Error Indicators for Multilevel Visualization and Computing on Nested Grids

Thomas Gerstner, Martin Rumpf, Ulrich Weikard

Department for Applied Mathematics, University of Bonn, 53115 Bonn, Germany

Abstract

Nowadays computing and post processing of simulation data is often based on efficient hierarchical methods. While multigrid methods are already established standards for fast simulation codes, multiresolution visualization methods have only recently become an important ingredient of real–time interactive post processing. Both methodologies use local error indicators which serve as criteria where to refine the data representation on the physical domain. In this article we give an overview on different types of error measurement on nested grids and compare them for selected applications in 2D as well as in 3D. Furthermore, it is pointed out that a certain saturation of the considered error indicator plays an important role in multilevel visualization and computing on implicitly defined adaptive grids.

1 Introduction

Today, efficient adaptive multigrid methods are available for a large variety of simulation problems [11]. Following this frontier a variety of multiresolution visualization methods has been designed to serve as tools for interactive visualization of large data sets [3,9,12,21]. Here we will especially focus on methods which are based on nested grids since they frequently appear in data computed by multigrid methods [8]. The local resolution of the generated visual objects, such as 2D graphs, or isosurfaces and color shaded slices in 3D, depends on error indicators which measure the error due to a locally coarser approximation of the data.

Different approaches have been presented to solve the outstanding continuity problem, i.e. to avoid cracks in adaptive isosurfaces. In the Delaunay approach by Cignoni et al. [4] and in the nested mesh method by Grosso et al. [10] the successive remeshing during the refinement guarantees continuity. Alternatively, Shekhar et al. [22] rule out hanging nodes by inserting additional points on faces with a transition from finer to coarser elements due to an adaptive stopping criterion.

We apply the method of adaptive projection on nested grids, which has been described in earlier publications. Thereby we do not explicitly generate an adaptive grid, but during a hierarchical traversal of the whole nested grid, error indicator values implicitly define those elements, which build up the adaptive subset depending on a prescribed threshold value. For the general concept we refer to [18]. Implementational aspects are especially described in [17]. The core of our approach is identical to the method of Zhou et al. [25]. In 3D it can be regarded as a generalization of the techniques presented by Livnat et al. [14] and in [7].

In this paper we give a detailed comparison of error indicators and the performance of corresponding multilevel methods. Furthermore we outline the usage of such error indicators also for computational purposes. We do not focus on the methodology itself but on the indicators, their effect on cost reduction, and their relation to the actual error in a corresponding norm. Therefore, especially to simplify the exposition, we confine ourselves to simplicial grids generated by bisection, which are well known from adaptive numerical methods [1,20]. In explicit, we deal with the recursive bisection of [15,16]. Let us emphasize that the methodology is fairly independent of the selected grid type (cf. Figure 4, which shows slices of a medical data set on adaptive tetrahedral and hexahedral grids, respectively).

2 A general multilevel algorithm on nested grids

We confine ourselves here to hierarchical simplicial grids which carry a piecewise linear data function. Let us consider a family of nested, conforming, simplicial meshes $\{\mathcal{T}^l\}_{0\leq l\leq l_{\max}}$ in two or three dimensions. Here, conforming means that hanging nodes do not arise. We denote by $h(T)$ the diameter of a simplex $T \in \mathcal{T}^l$ and by $h(e)$ the diameter of its longest edge e. Furthermore, $\mathcal{N}(T)$, $\mathcal{N}(\mathcal{T}^l)$ denote nodal sets of single simplices, respectively entire simplicial grids. The simplices, triangles in 2D, respectively tetrahedra in 3D are assumed to be refined by recursive bisection. For a simplex T, the midpoint of a predestined edge $e_{\text{ref}}(T)$ is thereby picked up as a new node $x_{\text{ref}}(T)$, and the simplex is cut at the edge, respectively face, $F_{\text{ref}}(T)$ spanned by $x_{\text{ref}}(T)$ and the nodes of T, which are not endpoints of the refinement edge $e_{\text{ref}}(T)$, into two child simplices $\mathcal{C}(T) = \{T^1_{\mathcal{C}}, T^2_{\mathcal{C}}\}$. A simple alternating scheme for the refinement edge e_{ref} [1,15] guarantees the conformity of the resulting grids. Finally, let U^l denote the piecewise linear function on $\mathcal{T}^l$ uniquely described by the data values on the corresponding nodes.

The multilevel algorithm for computing or postprocessing is based on a depth first traversal of the grid hierarchy. During such a depth first traversal of the binary tree structure we check for a stopping criterion on every simplex T. If it is true we stop and visualize locally. Otherwise, we recursively proceed on the child set $\mathcal{C}(T)$. If we stop on a specific simplex T and refine another simplex $\tilde{T}$ which shares the refinement edge with T, i.e. $e_{\text{ref}}(T) = e_{\text{ref}}(\tilde{T})$, an inconsistency occurs at the hanging node x_{ref}. This leads to jumps in the current adaptive data projection. In the case of general nested grids we can apply a specific adaptive projection operators to ensure consistency [18]. Here, we simply have to ensure that, whenever a simplex is refined, all the simplices sharing its refinement edge – in 2D only the one triangle opposite to T at the edge $e_{\text{ref}}(T)$ – are refined as well. This can be achieved by defining error indicators $\eta(x)$ on the grid nodes and choosing $\eta(x_{\text{ref}}(T)) < \varepsilon$ as stopping criterion on a simplex for a user–prescribed threshold value ε. Since all nodes, except those on the coarsest level, are refinement nodes $x_{\text{ref}}(T)$ on a refinement edge $e_{\text{ref}}(T)$, the indicator value $\eta(x)$ measures the approximation error or a "visual" error on those simplices sharing the edge. Therefore, the recursive traversal would stop not only on T but – if visited – on all other simplices sharing the refinement edge if their common stopping criterion is true.

