

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*

Abstract

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 \mathbb{P}_1 -finite element solution into a sum of edge contributions. The gradient values at edge midpoints are interpolated from the two adjacent vertices. Traditional techniques to recover a (superconvergent) nodal gradient from the consistent finite element gradients are reviewed. The deficiencies of standard smoothing procedures – global L_2 -projection and the Zienkiewicz-Zhu patch recovery – as applied to non-smooth 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.

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. At the same time, this complexity has made it increasingly difficult to guarantee the reliability of the numerical solution. The recent trend for *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 [4] in the late seventies of the last century, theories and methods of *a posteriori* error estimation have been developed extensively [2],[3],[35]. 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.

*Correspondence to: matthias.moeller@math.uni-dortmund.de

In a series of recent publications [19], [20], [21], [22], [23], we have developed a family of high-resolution schemes subsumed under the *algebraic flux correction* (AFC) paradigm. 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 sufficiently 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 the incorporation of an adaptive mesh (de)refinement procedure in order to increase the capability of resolving even small scale features. 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.

For the adaptive treatment of hyperbolic problems, Berger *et. al.* [8], [9] employ Richardson extrapolation to estimate the truncation error in the solution and perform local grid refinement so as to evenly distribute the error. This technique requires the *a priori* knowledge of the order of approximation which varies locally for discretizations based on flux/slope limiters and thus is not applicable in the context of AFC.

In [14], both the smoothness sensor and the correction factors resulting from the limiter function have been utilized to steer grid adaptivity. 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 in addition it vanishes for (nearly) zero flow velocity.

Recovery-based error estimators were first suggested by Zienkiewicz and Zhu [42], 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' [1], these jumps in the gradient serve as an indicator for errors in the numerical solution. Several methods for recovering piecewise continuous gradients have been proposed in the literature. Some of them, including the well-known Zienkiewicz-Zhu patch recovery technique [44], [45], rely on the superconvergence property of the finite element method obtained 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 [30] 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' [29]. 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 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 i th spatial direction one after another. Then the task of finding a good approximation to the convective term reduces to that of computing the i th 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) [19], [20], 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 Finite element discretization

As a model problem, consider the time-dependent continuity equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0, \quad \text{in } \Omega. \quad (1)$$

Its variational form is derived by first multiplying the governing equation by the weighting function w and integrating over the computational domain Ω

$$\int_{\Omega} w \left[\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) \right] d\mathbf{x} = 0. \quad (2)$$

A common practice in finite element methods for conservation laws is to interpolate the convective fluxes in the same way as the numerical solution

$$u_h = \sum_j u_j \varphi_j, \quad (\mathbf{v}u)_h = \sum_j (\mathbf{v}_j u_j) \varphi_j, \quad (3)$$

where φ_j denotes the basis function spanning the finite-dimensional subspace. This kind of approximation was promoted by Fletcher [13] who called it the *group finite element formulation*. The substitution of (3) into (2) yields a DAE system for the vector of nodal values which can be written compactly in matrix notation

$$M_C \frac{du}{dt} = K u, \quad (4)$$

where $M_C = \{m_{ij}\}$ denotes the consistent mass matrix and $K = \{k_{ij}\}$ stands for the discrete transport operator. The corresponding matrix entries are given by

$$m_{ij} = \int_{\Omega} \varphi_i \varphi_j d\mathbf{x}, \quad k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij}, \quad \mathbf{c}_{ij} = \int_{\Omega} \varphi_i \nabla \varphi_j d\mathbf{x}. \quad (5)$$

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 mesh refinement/coarsening has been performed. Particularly for steady state problems solved by pseudo time-stepping, the grid remains unchanged for a long period of time such that the matrix K can be updated efficiently by computing its entries k_{ij} from formulae (5) without resorting to costly numerical integration in each step. Moreover, the coefficients m_{ij} and \mathbf{c}_{ij} will be needed in the gradient recovery procedure.

3 A posteriori error indicators

No matter how sophisticated high-resolution finite element schemes employed to solve equation (1) are, approximations involve all sorts of numerical errors, integration errors, round-off errors, implementation errors (!), algorithmic error, discretization errors, etc., to name just a few. In what follows, we shall concentrate on errors due to the finite element discretization of the spatial derivatives. Since we are mainly interested in steady state solutions it is acceptable to ignore those resulting from the discretization in time.

The numerical error relates the exact solution \mathbf{u} of the continuous problem (1) to the nodal values u_h of the finite element approximation satisfying equation (4)

$$e = \mathbf{u} - u_h. \quad (6)$$

It is well known, that obtaining an approximation to e yields a problem as complex as the one for \mathbf{u} . Thus the main objective of *a-posteriori* error estimation is not to get an approximation of equation (6), but to estimate the magnitude of the error.

The first step in a *posteriori* error estimation is to choose a suitable norm in which the accuracy of the finite element approximation should be measured. Different norms show different aspects of the error, and for convection-dominated problems, the choice of an appropriate norm is still an open question. Let

$$\mathbf{e}_\sigma = \sigma - \sigma_h \quad (7)$$

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

$$\sigma_h = \nabla u_h = \sum_j u_j \nabla \varphi_j. \quad (8)$$

In what follows, we will refer to σ_h as the low-order gradient. In general, pointwise error estimates are difficult to obtain, so integral measures are typically employed in the finite element framework. A widely used representative of such measures is the standard L_2 -norm which can be associated with either the error of the solution

$$\|e\|_{L_2} = \left(\int_{\Omega} e^T e \, d\mathbf{x} \right)^{1/2} \quad (9)$$

or in terms of the gradient error by replacing e by \mathbf{e}_σ in the equation above. Although the integral measure (9) is defined in the whole domain Ω , its square can be obtained by summing all element contributions over the triangulation \mathcal{T}_h of Ω . Thus

$$\|e\|_{L_2}^2 = \sum_{T \in \mathcal{T}_h} \|e\|_{L_2(T)}^2, \quad (10)$$

where subscript $L_2(T)$ refers to the local L_2 -norm computed on element $T \in \mathcal{T}_h$. Since we employ piecewise linear 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. The aim of recovery-based estimators, originally introduced by Zienkiewicz and Zhu in [42], is to replace the exact value σ , which in general is not known, by a smoothed gradient field $\hat{\sigma}$ (to be defined below), such that

$$\mathbf{e}_\sigma \approx \hat{\mathbf{e}}_\sigma = \hat{\sigma} - \sigma_h \quad (11)$$

gives a good approximation to the exact error defined in (7). Note that in our case the computation of the L_2 -norm of the gradient error requires the numerical integration of a piecewise quadratic function. Thus the use of a quadrature rule which is exact for \mathbb{P}_2 -functions is mandatory. We employ the following Newton-Cotes quadrature rule to compute the element contribution of the triangle $T \in \mathcal{T}_h$ to the global gradient error

$$\int_T \hat{\mathbf{e}}_\sigma^T \hat{\mathbf{e}}_\sigma \, d\mathbf{x} = \frac{|T|}{3} \sum_{d=1}^2 \sum_{j \neq i} [\hat{\mathbf{e}}_\sigma^d(\mathbf{x}_{ij})]^2, \quad (12)$$

where superscript d denotes the d th component of the multidimensional vector $\hat{\mathbf{e}}_\sigma$ and $\mathbf{x}_{ij} := \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$. Let us single out the contribution of the edge $\vec{i}j$ to equation (12)

$$[\hat{\mathbf{e}}_\sigma^d(\mathbf{x}_{ij})]^2 = [(\hat{\sigma}^d - \sigma_h^d)(\mathbf{x}_{ij})]^2. \quad (13)$$

Here, σ_h is constant on the whole element and $\hat{\sigma}$ varies linearly along the edge. Note that equation (13) can be expanded to the following form

$$[\hat{\mathbf{e}}_\sigma^d(\mathbf{x}_{ij})]^2 = [\hat{\sigma}^d(\mathbf{x}_{ij})]^2 - 2\hat{\sigma}^d(\mathbf{x}_{ij})\sigma_T^d + [\sigma_T^d]^2, \quad (14)$$

where $\sigma_T^d := \sigma_h^d|_T$ stands for the d th component of the vector-valued constant gradient on element T . It remains to compute a smoothed gradient value $\hat{\sigma}^d$ at the midpoint of edge $\vec{i}j$ as will be explained in the subsequent section. Unless indicated otherwise, all modifications are to be performed individually for each spatial dimension so that superscript d can be dropped for convenience.

4 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 [10]. 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, geometrical 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 $\vec{i}j$.

