unstructured grids resulting from local refinement call for the use of fully implicit AFC methods which are unconditionally stable/positivity-preserving. However, it is rather difficult to march the resulting nonlinear system of equations to steady state on strongly nonuniform meshes. Full multigrid (FMG) has been employed to compute the steady state solution on the initial mesh. It could be worthwhile to employ a full approximation scheme (FAS) to tackle the strong nonlinearity. In addition, the (nonlinear) TVD operator can be constructed explicitly and used as a better preconditioner for the defect correction procedure so as to improve the nonlinear rate of convergence.

Sophisticated mesh optimization techniques tailored to the peculiarities of algebraic flux correction schemes will be considered in forthcoming publications. An algebraic approach to the design of grid refinement/coarsening strategies and mesh smoothing/optimization algorithms constitutes an interesting direction for further research.

ACKNOWLEDGEMENTS

The authors would like to thank Stefan Turek for fruitful discussions.

REFERENCES

1. Babuška I, Rheinboldt W. *A posteriori* error estimates for the finite element method. *International Journal for Numerical Methods in Engineering* 1978; **12**:1597–1615.
2. Ainsworth M, Oden JT. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley: New York, 2000.
3. Babuška I, Szabo B. *Finite Element Analysis*. Wiley: New York, 1991.
4. Verfürth R. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Wiley/Teubner: New York/Stuttgart, 1996.
5. Kuzmin D, Turek S. Flux correction tools for finite elements. *Journal of Computational Physics* 2002; **175**: 525–558.
6. Kuzmin D, Möller M, Turek S. High-resolution FEM–FCT schemes for multidimensional conservation laws. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:4915–4946.
7. Kuzmin D, Turek S. High-resolution FEM–TVD schemes based on a fully multidimensional flux limiter. *Journal of Computational Physics* 2004; **198**:131–158.
8. Kuzmin D, Möller M. Algebraic flux correction I. Scalar conservation laws. In *Flux-Corrected Transport*: *Principles*, *Algorithms*, *and Applications*, Kuzmin D, Löhner R, Turek S (eds). Springer: Berlin, 2005; 155–206.

Copyright © 2006 John Wiley & Sons, Ltd.

9. Kuzmin D, Möller M. Algebraic flux correction II. Compressible Euler equations. In *Flux-Corrected Transport*: *Principles*, *Algorithms*, *and Applications*, Kuzmin D, Löhner R, Turek S (eds). Springer: Berlin, 2005; 207–250.
10. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984; **53**:484–512.
11. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* 1989; **82**:64–84.
12. Berger MJ, LeVeque R. Adaptive mesh refinement using wave propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis* 1998; **35**:1439–1461.
13. Roache PJ. Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering* 1994; **116**:405–413.
14. Roy CJ. Grid convergence error analysis for mixed-order numerical schemes. *AIAA Journal* 2003; **41**:595–604.
15. Geppert J. Adaptive Gitterverfeinerung bei Flux-Limiter-Verfahren. *Ph.D. Thesis*, University of Hamburg, 1996.
16. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering* 1987; **24**:337–357.
17. Ainsworth M, Zhu JZ, Craig AW, Zienkiewicz OC. Analysis of the Zienkiewicz–Zhu *a posteriori* error estimator in the finite element method. *International Journal for Numerical Methods in Engineering* 1989; **28**:2161–2174.
18. Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery and *a posteriori* error estimates. Part 1: the recovery techniques. *International Journal for Numerical Methods in Engineering* 1992; **33**:1331–1364.
19. Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery and *a posteriori* error estimates. Part 2: error estimates and adaptivity. *International Journal for Numerical Methods in Engineering* 1992; **33**:1365–1382.
20. Prudhomme S, Oden JT. Computable error estimators. In: *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, Barth TJ, Deconinck H (eds). Springer: Berlin, 2003.
21. Prudhomme S. Adaptive finite element simulation of a supersonic underexpanded jet. *Master Thesis*, University of Virginia, 1992.
22. Cockburn B, Shu CW. TVD Runge–Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: general framework. *Mathematics of Computation* 1989; **52**:411–435.
23. Jameson A. Analysis and design of numerical schemes for gas dynamics 1. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *International Journal of Computational Fluid Dynamics* 1995; **4**:171–218.
24. Oden JT, Brauchli HJ. On the calculation of consistent stress distribution in finite element applications. *International Journal for Numerical Methods in Engineering* 1971; **3**:317–325.
25. Oden JT, Reddy JN. Note on an approximate method for computing consistent conjugate stresses in elastic finite elements. *International Journal for Numerical Methods in Engineering* 1973; **6**:55–61.
26. Hinton E, Campbell J. Local and global smoothing of discontinuous finite element functions using a least squares method. *International Journal for Numerical Methods in Engineering* 1974; **8**:461–480.
27. Cantin G, Loubignac C, Touzot C. An iterative scheme to build continuous stress and displacement solutions. *International Journal for Numerical Methods in Engineering* 1978; **12**:1493–1506.
28. Lyra PRM. Unstructured grid adaptive algorithms for fluid dynamics and heat conduction. *Ph.D. Thesis*, University of Wales, Swansea, 1994.
29. Zhang Z, Zhu JZ. Analysis of the superconvergent patch recovery technique and *a posteriori* error estimator in the finite element method (I). *Computer Methods in Applied Mechanics and Engineering* 1995; **123**:173–187.
30. Zhang Z, Zhu JZ. Analysis of the superconvergent patch recovery technique and *a posteriori* error estimator in the finite element method (II). *Computer Methods in Applied Mechanics and Engineering* 1998; **163**:159–170.
31. Zhang Z. Ultraconvergence of the patch recovery technique II. *Mathematics of Computation* 2000; **69**:141–158.
32. Naga A, Zhang Z. *A posteriori* error estimates based on polynomial preserving recovery. *SIAM Journal on Numerical Analysis* 2004; **42**:1780–1800.
33. Zhang Z, Naga A. A new finite element gradient recovery method: superconvergence property. *SIAM Journal on Scientific Computing* 2005; **26**:1192–1213.
34. *GiD—The personal pre and postprocessor*. `http://gid.cimne.u0pc.es/`
35. Brandt A. Multilevel adaptive computations in fluid dynamics. *AIAA Journal* 1980; **18**:1165–1172.
36. Hackbusch W. *Iterative Solution of Large Sparse Systems of Equations*. Applied Mathematical Sciences, vol. 95. Springer: Berlin, 1994.
37. Brandt A. Multi-level adaptive solutions to boundary value problems. *Mathematics of Computation* 1977; **31**:333–390.
38. Feistauer M, Felcman J, Straškraba I. *Mathematical and Computational Methods for Compressible Flow*. Oxford University Press: Oxford, 2003.
39. Shapiro RA. *Adaptive Finite Element Solution Algorithm for the Euler Equations*. Notes on Numerical Fluid Mechanics, vol. 32. Vieweg: Braunschweig, 1991.
40. Hansbo P. A free-Lagrange finite element method using space-time elements. *Computer Methods in Applied Mechanics and Engineering* 2000; **188**:347–361.