However, for an arbitrary error indicator it might still occur that, although $\eta(x_{\text{ref}}(T)) < \varepsilon$, $\eta(x_{\text{ref}}(\hat{T})) \geq \varepsilon$ on some descendant $\hat{T}$ of a neighbouring simplex of T whose refinement node $x_{\text{ref}}(\hat{T})$ is located on the boundary of T. The adjacent simplex will possibly be visited and then refined, whereas on T the stopping criterion already holds. To avoid this we assume the following saturation condition on the error indicator (for a generalization compare [18]):

Saturation condition on an error indicator η:
$\eta(x_{ref}(T)) > \eta(x_{ref}(T_{\mathcal{C}}))$ for all $T \in \mathcal{T}^l$ with $l < l_{max}$ and $T_{\mathcal{C}} \in \mathcal{C}(T)$.

An error indicator η is called admissible, if it fulfills the saturation condition. Otherwise, it can easily be adjusted in a preroll step (cf. section 3). The adaptive algorithm can be sketched in pseudo code as follows

```
AdaptTraverse(T) {
    if SimplexIsOfInterest(T)
        if C(T) ≠ ∅ ∧ η(x_ref(T)) ≥ ε
            { AdaptTraverse(T_C^1); AdaptTraverse(T_C^2); }
        else SimplexAction(T);
}
```

where the function *SimplexIsOfInterest()* checks whether the simplex is a candidate for the local processing or not. For example, in multilevel isosurface extraction this function checks whether the isosurface intersects the current simplex (see Section 5 for an efficient implementation of such a function).

3 Principle aspects of nodal error indicators

At first, let us discuss several principle techniques of local error measurement. The starting point will be some actual local error measure on the grid hierarchy. The local resolution and the visual impression of the numerical data are closely related to the specific type of error measurement applied in the adaptive traversal of the tree structure.

Let $\eta^*(x)$ be a measure on nodes x, which weights the effect of stopping for some local rendering already on a simplex T with $x = x_{\mathrm{ref}}(T)$ instead of traversing the locally finest grid level. Furthermore, let us denote by $S(x)$ the support of the piecewise linear base function corresponding to the node x. Then, given a fine grid data function U and a coarse grid function U^l on level l with $T \in \mathcal{T}^l$, we assume $\eta^*(x)$ to be the distance between U and U^l measured locally on $S(x)$ by some metric $d_{S(x)}$, i.e.

$$\eta^*(x) = d_{S(x)}(U, U^l)$$

Let us consider several widely used metrics:

- We can choose some local norm of the difference functions such as

  $$\eta^*(x) := \|U - U^l\|_{p,S(x)},$$

  where $\|\cdot\|_{p,S(x)}$ is the usual L^p norm for $p \in [1, \infty]$ restricted to the domain $S(x)$. Due to Hölder's inequality the error indicators obviously become sharper for increasing values of p.
- Instead of function values we can consider derivatives and define

  $$\eta^*(x) := \|\nabla U - \nabla U^l\|_{p,S(x)}.$$

  In general the resulting error measurement is sharper then the one based on function values. By some worst case analysis based on inverse estimates we obtain the estimate $\|\nabla U - \nabla U^l\|_{p,S(x)} \leq C h_{\min}^{-1} \|U - U^l\|_{p,S(x)}$, where $h_{\min} := \min_{T \in \mathcal{T}_l, T \subset S(x)} h(T)$. This estimate is asymptotically sharp on fine grid levels for a function U, which is the interpolation of some smooth function. Frequently, the norm of the gradient is taken as an error indicator. Especially in post processing this is questionable, because rendering is "linearly exact" and therefore refinement in areas of uniformly large gradient norms does not improve the graphical representation.
- Third – a smooth graphical representation in mind – we may be interested in measuring the geometric smoothness of the approximation independently of the true function values. A possible measure is a discrete curvature quantity. For surfaces this should be related to the absolute curvature $\kappa = \sqrt{\kappa_1^2 + \kappa_2^2}$ where the κ_i are the principle curvature terms. As clearly indicated in the

case of minimal surfaces with vanishing mean curvature or cylinders with vanishing Gaussian curvature, mean or Gaussian curvature discretization does not make sense in terms of general error control.

- A fourth choice of a suitable measure is closely related to geometric shapes [13]. In the simple case of a scalar function U a suitable approach is to compare the graphs of U, respectively U^l on $S(x)$. If $\mathrm{dist}(\cdot,\cdot)$ is a geometric distance metric on graphs, we are lead to $\eta^*(x){:=}\mathrm{dist}(graph(U), graph(U^l))$. Let us emphasize that for rather flat surface graphs this error indicator only slightly differs from measuring the difference of the function values.

Furthermore, the viewing direction and distance may enter the error metric [14], or the error measurement may depend on the distance to a specific region of interest [2,5,18]. We will here restrict ourselves to the basic error norms and discrete curvature measurement.

4 Different hierarchical error indicators

Usually, an error measurement which locally compares coarse grid functions with the functions on the finest grid is expensive to evaluate even in a preprocessing step. We will apply an often used simplification, which only compares data on the current grid level to data on the next finer grid level. We will denote the corresponding "one level look ahead" error indicator by $\eta(x)$. However, the saturation condition as a minimum precondition to guarantee continuity of the adaptive projection may fail for η.