As an alternative, a provisional gradient at the midpoint of edge $\vec{i\hat{j}}$ 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 set up by the first-order slopes of the two adjacent cells. This can be rectified by applying a slope limiter edge-by-edge so as to satisfy geometric constraints.

4.1 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 \bar{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

$$\tilde{u}_h(x) = \bar{u}_j + (x - x_j)\sigma_j \quad \forall x \in I_j, \quad (15)$$

where σ_j denotes an approximation of the solution gradient on the j th cell. In the simplest case, one-sided or centered slopes have been employed to obtain first- and second-order accurate schemes, respectively. However, oscillations are quite likely to appear in the second case. For a numerical scheme to be nonoscillatory, it should possess certain properties, e.g., be monotone or total variation / local extremum diminishing. This can be accomplished by employing *limited* slopes σ_j in equation (15).

For the construction of LED and TVD schemes, Jameson utilized limited average operators $\mathcal{L}(a, b)$ which are characterized by the following properties [18]:

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

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 form. Let the modified sign function be given by

$$\mathcal{S}(a, b) = \frac{\text{sign}(a) + \text{sign}(b)}{2} \quad (16)$$

which equals zero for $ab \leq 0$ 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:	$\mathcal{L}(a, b) = \mathcal{S}(a, b) \min\{ a , b \}$
2. Van Leer:	$\mathcal{L}(a, b) = \mathcal{S}(a, b) \frac{2 a b }{ a + b }$
3. MC:	$\mathcal{L}(a, b) = \mathcal{S}(a, b) \min\left\{\frac{ a+b }{2}, 2 a , 2 b \right\}$
4. superbee:	$\mathcal{L}(a, b) = \mathcal{S}(a, b) \max\{\min\{2 a , b \}, \min\{ a , 2 b \}\}$

In light of the above, equation (15) can be turned into a high-resolution scheme by setting

$$\sigma_j := \mathcal{L}\left(\frac{\bar{u}_j - \bar{u}_{j-1}}{\Delta_j}, \frac{\bar{u}_{j+1} - \bar{u}_j}{\Delta_j}\right), \quad (17)$$

where the cellwidth of the j th 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{i}_j so as to estimate its contribution (14) to the local error. Let σ_{ij}^+ and σ_{ij}^- denote the piecewise constant finite element values evaluated on the two elements to the left and to the right of edge \vec{i}_j , respectively. Then, the auxiliary quantities

$$\sigma_{ij}^{\min} = \max_{\min} \{\sigma_{ij}^+, \sigma_{ij}^-\} \quad (18)$$

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

$$\hat{\sigma}_{ij} = \mathcal{L}(\sigma_{ij}^{\min}, \sigma_{ij}^{\max}). \quad (19)$$

Unfortunately, the choice of the limiter function is far from being unique and some kind of empiricism is unavoidable. This has led us to the derivation of an alternative approach whereby a slope limiter is applied to the provisional value of the averaged edge gradient. Established recovery techniques are employed to generate smoothed slope values at the vertices. Due to the linearity of edge gradients, their values at the midpoints can simply be interpolated from adjacent nodes. Finally, slope limiting is carried out so as to guarantee that the corrected quantity $\hat{\sigma}_{ij}^*$ stays within the upper/lower bounds defined in (18).

4.2 Nodal gradient recovery

Since the advent of recovery-based schemes introduced by Zienkiewicz and Zhu [42], various methods have been proposed in the literature to construct an ‘improved’ gradient from the finite element solution by means of *averaging projection* schemes

$$\hat{\sigma} = \sum_j \hat{\sigma}_j \phi_j, \quad (20)$$

where the coefficients are obtained by asking that

$$\int_{\Omega} \phi_i (\hat{\sigma} - \sigma_h) \, d\mathbf{x} = 0, \quad \forall i. \quad (21)$$

Here, ϕ_i stands for the basis functions onto which σ_h is to be projected. Note that the corresponding element shape functions may differ from the ones of the finite element approximation (3) in their polynomial degree. A detailed analysis of projection based error estimators can be found in a paper by Ainsworth *et. al.* [1]. They finally conclude that the original recovery procedure introduced in [42], which corresponds to choosing $\phi = \varphi$ in the equations above, ‘is not only effective, but also the most economical’ one.

This has already been mentioned in an earlier publication by Oden and Brauchli [26]. The substitution of equation (20) into (21) yields the following discrete problem

$$\sum_j \left[\int_{\Omega} \varphi_i \varphi_j \, d\mathbf{x} \right] \hat{\sigma}_j - \sum_j \left[\int_{\Omega} \varphi_i \nabla \varphi_j \, d\mathbf{x} \right] u_j = 0, \quad \forall i. \quad (22)$$

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

$$M_C \hat{\sigma} = \mathbf{q}, \quad (23)$$

where $M_C = \{m_{ij}\}$ denotes the consistent mass matrix and the multicomponent right-hand side is given by $\mathbf{q} = \mathbf{C}u$. The matrix $\mathbf{C} = \{\mathbf{c}_{ij}\}$ is assembled from the auxiliary coefficients defined in (5) which correspond to the discretized space derivatives and have zero row sum, i.e., $\sum_j \mathbf{c}_{ij} = 0$ as long as the sum of the basis functions φ_j is equal to one at every point. Hence, an edge-by-edge assembly of $\mathbf{q}_i = \sum_{j \neq i} \mathbf{c}_{ij}(u_j - u_i)$ is feasible.

Note that the algebraic system (23) 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 (20)–(21) are called *variational recovery* [24] and can be applied repeatedly so as to determine an approximation to a higher-order derivative. The solution to the algebraic system (23) can be computed iteratively by successive approximation preconditioned by the lumped mass matrix $M_L = \{m_i\}$, where $m_i = \sum_j m_{ij}$

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

If mass lumping is applied directly to equation (23), this yields an explicit formula for computing the recovered gradient at each node as follows

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

Over the years, a more accurate patch recovery technique was introduced by Zienkiewicz and Zhu [44], [45], 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

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

where the vector $\mathbf{p}(\mathbf{x}) = [1, x, y, x^2, \dots, x^k, x^{k-1}y, \dots, xy^{k-1}, y^k]$ contains the appropriate monomials of degree k at most. Since each vertex is surrounded by a patch of elements sharing this node, the vector of coefficients $\mathbf{a} = [a_1, a_2, \dots, a_m]^T$ with $m = (k+1)(k+2)/2$ can be computed from a discrete least square fit to the set of sampling points \mathbf{x}_j [44]. From that it follows, that \mathbf{a} is the solution to the linear system

$$\mathbf{M}\mathbf{a} = \mathbf{b}, \quad (27)$$

where the local matrix \mathbf{M} and the right-hand side vector \mathbf{b} are given by

$$\mathbf{M} = \sum_j \mathbf{p}(\mathbf{x}_j)^T \mathbf{p}(\mathbf{x}_j), \quad \mathbf{b} = \sum_j \mathbf{p}^T(\mathbf{x}_j) \sigma_h(\mathbf{x}_j). \quad (28)$$

For linear triangles, $\mathbf{a} = [a_1, a_2, a_3]^T$, $\mathbf{p} = [1, x, y]$ 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 $\text{rank } \mathbf{M} = m$.

Since the advent of the *superconvergent patch recovery* (SPR) technique [44] its super- and even ultraconvergence property has been analyzed extensively in the literature [36], [37], [38]. This paved the way to the development of so-called *polynomial preserving* (PPR) gradient recovery schemes [39]. While in SPR methods a \mathbb{P}_k -polynomial is best fitted to ∇u_h directly, PPR schemes compute the nodal quantity $p \in \mathbb{P}_{k+1}$ as a polynomial approximation to u_h and apply the derivative operator afterwards ($\nabla u_h \approx \nabla p$). 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 [41] in which the idea of element patches is abandoned in favor 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 (see below) 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 non-physical 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.

4.3 Edgewise slope limiting

No matter if patch recovery or 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 (12) we have to sum the contributions of the adjacent edges (13) which (in the interior) can be associated with exactly *two* triangles sharing this edge. Hence, the auxiliary quantities defined in (18) constitute excellent upper/lower bounds for the final edge gradient.

In the first step, provisional edge gradient values are recovered at the midpoint of edges which are always 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., $\hat{\sigma}_{ij} = \frac{1}{2}(\hat{\sigma}_i + \hat{\sigma}_j)$. Alternatively, patch recovery can be used to obtain the midpoint gradient values directly. In the next step, the upper/lower bounds (18) are imposed so that the corrected edge gradient is given by

$$\hat{\sigma}_{ij}^* = \max\{\sigma_{ij}^{\min}, \min\{\hat{\sigma}_{ij}, \sigma_{ij}^{\max}\}\}. \quad (29)$$

The edgewise slope-limiting procedure is illustrated in Figure 1 for an interior edge.