Copyright © 2006 John Wiley & Sons, Ltd.

41. Bank RE, Sherman AH, Weiser A. Refinement algorithms and data structures for regular local mesh refinement. *Scientific Computing* (*Applications of Mathematics and Computing to the Physical Sciences*). North-Holland: Amsterdam, 1983; 3–17.
42. Babuška I, Rheinboldt WA. Reliable error estimation and mesh adaptation for the finite element method. In *Computational Methods in Nonlinear Mechanics*. North-Holland: Amsterdam, 1980; 67–108.
43. Rivara MC. Design and data structure of fully adaptive multigrid finite element software. *ACM Transactions on Mathematical Software* 1984; **10**:242–264.
44. Rivara MC. New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations. *Proceedings of the 5th International Meshing Roundtable 96*, Pittsburgh, 1996; 77–86.
45. Rivara MC. Using longest-side bisection techniques for the automatic refinement of delaunay triangulations. *International Journal for Numerical Methods in Engineering* 1997; **40**:581–597.
46. De Cougny HL, Shephard MS. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering* 1999; **46**:1101–1125.
47. Kobbelt L, Schroder P. A multiresolution framework for variational subdivision. *Technical Report CS-TR-97-05*, Computer Graphics Research Group, Computer Science Department, California Institute of Technology, 1997.
48. Walter MAT, Abdu AAQ, da Silva LFF, Azevedo JLF. Evaluation of adaptive mesh refinement and coarsening for the computation of compressible flows on unstructured meshes. *International Journal for Numerical Methods in Fluids* 2005; **49**:999–1014.
49. Bank RE, Smith RK. Mesh smoothing using *a posteriori* error estimates. *SIAM Journal on Numerical Analysis* 1997; **34**:979–997.
50. Forsyth PA. A control volume finite element approach to NAPL groundwater contamination. *SIAM Journal on Scientific and Statistical Computing* 1991; **12**:1029–1057.
51. Freitag LA. On combining Laplacian and optimization-based mesh smoothing techniques. Trends in unstructured mesh generation. *ASME Applied Mechanics Division* 1997; **220**:37–44.
52. Peraire J, Vahdati M, Morgan K, Zienkiewicz OC. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics* 1987; **72**:449–466.
53. Oran ES, Boris JP. *Numerical Simulation of Reactive Flow*. Cambridge University Press: Cambridge, MA, 2001.
54. *NPARC Alliance*. Computational fluid dynamics (CFD) verification and validation web site: `http://www.grc.nasa.gov/WWW/wind/valid/`

Copyright © 2006 John Wiley & Sons, Ltd.