- **Hierarchical offset error indicators:** In analogy to the norm of the difference function we can consider the hierarchical offset function U_δ defined on a simplex as

  $$U_\delta|_T = U^l|_T - U^{l-1}|_T$$

  The values of U_δ on $\mathcal{N}^l \setminus \mathcal{N}^{l-1}$ are related to the original data values by the following recursive formula

  $$U^l(x_{\mathrm{ref}}(T)) = \frac{U^l(x_1) + U^l(x_2)}{2} + U_\delta(x_{\mathrm{ref}}(T))$$

  where x_1 and x_2 are the end points of the edge corresponding to $x_{\mathrm{ref}}(T)$ on a simplex T. For smooth data, i.e. $U(x) = u(x)$ for all nodes x with $u \in C^2$, $|U_\delta(x_{\mathrm{ref}}(T))| = O(h(T)^2)$, which implies the saturation condition holds asymptotically on grids $\mathcal{T}^l$ for l sufficiently large. Hence, we define the hierarchical L^∞ error indicator

  $$\eta_\infty(x){:=}|U_\delta(x)|\,.$$

Instead of the L^∞ norm we can analogously consider different integral norms applied to the difference function which corresponds to a new node. Using lumped mass integration one obtains for a d–dimensional simplex T

$$\eta_p(x) = \frac{1}{d+1}\left(\sum_{T\,,x\in\mathcal{N}(T)} |T|\right)^{\frac{1}{p}} |U_\delta(x)|$$

for $1 \le p < \infty$, where $|T|$ is the volume of the simplex. Decreasing p leads to an earlier stopping of the tree traversal on simplices of small size.

- **Gradient type error indicators:** Instead of measuring the error with respect to function values, we can consider the error of the function gradient:

$$\eta_{1,p}(x) := \begin{cases} \left(\sum_{T\,,x\in\mathcal{N}(T)} |T|\right)^{\frac{1}{p}} \|\nabla U_\delta|_T\| & \text{for} \quad 1 \le p < \infty, \\ \max_{T\,,x\in\mathcal{N}(T)} \|\nabla U_\delta|_T\| & \text{for} \quad p = \infty. \end{cases}$$

The evaluation of these error indicators takes some effort in the precomputing step. If we replace simplices by simplex refinement edges without modifying the scaling we gain at least for $p = \infty$

$$\eta_{1,e} := \frac{2\,|U_\delta(x_{\text{ref}}(T))|}{h(e_{\text{ref}}(T))}.$$

- **Discrete curvature type indicators:** With a focus on the geometric shape of an isosurface, we will now consider an error indicator related to curvature. Let us recall that the continuous curvature vector of a iso–hypersurface [6] can be evaluated by

$$-|\nabla u| \text{div}\left(\frac{\nabla u}{|\nabla u|}\right) .$$

Now, we ask for some discrete counterpart. We replace the divergence by a discrete difference and scale it by the inverse grid size. Thus, we obtain a scaled discrete curvature quantity which locally measures the quality of the data approximation from the perspective of the visual appearance [18].

On each simplex the data gradient ∇U^l is always perpendicular to an isoline or isosurface. Therefore, at any face F the normal component of the jump of the normalized gradient, denoted by $[\frac{\nabla U^l}{|\nabla U^l|}]_F$, locally measures the fold in the data function and motivates the definition

$$\eta_N(x_{\text{ref}}(T)) := \left[\frac{\nabla U^l}{|\nabla U^l|}\right]_{F_{\text{ref}}(T)} .$$

Here the jump operator $[\cdot]_F$ is defined as the difference of the argument on both sides of the face. We can also apply the simplification of the previous indicator here and denote the resulting error indicator by $\eta_{N,e}$.

5 Saturated error indicators

As pointed out above, the hierarchical error indicators in general do not fulfill the saturation condition. We can overcome this drawback by defining a modified error indicator $\bar{\eta}$, which is defined as the minimal saturated error indicator larger or equal than η. This definition is constructive in the sense that in a bottom up, breadth first traversal of the grid, we can blow up the original error indicator values. In pseudo code this blow up mechanism looks as follows:

```
for (l = l_max − 1 ; l ≥ 0 ; l−− )
    for all T ∈ 𝒯^l and x = x_ref(T)
        η̄(x) = max{ max_{T_C ∈ C(T)} η̄(x_ref(T_C)), η(x)};
```

Let us emphasize that a depth first traversal of the hierarchy in the adjustment procedure would not be sufficient. But, if the error indicators are adjusted as proposed above continuity problems are solved automatically.

Alternatively, we can ensure saturation of an indicator η by recursively defining:

$$\eta^+(x) = \eta(x) + \max_{T_{\mathcal{C}} \in \mathcal{C}(T)} \eta^+(x_{\mathrm{ref}}(T_{\mathcal{C}})) \qquad \text{for } x = x_{\mathrm{ref}}(T).$$

On the finest grid level, where $\mathcal{C}(T) = \emptyset$, we simply set $\eta^+(x) = \eta(x)$. The different error measures are obviously related to each other by $\eta \leq \bar{\eta} \leq \eta^+$. Applying the triangle inequality we obtain $\eta^*_\infty \leq \eta^+_\infty$ and $\eta^*_{1,\infty} \leq \eta^+_{1,\infty}$. The indicator η^+, although the largest one derived from the original indicator η^*, and thus the weakest, can have other desirable properties. For instance, an easy computation of min/max–values for isosurface extraction or criteria for multilevel backface culling are possible:

- On the one hand, we are able to compute a bound $\beta_0(T)$ for the difference of the true function and its linear approximation on a simplex T on level l. This can be applied in the implementation of the *SimplexIsOfInterest()*–function. We obtain

  $$\min_{x \in T} U^l(x) - \beta_0(T) \leq U \leq \max_{x \in T} U^l(x) + \beta_0(T)\,.$$

  In the hierarchical offset case and for the choice η^+_∞ we simply define

  $$\beta_0(T) = \begin{cases} \frac{1}{2}\eta^+_\infty(x_{\mathrm{ref}}(T)), & \text{for hierarchical offset indicators} \\ h(T)\eta^+_{1,\infty}(x_{\mathrm{ref}}(T)), & \text{for gradient type indicators} \end{cases}$$

In both cases, the expensive storing of min/max–values, which was necessary to prevent us from missing certain components of the isosurface (cf. [24]) can be avoided.

- On the other hand, we may check in an isosurface extraction algorithm – based on coarse grid simplices – whether all polygons extracted by the algorithm will be backfaces. Let $N^l = \frac{\nabla U^l}{\|\nabla U^l\|}$ denote the normal of some triangle of the final isosurface triangulation on the simplex $T \in \mathcal{T}^l$, and V the viewing vector from the object to the eye (we confine ourselves here to parallel projection). If $N^l \cdot V \geq 0$, the triangle is faced towards the viewer. Otherwise it does not need to be drawn. We obtain a significant acceleration of our isosurface algorithm, if on a much coarser grid level we recognize simplices containing only isosurface triangles which are faced away from the viewer. In this case we are already able to stop the local traversal at this level. If $\beta_N(T)$ is a bound of the variation of N^l in $T \in \mathcal{T}^l$, we obtain the multilevel backface test $N^l \cdot V + \beta_N(T) \leq 0$. One possible choice for $\beta_N(T)$ is $\eta_N^+(x_{\text{ref}}(T))$. Skipping the normalization and considering instead a bound $\beta_1(T)$ which measures the possible offset in $\|\nabla U\|$, we alternatively obtain the rejection criterion $\nabla U^l \cdot V + \beta_1(T) \leq 0$, with $\beta_1(x) = \eta_{1,\infty}^+(x_{\text{ref}}(T))$. It can easily be seen that, on average, while arbitrarily rotating the object, we save up to one half of the computing time by traversing only half of the tetrahedra for the extraction of an isosurface (Fig. 5).

6 A quantitative comparison

So far, qualitative aspects of error measurement and provisions for different applications have been discussed. In what follows, we focus on a detailed quantitative discussion. Therefore, we study certain test problems in 2D as well as in 3D.

In 2D we consider different examples from different classes of data sets. On the one hand, we choose a typical measurement data set, which represents a geographical map, originally sampled on a 257^2 regular grid, which we afterwards cover with a hierarchical triangular grid (cf. Fig. 7(b)). It consists of regions with a significant roughness and other areas which are almost planar. On the other hand, we apply multilevel visualization to a typical numerical data set already computed on a triangular grid hierarchy. It is characterized by smooth, less steep areas which alternate with thin transition zones where the data function is rather steep. Here we consider a timestep of a Cahn–Hilliard simulation on the same 257^2 regular grid (cf. Fig. 7(c)). It represents the density of an alloy after quenching (rapidly cooling), which leads to phase separation [23]. Nevertheless, this numerical data set is much smoother than the geographical map.

In the 3D case we consider isosurface extraction and color slicing on the well known 129^3 bucky ball data set (cf. Fig. 7(f) and 7(d)).

The cost of the visualization method is mainly controlled by the number of visited grid cells in the recursive traversal. As there are different ranges of indicator values on the grid nodes, we ensure comparability by normalizing the maximal indicator value to 1. An alternative measure of the cost would be the number of rendered primitives. Not surprisingly both measures are closely related and therefore it does not really matter which one we choose. The following results are based on the visited–cell–count cost measure.

The crucial measure of the quality of an adaptive projection in visualization is the visual impression of the rendered image. However, this is impossible to quantify. So in order to get a comparable notion of the quality we chose the reciprocal of the corresponding global norm of the difference between the adaptively extracted function and the function on the finest grid. In this context the efficiency E of an error indicator is the quotient of quality and cost and would thus be

$$E_\eta(U,\varepsilon) = \frac{1}{k \cdot \|P_\eta U - U\|}$$

where k is the number of visited cells used for the adaptive projection $P_\eta U$.

Fig. 1 and 2 compare results obtained for the different classes of error indicators. The scaling on the y axes is logarithmic. For the geographical data the different characteristics of the hierarchical offset error indicators compared to the error indicators based on derivatives are clearly visible.

The smoother numerical data show a similar behaviour. Not surprisingly, the graphs for $\bar{\eta}_N$ and $\bar{\eta}_{N,e}$ are especially for the geographical data set nearly the same. Therefore, the simplification incorporated in $\bar{\eta}_{N,e}$ seems to be admissible and as $\bar{\eta}_{N,e}$ is easier to calculate, it is more favourable for practicable purposes than $\bar{\eta}_N$. As expected, the indicators $\bar{\eta}_1$ and $\bar{\eta}_2$ are – in comparison to $\bar{\eta}_\infty$ – rather similar.

The efficiency of these indicators is depicted in Fig. 3. It becomes clear that $\bar{\eta}_\infty$ is less efficient than $\bar{\eta}_1$ and $\bar{\eta}_2$. In the case of the geographical data set and also for not too high threshold values in the case of the numerical data, the qualities of the three error indicators differ only slightly. So the main reason for the low efficiency of $\bar{\eta}_\infty$ is that even for high threshold values a large number of cells is visited. In the 3D–case the results are similar.