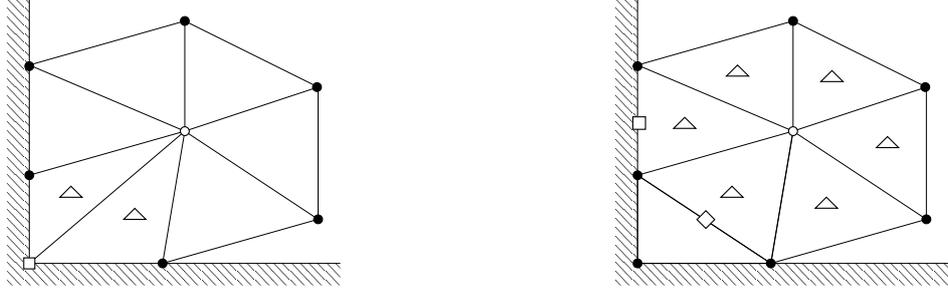


Figure 2: Boundary treatment: nodal *vs.* edgewise recovery.

5 Adaptation strategy

In adaptive solution procedures one typically starts with an initial grid coarse enough for the solution to be inexpensive to compute. Yet it needs to be fine enough so as to capture the essential flow features. As pointed out in the introduction, the misfortune experienced with applying the ZZ error estimator to an adaptive shock wave simulation [29] 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.

As a first step in adaptive solution procedures for steady state flow simulations, a provisional solution is computed on the initial mesh. The relative error of the density has been employed as a criterion to measure convergence [12]. Following [33], the flow solver is stopped if this criterion is satisfied by 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 final solution is only necessary once the finest grid has been generated. To this end, one or more mesh ‘convergence’ criteria need to be defined. For steady state problems one typically prescribes the maximum number of refinement levels. As a consequence, one needs to keep track of the complete mesh hierarchy so as to control the father-son relationship and the number of times an element has been refined. If both mesh refinement and coarsening take place, alternative stopping criteria making use of minimum element size and/or element shape conditions need to be considered.

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 final solution and the gradient, respectively

$$\eta := \frac{\|\mathbf{e}_\sigma\|_{L_2}}{\|\nabla u\|_{L_2}} \leq \eta_{\text{tol}}. \quad (31)$$

Since neither the exact slope values nor the true error are known, the best approximation available is utilized instead. From (10) it follows that the global L_2 -norm can be decomposed into element contributions. Moreover, let us assume that the error is equally distributed between cells then the condition $\eta \leq \eta_{\text{tol}}$ (c.f. (31)) can be rewritten as follows

$$\|\hat{\mathbf{e}}_\sigma\|_{L_2(T)} \leq \eta_{\text{tol}} \left[(\|\sigma_h\|_{L_2}^2 + \|\hat{\mathbf{e}}_\sigma\|_{L_2}^2) / |\mathcal{T}_h| \right]^{1/2} =: e_{\text{tol}}, \quad (32)$$

where $|\mathcal{T}_h|$ denotes the number of elements employed in the current triangulation \mathcal{T}_h .

A similar estimate in terms of $\|e\|$ is formulated in [42]. Finally, the ratio

$$\xi_T = \|\hat{\mathbf{e}}_\sigma\|_{L_2(T)}/e_{\text{tol}} > 1 \quad (33)$$

specifies the set of elements T to be refined. At the same time, the cells to be coarsened can be determined by inverting all inequalities and replacing η_{tol} by some $\eta'_{\text{tol}} \ll \eta_{\text{tol}}$.

Grid refinement and coarsening techniques In a loop over elements, cells marked for refinement 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 *blue refinement* is applied to bisected edges that still remain, that is, their midpoints are connected to the opposite vertices. The regular refinement of triangles is extensively dealt with in [5] and the *PLTMG User's Guide* [7] including a detailed description of efficient data structures.

Another conforming mesh refinement algorithm is based on bisection of the longest-side as proposed by Rivara in [31]. In a loop over flagged elements, new vertices are inserted at the midpoints of the longest edges and connected to the opposite nodes. Adjacent triangles containing bisected edges are also marked and the refinement process continues until all hanging nodes have been eliminated. A summary of various *geometrical* properties proven for the longest-side bisection algorithm can be found in [32].

Preliminary results from simulation of shock waves indicate that the marriage of generalized edge bisection techniques and algebraic flux correction may lead to further improvements. Recall that our AFC methodology [22], [23] 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 smaller than that for another edge the latter one should be bisected. The same applies to an edge \vec{ij} related to a ‘strongly’ antidiffusive flux. Consider the situation where this flux into one node, say i , cannot be balanced by diffusive fluxes from neighboring nodes so that its magnitude needs to be drastically limited. In this case, edge \vec{ij} should be bisected, unless this would entail a reduction of the correction factor for node j . Obviously, this *algebraic* edge bisection approach may lead to highly anisotropic meshes which are tailored to the peculiarities of our AFC schemes [22].

Mesh coarsening is accomplished as described in [17]. In essence, edge-swapping is performed repeatedly so as to ‘isolate’ the vertex to be erased. Iteration continues until the corresponding node is connected to just three triangles and can be safely removed. Vertices to be deleted from the boundary are first ‘moved’ into the interior by introducing an artificial boundary element before the standard procedure can be applied.

Grid improvement techniques Edge-swapping can also be utilized as a postprocessing step so as to improve the mesh quality with respect to some geometric measures, e.g., the normalized shape regularity functional presented in [6]:

$$q(T) = 4\sqrt{3}|T| \left[\sum_{j \neq i} |\vec{ij}|^2 \right]^{-1}. \quad (34)$$

Here, $|\vec{ij}|$ 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 the standard Galerkin finite element approximation of the second-order diffusion operator is an M-matrix [15]. 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 physical diffusion operator (if any) remain nonnegative.

In addition, ‘smart’ Laplacian or optimization-based mesh smoothing [16] 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.

5.1 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 self-adaptive AFC schemes are as follows:

In the outer loop:

1. Generate the required adjacency lists for nodes and elements and initialize the edge-based data structure. Furthermore, compute the coefficient matrices (5).

In the solution loop:

2. Employ algebraic flux correction techniques to transform the linear high-order scheme (4) into its high-resolution counterpart of TVD type [20].
3. Use an iterative defect correction procedure to solve the resulting nonlinear algebraic system for the current time step.
4. Compute the relative error of some indicator variable, i.e., density, to check if the solution has ‘half-converged’. Otherwise, repeat steps 2–4.
5. Evaluate the constant gradient (8) and recover improved slope values for each edge either directly (19) or by means of edgewise slope limiting (29)/(30) applied to the average of smoothed nodal gradients (c.f. (20)–(21) or (26)–(28)).
6. Assemble the L_2 -norm of the element gradient error (12) from the edge contributions (14) and refine/coarsen all triangles according to conditions (32)–(33).
7. Apply edge-swapping and/or grid improving methods in order to increase the mesh quality with respect to *algebraic* quality measures.
8. If the final/‘converged’ grid has been constructed, repeat steps 2–4 until the fully converged solution is obtained. Otherwise, go to step 1.

6 Numerical examples

In order to demonstrate the behavior 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 [20]–[22]. 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. At the same time, a nonconservative projection scheme may be employed in order to transfer the old solution to the newly generated grid.

6.1 One-dimensional profiles

Example 1: Let us start with the classical *hat function* given by

$$u(x) = 1 - r_0^{-1}|x - x_0|, \quad \text{in } [0, 1], \quad (35)$$

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.

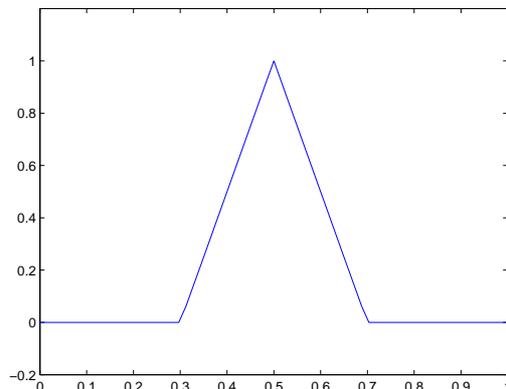


Figure 3: One-dimensional hat function.

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 non-physical 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 from the outset (see also remark (ii) in [44]) 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 $\hat{\sigma}_{la}$ computed directly by means of limited averaging (19) of constant slope values σ_h is depicted in Figure 4 (d). Remarkably, the results recovered by the *monotonized centered* (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.

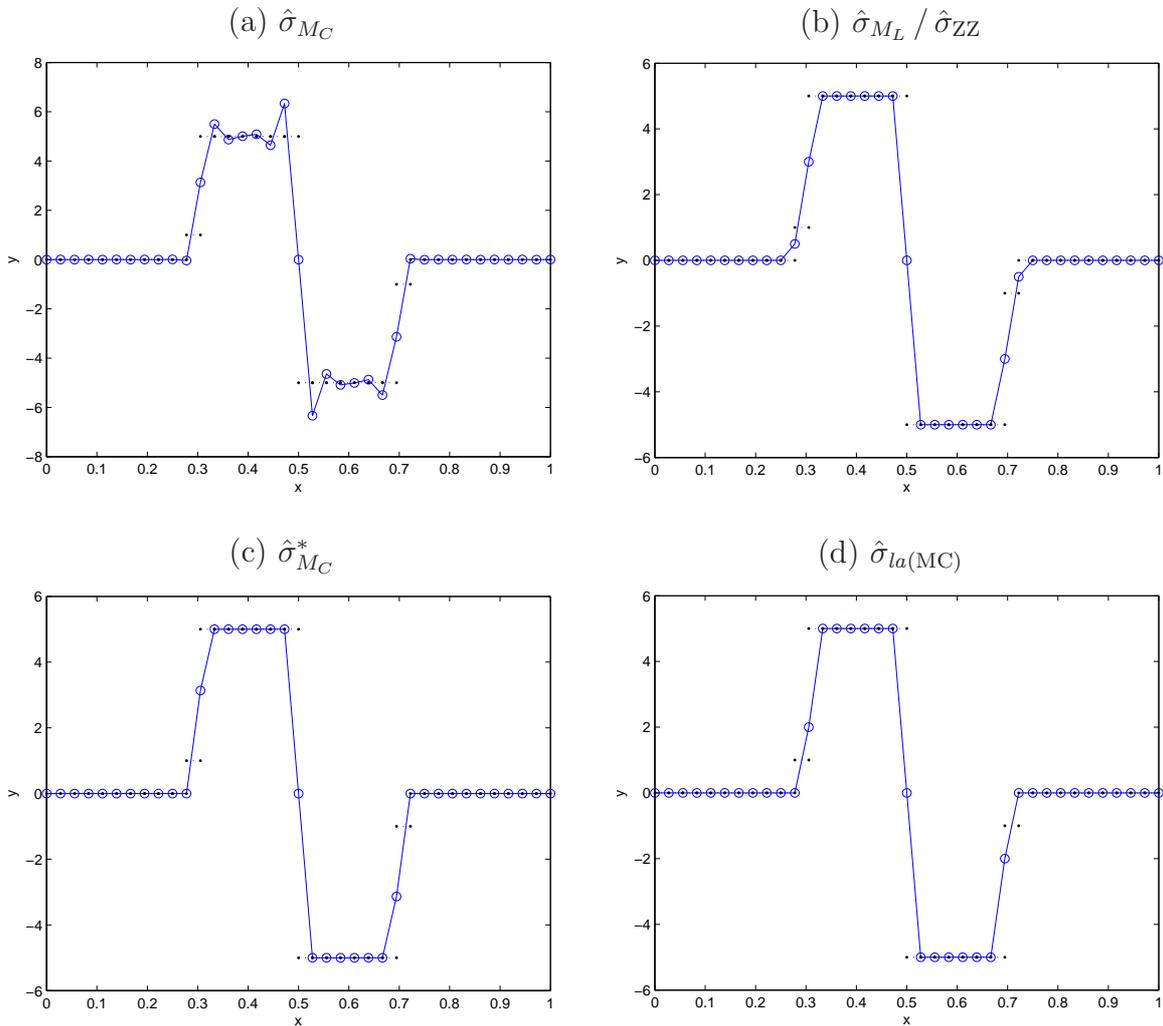


Figure 4: One-dimensional hat function: Recovered gradients.

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

$$u(x) = (4\pi\epsilon)^{-1}e^{-\frac{(x-x_0)^2}{4\epsilon}}, \quad \text{in } [0, 1], \quad (36)$$

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 $\epsilon = 0.005$ 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 $h = 0.0004$.

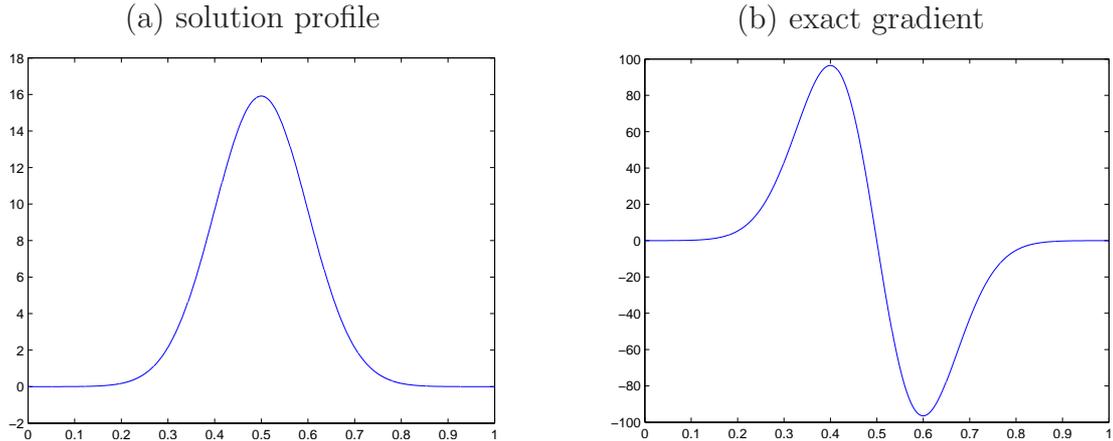


Figure 5: One-dimensional Gaussian hill.

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

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 [27] which can be attributed to the fact that the upper and lower bounds (18) are too restrictive to preserve the accuracy of the 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 $\hat{\sigma}_{MC}$ carries over to its slope limited counterpart. 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 limited averaging techniques constitute a useful tool for the design of cost effective gradient reconstruction procedures which may be extended to the reconstruction of higher order derivatives.