Finally, we compare the different methods for ensuring the saturation condition in the case of the η_∞–indicator. In our experiments the differences in smoothness between the geographical and the numerical data are clearly visible in the characteristic if $\bar{\eta}_\infty$ is used. However, these differences are lost for η^+_∞. This is also true for other indicators as for example η^+_2 compared to $\bar{\eta}_2$. Hence, an application of an η^+–type saturated indicator is only reasonable if the advantages concerning min/max–bounds or backface culling are exploited.

We also want to show that for reasonable threshold values the visual impression of the original and adaptively projected images are rather close (Fig. 7(a) and 7(b)). For the geographical map, the adaptive image consists of 13666 patches whereas the original image has a size roughly ten times larger (131072 patches). Additionally, we show extracted isosurfaces of the bucky ball data set with 128709 and 590018 triangles, respectively (cf. Fig. 7(e) and 7(f)). In all these figures we used the $\bar{\eta}_\infty$ error indicator.

7 Computing on implicitly defined grids

Frequently, a graph structure managing the hierarchy of elements – explicitly stored in memory – serves as the algorithmical base of an adaptive finite element method. For very large applications with several million elements, such a strategy is characterized by an enormous memory consumption. The typical memory requirement for a single node in such a structure is several hundred bytes, much more than the four or eight bytes needed for the nodal value itself. On the other hand, information which elements have to be refined already uniquely defines the adaptive grid. Thus, the alternative for an explicitly stored element graph is to generate the actual grid elements temporarily and locally during a recursive grid traversal also in the numerical algorithm. Thereby nodal values are either stored in an array corresponding to a fixed fine grid, or they are retrieved on demand from a hash table. In case of simplicial grids the hash key corresponding to a new node $x_{\text{ref}}(T)$ can be taken as the concatenation of the known indices of the parent nodes. Every time a new node is generated a new nodal index is provided as well.

Furthermore, the basic operations on the grid and on the nodal vectors which are performed in a finite element code can be implemented with the help of the general traversal algorithm of Section 2. To give an impression how this can be done concretely let us briefly study the implementation of a prodcut of a finite element matrix with a vector $z = Aw$. Such a product is the key ingredient of any interative solver. Here A is a discretization matrix (e.g. a stiffness matrix) and w a nodal vector. Finite element matrices split up into local element matrices which consider local basis functions only. Thus we are able to replace the iteration over the nodal indices by an *outer grid traversal* running over all elements of the grid and an *inner iteration* running only over the local nodes. After an initialization $z = 0$ we invoke the traversal routine with the following local action on every simplex:

```
SimplexAction(E) {
    assemble local matrix on E;
    compute local matrix vector product;
    for all x ∈ 𝒩(E)
        update z(x);
}
```

Here the *update* consists in the identification of global nodes corresponding to local indices and adding the corresponding local z vector components to the already assembled global vector component $z(x)$. For details we refer to [11].

Figure 6 shows an image smoothing application solved by nonlinear diffusion [19]. In each timestep one is lead to a linear system $(M + \tau B)u = f$ which has to be solved in the nodal vector u. Here, M is the mass matrix, τ is the timestep, B is a nonlinear stiffness matrix and f the nodal vector on the right hand side. Edge indicators (i.e. $\eta_{1,\infty}$) are used as error indicators for this application. They ensure a sufficient grid refinement at edges in the images.

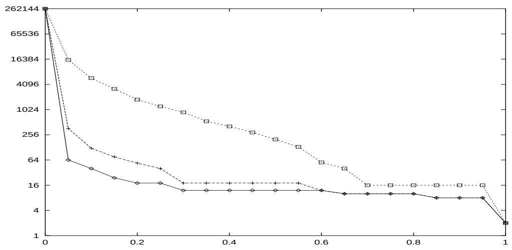
8 Concluding remarks

In this paper we have considered several error indicators which are used in multiresolutional visualization and compared their quantitative as well as their qualitative properties. We have specifically looked at local norms of difference functions, differences of gradients and discrete curvature measures. We have employed the saturation condition as an important prerequisite for interactive visualization since it solves the continuity problem.

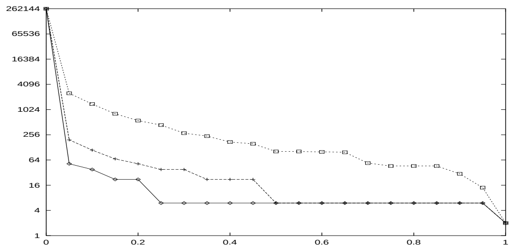
It has been shown how this condition can be fulfilled by a blowup of η. We have thereby defined minimal saturated hierarchical error indicators $\bar{\eta}$, which indeed give good results concerning the ratio of triangle count to global error.

On the other hand, by a minor modification of the blowup mechanism, we have defined the error indicators η^+ which have a slightly worse efficiency but other desirable properties. For instance, based on the error indicator data bounds on simplices can be computed which then serve as stopping criteria for multiresolutional isosurface extraction. We have also shown how gradient type error indicators allow multilevel backface culling. In a series of numerical experiments on application data the different error indicators have been compared and their mutual advantages have been outlined.