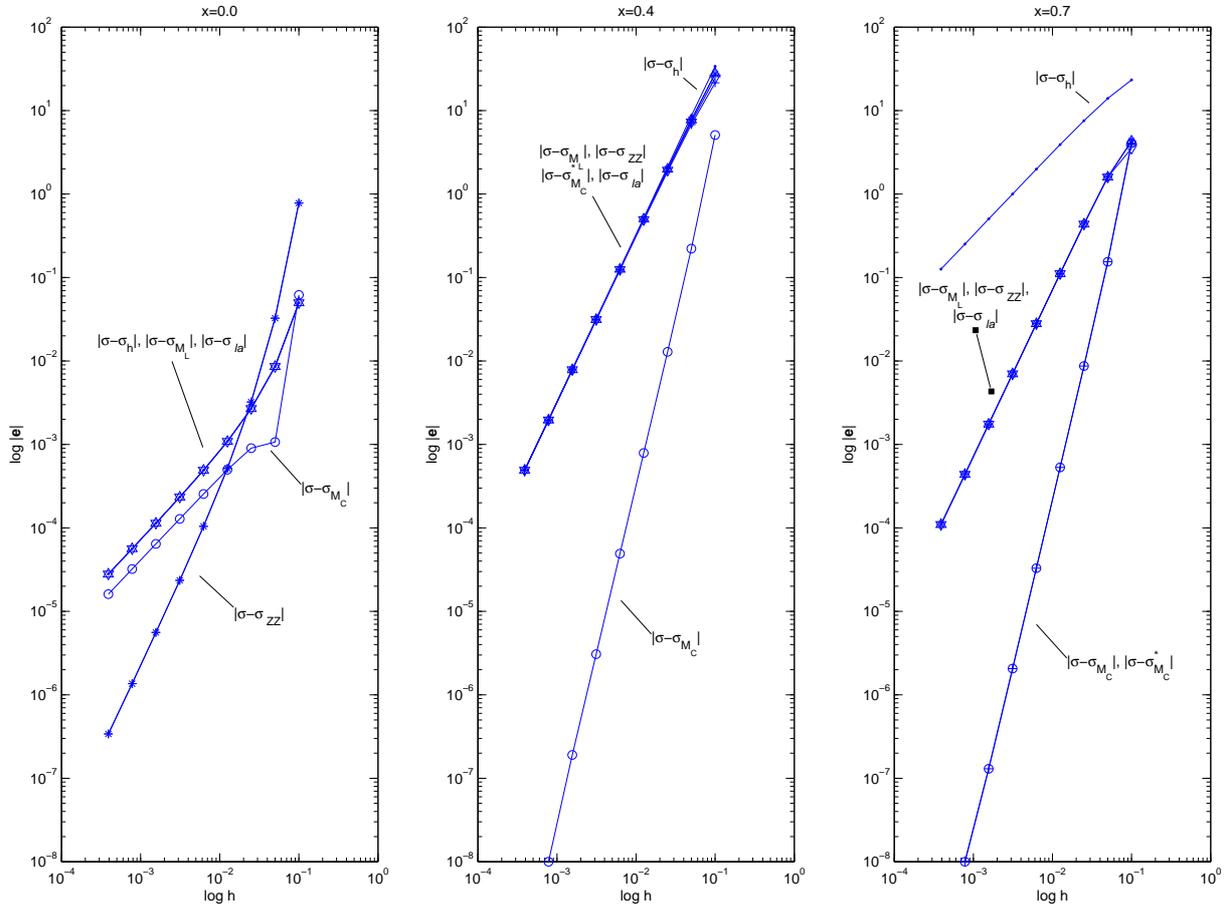


Figure 6: Gaussian hill: Nodal rate of convergence of σ_h , $\hat{\sigma}$ and $\hat{\sigma}^*$.

6.2 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 and 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 $\beta = 36.94^\circ$ 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 [25].

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 $\epsilon = 10^{-7}$. 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 *mesh5* 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 [23] 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 much smaller time step.

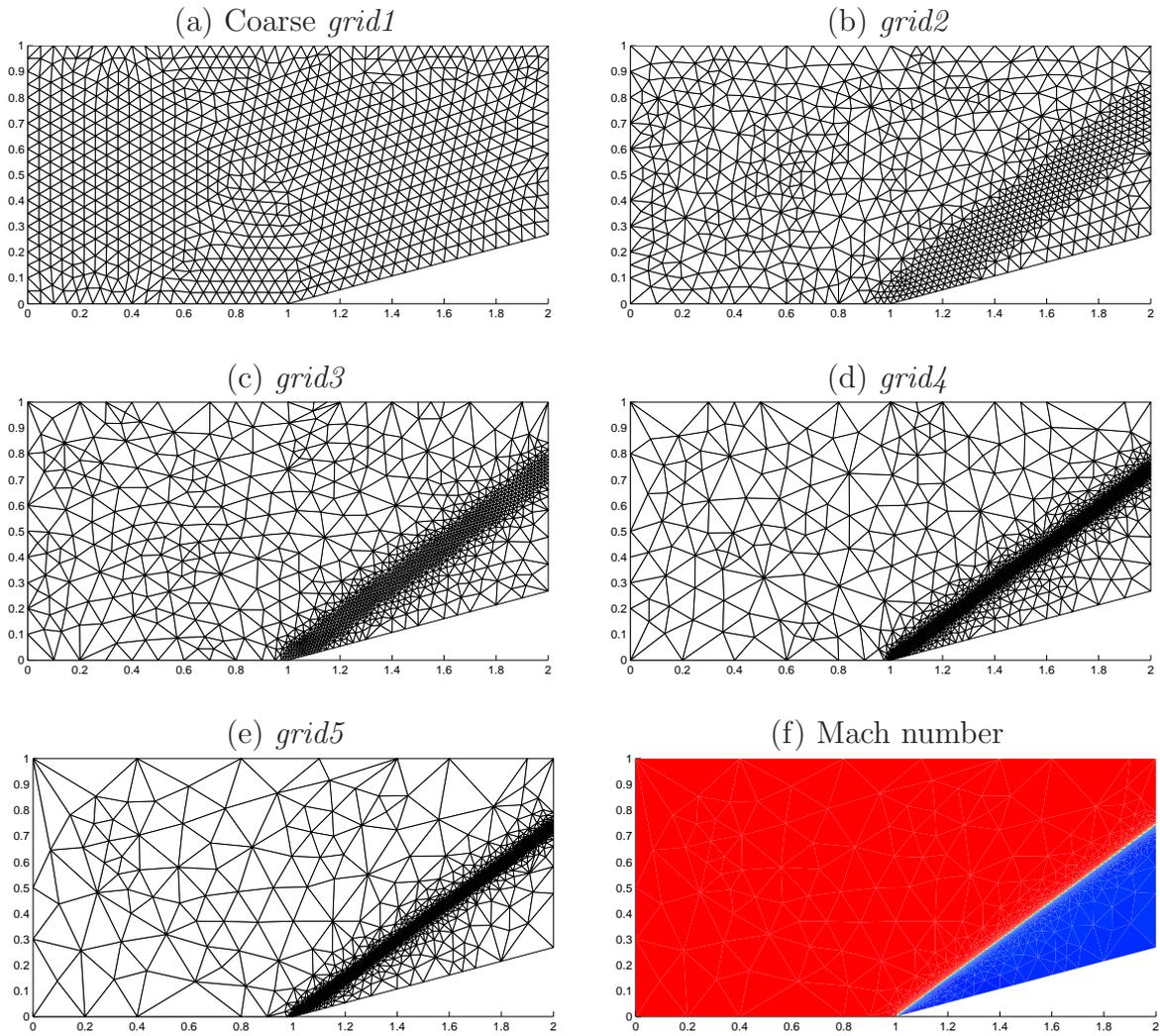


Figure 7: 15° Compression corner at $M_\infty = 2.5$.

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 1. The number of finite elements that constitute the finest grid for $\hat{\sigma}_{ZZ}$ exceeds that for $\hat{\sigma}_{MC}^*$ by as much as 20%. This moderate improvement may be attributed to the very simple structure of the considered benchmark.

	<i>grid1</i>	<i>grid2</i>	<i>grid3</i>	<i>grid4</i>	<i>grid5</i>
$\hat{\sigma}_{ZZ}$	1612	1877	2869	5000	9329
$\hat{\sigma}_{ZZ}^*$	1612	1830	2743	4815	8963
$\hat{\sigma}_{ML}$	1612	1874	2831	4950	9242
$\hat{\sigma}_{ML}^*$	1612	1827	2738	4781	8888
$\hat{\sigma}_{MC}$	1612	1699	2448	4238	7918
$\hat{\sigma}_{MC}^*$	1612	1705	2451	4221	7783

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

6.3 5° Converging channel

Our next example is taken from [33] 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 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 both density and Mach number for the exact and the numerical solution which are presented in Table 2 is quite 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 values on the slice have been recovered by means of linear interpolation from adjacent nodes. Obviously, no additional error has been introduced during the visualization procedure, since the \mathbb{P}_1 -solution 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.

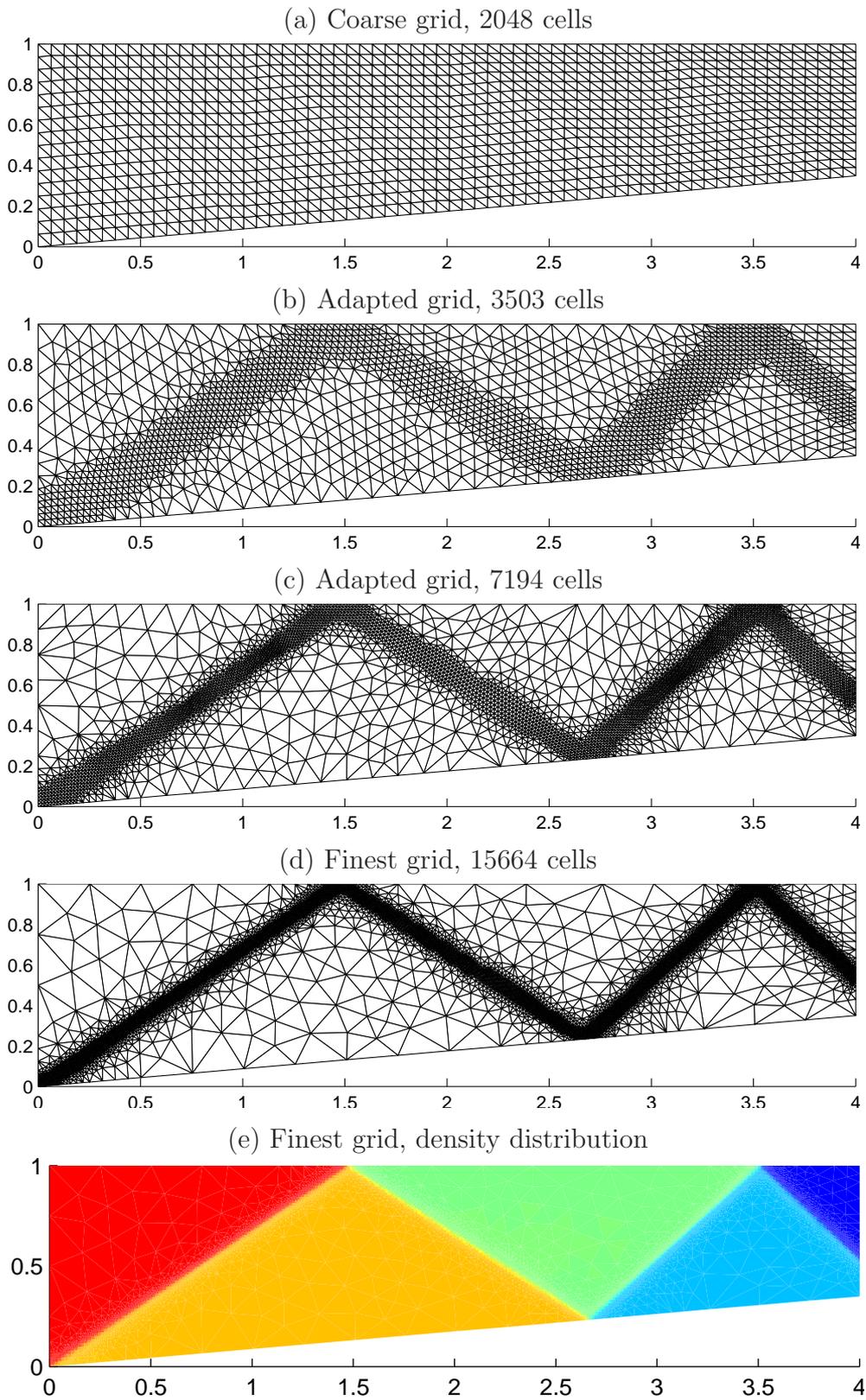


Figure 8: 5° Converging channel at $M_\infty = 2$.

	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

Table 2: Solution values.

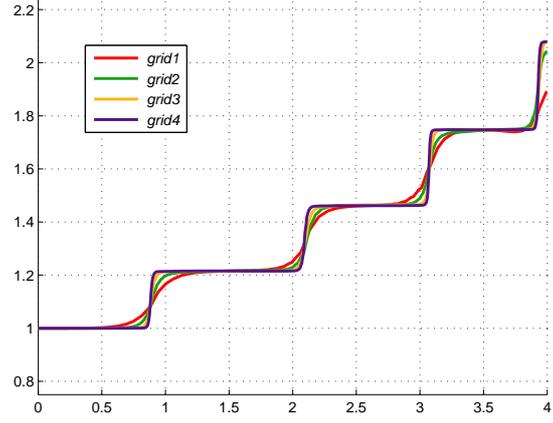


Figure 9: Density cutlines at $y = 0.6$.

6.4 GAMM channel

Our last example deals with a steady transonic flow over a 10% circular bump which is frequently referred to as the GAMM channel benchmark. The computational domain is given by a rectangular channel with a circular arc bump bounded by $\Gamma = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}$:

$$\begin{aligned}
\text{inlet: } \Gamma_{\text{in}} &= \{\mathbf{x} \in \mathbb{R}^2 : x = -1, y \in (0, 1)\}, \\
\text{outlet: } \Gamma_{\text{out}} &= \{\mathbf{x} \in \mathbb{R}^2 : x = 1, y \in (0, 1)\}, \\
\text{solid wall: } \Gamma_{\text{wall}} &= \{\mathbf{x} \in \mathbb{R}^2 : x \in [-1, -0.5], y = 0\} \\
&\quad \cup \{\mathbf{x} \in \mathbb{R}^2 : x \in [-0.5, 0.5], y = \sqrt{1.69 - x^2} - 1.2\} \\
&\quad \cup \{\mathbf{x} \in \mathbb{R}^2 : x \in [0.5, 1], y = 0\} \\
&\quad \cup \{\mathbf{x} \in \mathbb{R}^2 : x \in [-1, 1], y = 1\}.
\end{aligned} \tag{37}$$

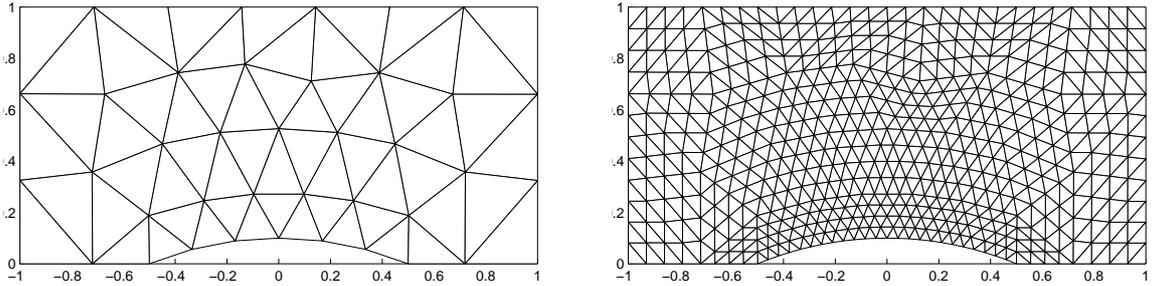


Figure 10: GAMM channel: coarse grid, level 1 and level 3.

The flow structure is very sensitive to the prescribed inlet Mach number which is set to $M_\infty = 0.67$. The initial conditions expressed in terms of primitive variables

$$[\rho, u, v, p] = [1.5 \text{ kg m}^{-3}, 205.709277 \text{ ms}^{-1}, 0 \text{ ms}^{-1}, 101000 \text{ Pa}] \tag{38}$$

are supposed to be constant in the whole domain Ω . In order to accelerate the convergence to a tentative steady state solution, full multigrid has been employed on three levels of regularly refined grids where the coarsest mesh (level 1), depicted in Figure 10 (left), consists of 42 triangles. After three steps of regular refinement the level 3 grid, made up from 944 elements, serves as initial mesh for the adaptation procedure. For each simulation, five sweeps of local grid refinement have been performed with $\eta_{\text{ref}} = 0.1\%$.

The final triangulation resulting from the use of superconvergent patch recovery employed as error indicator is depicted in Figure 11. As to be expected, the grid gets locally refined in the vicinity of the isolated shock wave impinging on the lower wall. A zoom of this region is drawn on the left which illustrates that the bowed shock is detected very well. Its position can be clearly seen from the isolines of the corresponding Mach number distribution presented in Figure 15 (a). For visualization purposes, the numerical solution given on an unstructured finite element mesh was interpolated onto a Cartesian background grid which leads to slight kinks in the isolines for the Mach number.

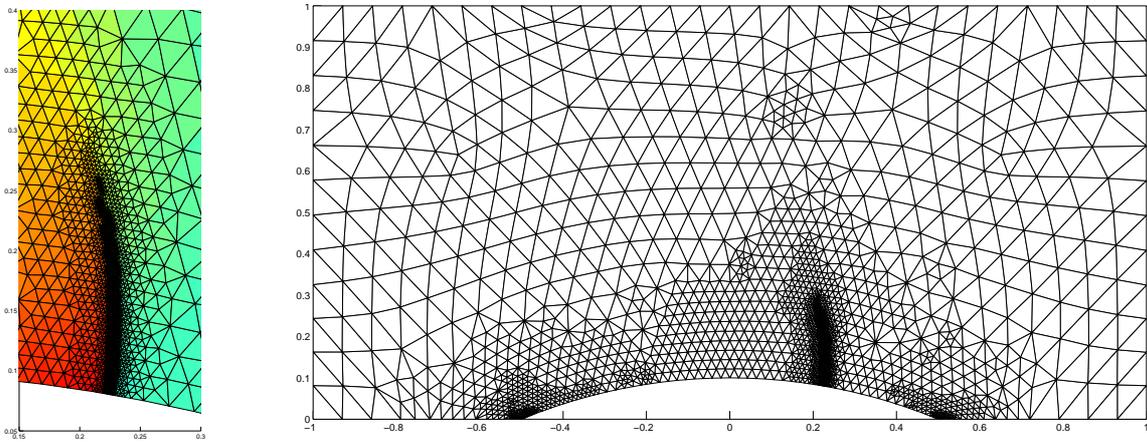


Figure 11: Superconvergent patch recovery, 4563 triangles.