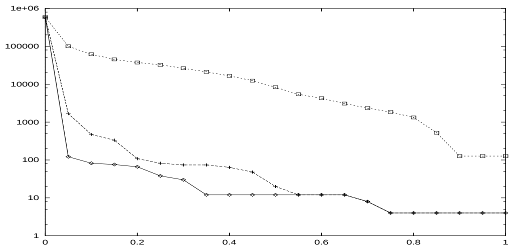
Let us finally remark that the user can choose which error indicator to use. He or she may be interested in robust maximum error bounds (η_∞), local mass measurement (η_1, η_2), surface shading ($\eta_{1,\infty}$) or discrete curvature control (η_N).



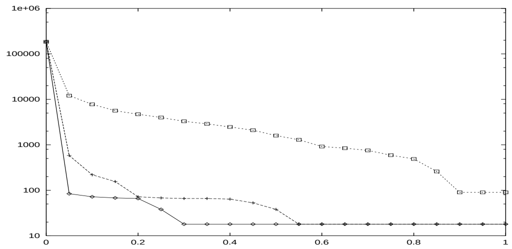
(a) geographical data



(b) numerical data

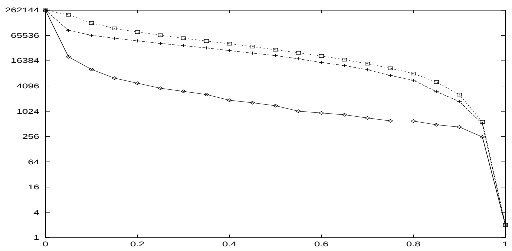


(c) isosurface

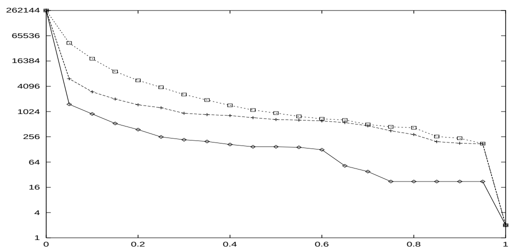


(d) slice

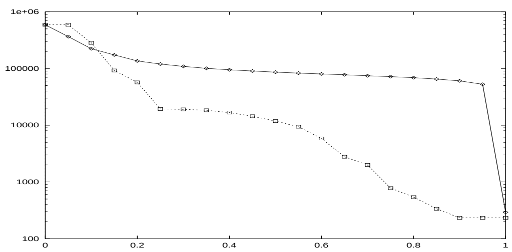
Fig. 1. Error indicators $\bar{\eta}_1$ (dashed), $\bar{\eta}_2$ (solid) and $\bar{\eta}_\infty$ (dotted) based on the different local L^1, L^2 and L^∞ norms of the hierarchical offset are compared, concerning the count of visited simplices for varying threshold values.



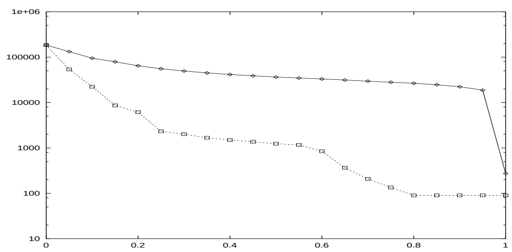
(a) geographical data



(b) numerical data

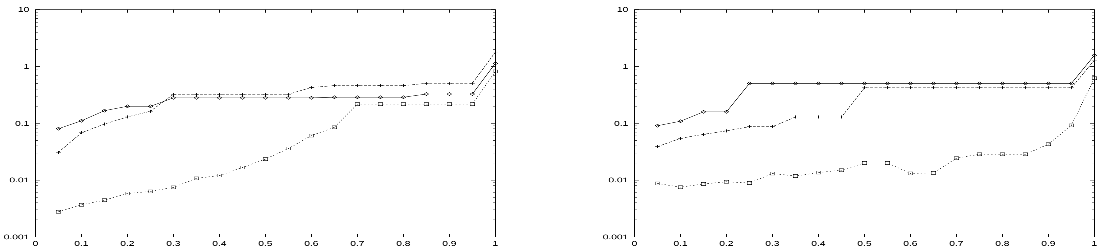


(c) isosurface



(d) slice

Fig. 2. The visited cell count is compared for the error indicators $\bar{\eta}_{1,\infty}$(solid), $\bar{\eta}_N$(dotted), and $\bar{\eta}_{N,e}$(dashed, only in 2D above) respectively.



(a) geographical data (b) numerical data

Fig. 3. Efficiency as function of threshold value for local $\bar{\eta}_\infty$–error (dotted), $\bar{\eta}_1$–error (solid) and local $\bar{\eta}_2$–error (dashed)

References

[1] E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *IMPACT of Computing in Science and Engineering*, 3:181–191, 1991.

[2] E.A. Bier, M.C. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: the see–through interface. *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 73–80, 1993.

[3] A. Certain, J. Popović, T. DeRose, T. Duchamp, D. Salesin, and W. Stuetzle. Interactive multiresolution surface viewing. *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 91–98, 1996.

[4] P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno. Multiresolution representation and visualization of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):352–369, 1997.

[5] P. Cignoni, C. Montani, and R. Scopigno. MagicSphere: an insight tool for 3D data visualization. *Computer Graphics Forum*, 13(3):317–328, 1994.

[6] M.P. Do Carmo. *Differential Forms and Applications.* Springer, 1994.

[7] T. Gerstner. Adaptive hierarchical methods for landscape representation and analysis. In S. Hergarten and H.-J. Neugebauer, editors, *Process Modelling and Landform Evolution*, Lecture Notes in Earth Sciences 78, pages 75–92. Springer, 1998.

[8] T. Gerstner, M. Rumpf, and U. Weikard. A comparison of error indicators for multilevel visualization on nested grids. In E. Gröller, H. Löffelmann, and W. Ribarsky, editors, *Data Visualization '99*, pages 199–211. Springer, 1999.

[9] M.H. Gross and R.G. Staadt. Fast multiresolution surface meshing. In *Proceedings IEEE Visualization '95*, pages 135–142. IEEE Computer Society Press, 1995.

[10] R. Grosso, C. Lürig, and T. Ertl. The multilevel finite element method for adaptive mesh optimization and visualization of volume data. In *Proceedings IEEE Visualization '97*, pages 387–394. IEEE Computer Society Press, 1997.

[11] W. Hackbusch. *Multi–Grid Methods and Applications.* Springer, Berlin, 1985.

[12] H. Hoppe. Progressive meshes. *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 99–108, 1996.

[13] R. Klein, G. Liebich, and W. Straßer. Mesh reduction with error control. In *Proceedings IEEE Visualization '96*, pages 311–318. IEEE Computer Society Press, 1996.

[14] Y. Livnat, H.W. Shen, and C.R. Johnson. A near optimal isosurface extraction algorithm using the span space. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):73–83, 1996.

[15] J. Maubach. Local bisection refinement for n-simplicial grids generated by reflection. *SIAM J. Sci. Comp.*, 16:210–227, 1995.

[16] W.F. Mitchell. A comparison of adaptive refinement techniques for elliptic problems. *ACM Transactions on Mathematical Software*, 15(4):326–347, 1989.

[17] R. Neubauer, M. Ohlberger, M. Rumpf, and R. Schwörer. Efficient visualization of large scale data on hierarchical meshes. In W. Lefer and M. Grave, editors, *Visualization in Scientific Computing '97*, pages 125–137. Springer, 1997.

[18] M. Ohlberger and M. Rumpf. Adaptive projection methods in multiresolutional scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 4(4):74–94, 1998.

[19] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[20] M.C. Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal on Numerical Methods in Engineering*, 20:745–756, 1984.

[21] W.J. Schroeder, J.A. Zarge, and W.A. Lorensen. Decimation of triangle meshes. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 65–70, 1992.

[22] R. Shekhar, E. Fayyad, R. Yagel, and J.F. Cornhill. Octree-based decimation of marching cubes surfaces. In *Proceedings IEEE Visualization '96*, pages 335–244. IEEE Computer Society Press, 1996.

[23] U. Weikard. Finite-element-methoden für die Cahn-Hilliard-gleichung unter einbeziehung elastischer materialeigenschaften. Master's thesis, Institut für Angewandte Mathematik, Universität Bonn, 1998.

[24] J.P. Wilhelms and A. Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 11(3):201–227, 1992.

[25] Y. Zhou, B. Chen, and A. Kaufman. Multiresolution tetrahedral framework for visualizing volume data. In *Proceedings IEEE Visualization '97*, pages 135–142. IEEE Computer Society Press, 1997.

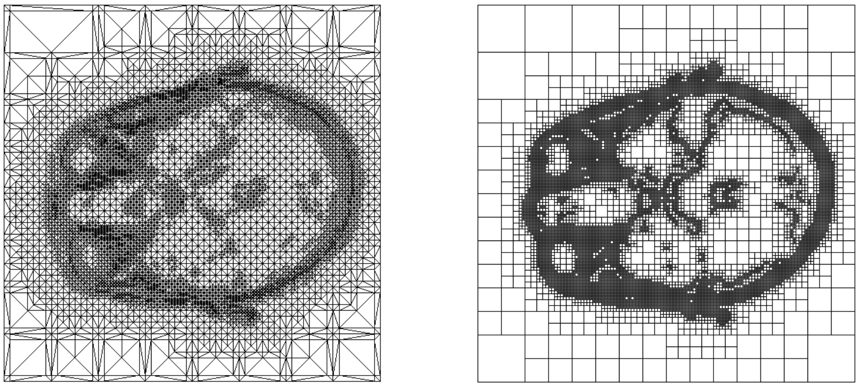


Fig. 4. Adaptive slices through an MRI head data set on a tetrahedral and a hexahedral grid.

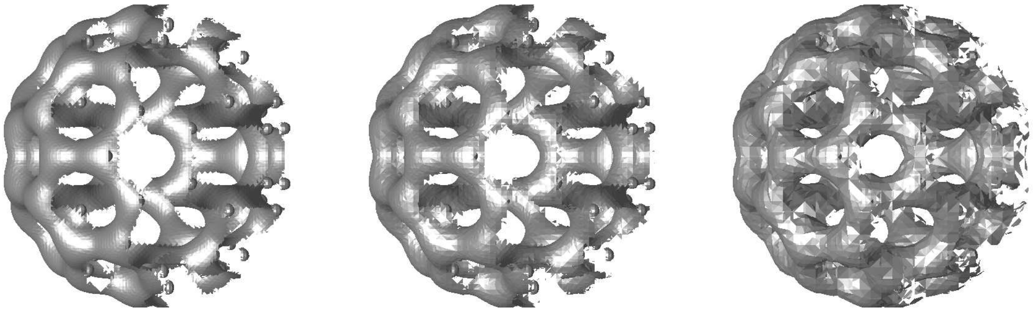


Fig. 5. The effect of multilevel backface culling using the error indicator $\eta^{+}_{1,\infty}$ is shown. The actual viewing direction is from the left. The threshold values are 0.0, 0.1, and 0.4.