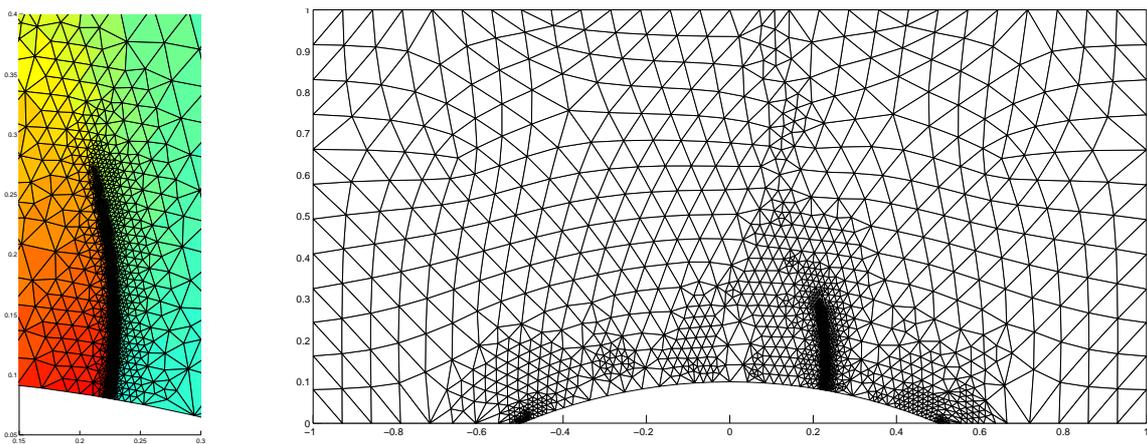


Figure 12: Edgewise limited averaging (MC), 4260 triangles.

Edgewise limited averaging employing the MC limiter function yields nearly indistinguishable results as shown in Figure 15 (b). Remarkably, this new error indicator is available at hardly any cost by just examining the constant slope values adjacent to each edge. At the same time, no additional memory is required for assembling element patches. Moreover, the zone of refined elements in the vicinity of the shock is confined more strictly such that the final computational grid consists of slightly less triangles.

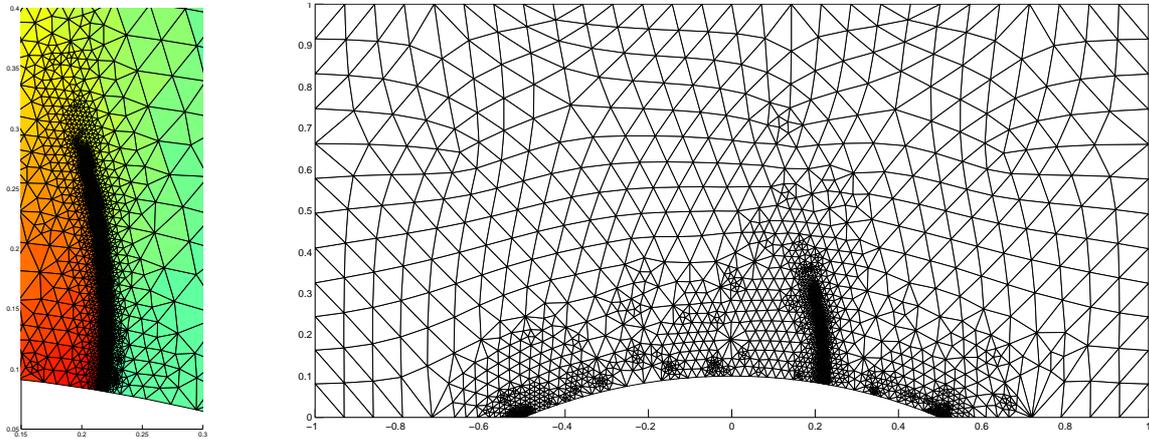


Figure 13: Consistent L_2 -projection with edgewise slope limiting, 6085 triangles.

The merit of slope limiting applied to the recovered gradient values resulting from a consistent L_2 -projection are illustrated in Figure 13. At first glance, the more expensive procedure only increases the number of triangles by a factor of 1.4. However, local grid refinement also takes place near the leading and trailing corners of the bump. It can be seen from Figure 14, that this leads to a significantly improved resolution of the Mach number near the lower wall in comparison to the error indicators based on superconvergent patch recovery and edgewise limited averaging.

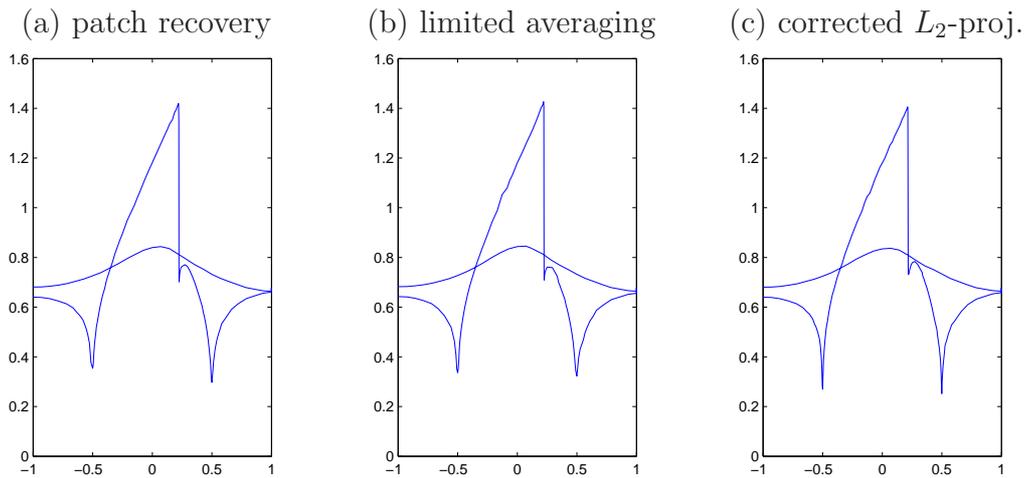


Figure 14: GAMM channel: Mach number distribution along the wall.

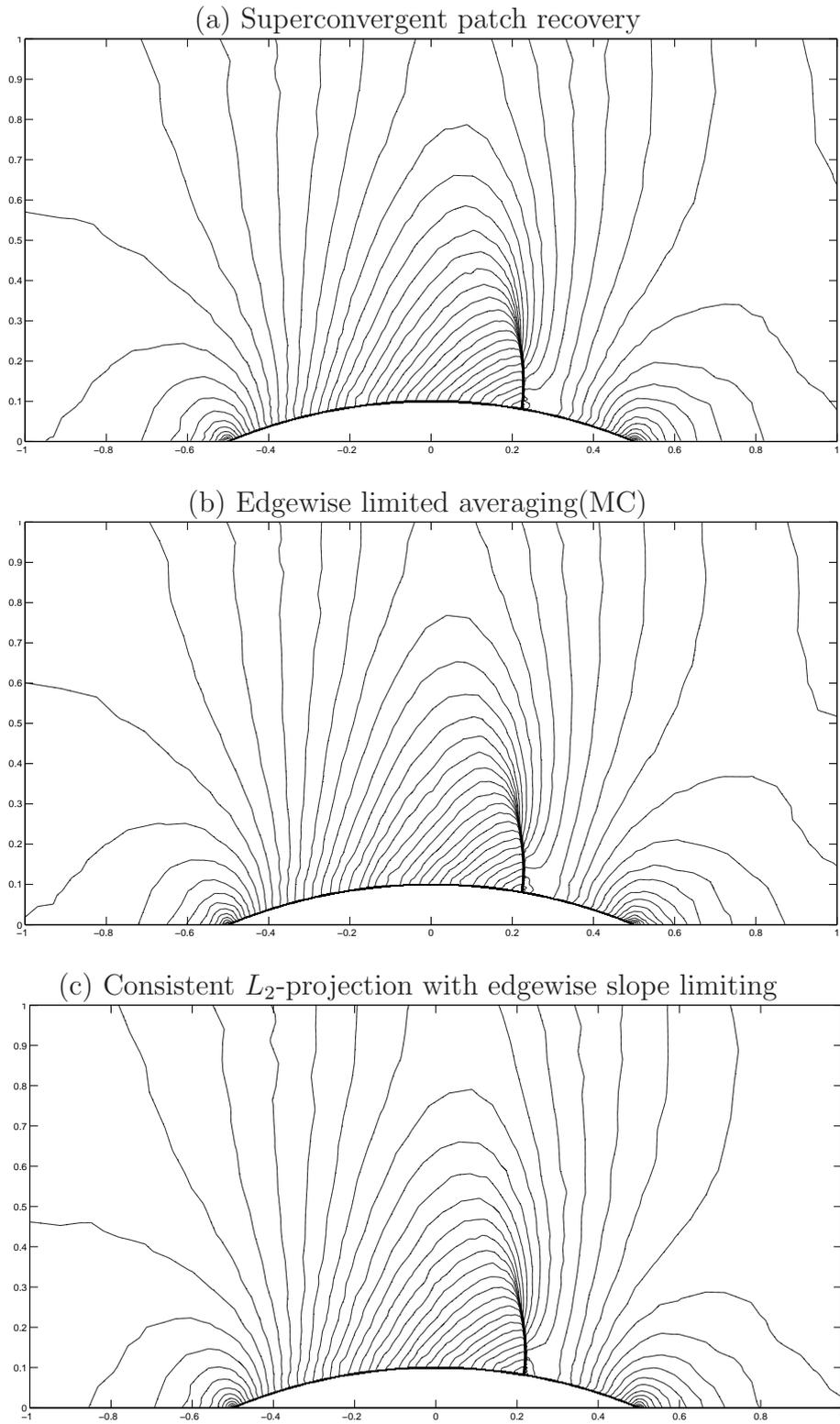


Figure 15: GAMM channel: Isolines of the Mach number.