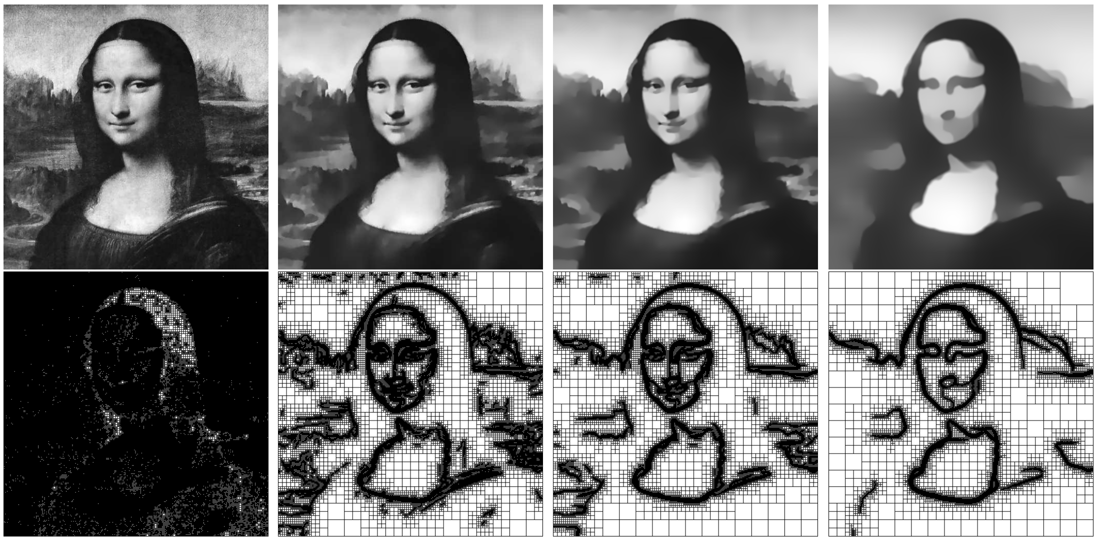
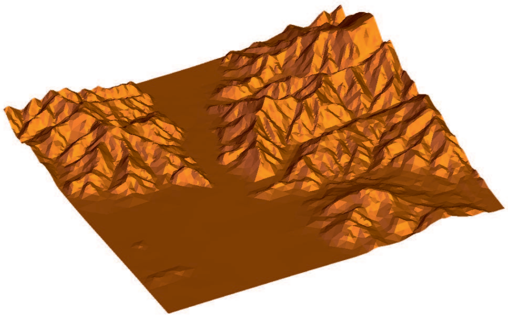
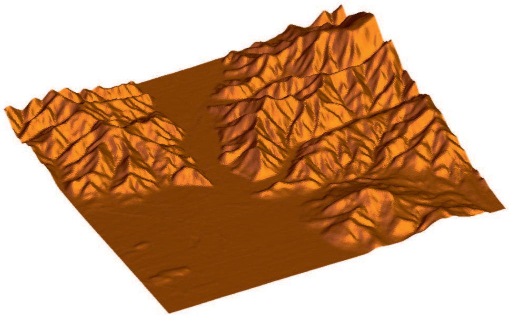


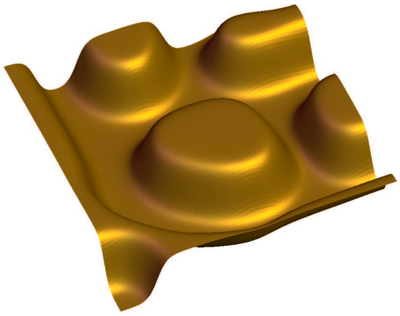
Fig. 6. Anisotropic image smoothing after 0, 10, 40, and 160 time steps. For each time step the smoothed image and the underlying grid is shown.



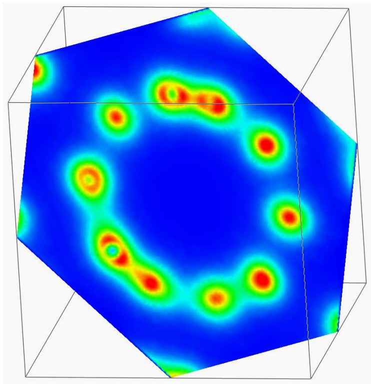
(a) Adaptive projection of the geographical map



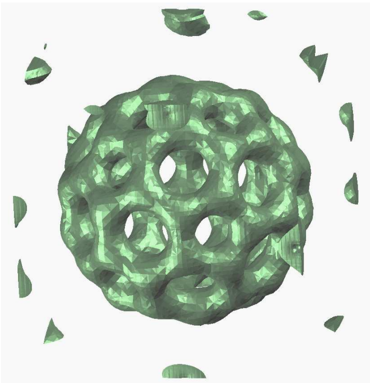
(b) Original data



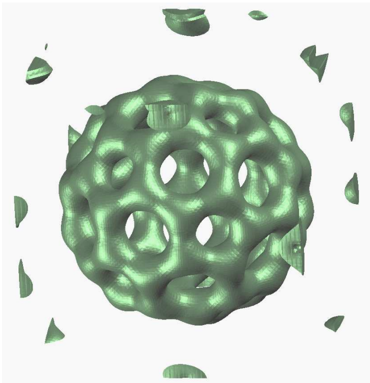
(c) Timestep of Cahn–Hilliard Equation



(d) Color shaded slice of the bucky ball



(e) Adaptive projection of the isosurface



(f) Original data

Fig. 7. Above the graph of a geographic height field (b), its adaptive projection (a) and a timestep of the Cahn–Hilliard–Equation (c) are shown. Of the bucky ball data set we show a color shaded diagonal slice (d), an adaptive projection (e) and a full resolution isosurface (f).