7 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 shortcomings in the vicinity of steep gradients have been illustrated. Geometrical constraints to be imposed on the high-order slopes have been derived. In particular, the values of the low-order consistent 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 an alternative. They could be 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 [22] have been successfully equipped with self-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 (non-linear) 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 will be considered. An algebraic approach to the design of mesh smoothing/optimization algorithms constitutes an interesting direction for further research.

References

- [1] M. Ainsworth, J.Z. Zhu, A.W. Craig, O.C. Zienkiewicz, Analysis of the Zienkiewicz-Zhu A-Posteriori Error Estimator in the Finite Element Method. *Int. J. Numer. Meth. Engng.* **28** (1989) 2161–2174.
- [2] M. Ainsworth, J.T. Oden, A Posteriori Error Estimation in Finite Element Analysis. John Wiley & Sons, 2000.
- [3] I. Babuška, B. Szabo, Finite Element Analysis. John Wiley & Sons, 1991.
- [4] I. Babuška, W. Rheinboldt, A Posteriori Error Estimates for the Finite Element Method. *Int. J. Numer. Meth. Engng.* **12** (1978) 1597–1615.
- [5] R.E. Bank, The efficient implementation of local mesh refinement algorithms. In: I. Babuška *et. al.* (eds.) *Adaptive Computational Methods*. Springer, 1983.
- [6] R.E. Bank, R.K. Smith, Mesh Smoothing Using A Posteriori Error Estimates. *SIAM J. Numer. Anal.* **34** (1997) 979–997.

- [7] R.E. Bank, PLTMG: A software package for solving elliptic partial differential equations. User's Guide 9.0. Department of Mathematics, University of California at San Diego, 2004.
- [8] M.J. Berger, J. Olinger, Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations. *J. Comput. Phys.* **53** (1984) 484–512.
- [9] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82** (1989) 64–84.
- [10] B. Cockburn, C.W. Shu, TVD Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework. *Math. Comp.* **52** (1989) 411–435.
- [11] B. Cockburn, S. Hou, C.W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.* **54** (1990) 545–581.
- [12] M. Feistauer, J. Felcman, I. Straškraba. Mathematical and Computational Methods for Compressible Flow. Oxford University Press, 2003.
- [13] C.A.J. Fletcher, The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225–243.
- [14] J. Geppert, Adaptive Gitterverfeinerung bei Flux-Limiter-Verfahren. PhD thesis, University of Hamburg, 1996.
- [15] P.A. Forsyth, A Control Volume Finite Element Approach to NAPL Groundwater Contamination. *SIAM J. Sci. Stat. Comput.* **12** (1991) 1029–1057.
- [16] L.A. Freitag, On Combining Laplacian and Optimization-based Mesh Smoothing Techniques, Trends in Unstructured Mesh Generation. *ASME Applied Mechanics Division* **220** (1997) 37–44.
- [17] P. Hansbo, A free-Lagrange finite element method using space-time elements. *Comput. Methods Appl. Mech. Engrg.* **188** (2000) 347–361.
- [18] A. Jameson, Analysis and design of numerical schemes for gas dynamics 1. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *Int. Journal of CFD* **4** (1995) 171–218.
- [19] D. Kuzmin, S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525–558.
- [20] D. Kuzmin, M. Möller, S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. *Computer Meth. Appl. Mech. Engrg.* **193** (2004) 4915–4946.
- [21] D. Kuzmin, S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.* **198** (2004) 131–158.

- [22] D. Kuzmin, M. Möller, Algebraic flux correction I. Scalar conservation laws. In: D. Kuzmin, R. Löhner, S. Turek (eds.) *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 155–206.
- [23] D. Kuzmin, M. Möller, Algebraic flux correction II. Compressible Euler equations. In: D. Kuzmin, R. Löhner, S. Turek (eds.) *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 207–250.
- [24] P.R.M. Lyra, Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction. PhD thesis, University of Wales, Swansea, 1994.
- [25] NPARC Alliance, Computational Fluid Dynamics (CFD) Verification and Validation Web Site: <http://www.grc.nasa.gov/WWW/wind/valid/>.
- [26] J.T. Oden, H.J. Brauchli, On the Calculation of Consistent Stress Distribution in Finite Element Applications. *Int. J. Num. Meth. Engrg.* **3** (1971) 317–325.
- [27] E.S. Oran, J.P. Boris, Numerical Simulation of Reactive Flow. Cambridge University Press, 2001.
- [28] S. Osher, Convergence of generalized MUSCL schemes. *SIAM J. Numer. Anal.* **22** (1984) 947–961.
- [29] S. Prudhomme, Adaptive Finite Element Simulation of a Supersonic Underexpanded Jet. PhD thesis, University of Virginia, 1992.
- [30] S. Prudhomme, J.T. Oden, Computable Error Estimators. In: *Error Estimation Adaptive Discretization Methods in Computational Fluid Dynamics*, T.J. Barth, H. Deconinck (eds), Springer, 2003.
- [31] M.C. Rivara, Design and Data Structure of Fully Adaptive, Multigrid, Finite-Element Software *ACM Trans. Math. Software* **10** (1984) 242–264.
- [32] M.C. Rivara, Using Longest-side Bisection Techniques for the Automatic Refinement of Delaunay Triangulations. *Internat. J. Numer. Methods Engrg.* **40** (1997) 581–597.
- [33] R.A. Shapiro, Adaptive Finite Element Solution Algorithm for the Euler Equations. *Notes on numerical fluid mechanics* **32**, Vieweg, 1991.
- [34] B. van Leer, Towards the ultimate conservation difference scheme II. *J. Comput. Phys.* **14** (1974) 361–376.
- [35] R. Verfürth, A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques. Wiley-Teubner, 1996.
- [36] Z. Zhang, J.Z. Zhu, Analysis of the superconvergent patch recovery technique and a posteriori error estimator in the finite element method (I). *Comput. Methods Appl. Mech. Engrg.* **123** (1995) 173–187.

- [37] Z. Zhang, J.Z. Zhu, Analysis of the superconvergent patch recovery technique and a posteriori error estimator in the finite element method (II). *Comput. Methods Appl. Mech. Engrg.* **163** (1998) 159–170.
- [38] Z. Zhang, Ultraconvergence of the patch recovery technique II. *Math. Comp.* **69** (2000) 141–158.
- [39] Z. Zhang, A. Naga, A meshless gradient recovery method, part I: superconvergence property. *Research Report 2* (2002), Department of Mathematics, Wayne State University.
- [40] A. Naga, Z. Zhang, A Posteriori error estimates based on polynomial preserving recovery. *SIAM J. Numer. Anal.* **42** (2004) 1780–1800.
- [41] Z. Zhang, A. Naga, A new finite element gradient recovery method: Superconvergence property. *SIAM J. Sci. Comput.* **26** (2005) 1192–1213.
- [42] O.C. Zienkiewicz, J.Z. Zhu, A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis. *Int. J. Numer. Methods Eng.* **24** (1987) 337–357.
- [43] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method: Basic Formulation and Linear Problems, Volume 1. McGraw-Hill, 1988.
- [44] O.C. Zienkiewicz, J.Z. Zhu, The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 1: The Recovery Techniques. *Int. J. Numer. Methods Eng.* **33** (1992) 1331–1364.
- [45] O.C. Zienkiewicz, J.Z. Zhu, The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 2: Error Estimates and Adaptivity. *Int. J. Numer. Methods Eng.* **33** (1992) 1365–1382.