

Regular article

Unstructured tetrahedral mesh quality analysis using an interactive haptic and visual interface

Lisa J.K. Durbeck^{*,**}, Martin Berzins

¹ School of Computing, University of Utah, Salt Lake City, Utah, USA
(<http://www.cs.utah.edu>)

² Computer Studies, University of Leeds, Leeds, UK
(<http://www.scs.leeds.ac.uk>)

Received: 15 June 2000 / Accepted: 16 May 2001

Communicated by: G. Wittum

Abstract. The problem of interactively probing a mesh to determine its quality is described for three-dimensional unstructured tetrahedral meshes. Mesh quality as a function of individual element error is defined for a specific class of problems. The importance of analyzing mesh quality within a geometrical representation of the mesh is discussed. The problems encountered when attempting to visualize the geometric and error information for a visually complex mesh are identified and used to motivate a design for an interactive user interface for mesh quality analysis. The primary intended user of such a system is one who is interested in per-element mesh quality, such as the developer of mesh generating software or the persons charged with generating a good quality mesh for a specific problem; however, it may also be used by end users of meshes to see the main problem areas of the mesh and to compare various available meshing strategies. The interface provides the user with necessary information about element quality in a form which allows the user to isolate “bad” mesh elements and analyze the individual contributions of element shape, orientation, geometric neighborhood, and solution behavior. The availability of this information when combined with a haptic device allows the user to easily identify poor quality mesh regions. A prototype implementation of the interface was constructed and used to examine two meshes in detail. This was done in part reflexively, to determine the feasibility of this approach to mesh quality analysis. It was also done in the interests of our larger goals, to try to determine the main contributors to poor element quality in the two meshes. User analysis of the problem meshes is presented along with visual output from the interface. A formal user study was not performed; however, informal results and timings are used to show the speed and effectiveness of the interface.

1 Introduction

Unstructured triangular and tetrahedral meshes are applied extensively in mechanical engineering, fluid dynamics and

* *Current address:* Cell Matrix Corporation, Blacksburg, VA, USA
(<http://www.cellmatrix.com>)

** Research of this author is partly supported by an NSF Traineeship and the DOE-sponsored Advanced Visualization Technology Center.

scientific computing for solving problems via finite element and finite volume methods. The price of using complex unstructured meshes is that they are quite difficult to analyze for correctness or quality. Various mesh quality metrics have been proposed as suitable for certain problems, based on various aspects of the problem class and the solution techniques used [2, 17, 30]. Current software packages for generating and adapting meshes now generally provide geometrically sound meshes, free of illegal elements, holes, or hanging vertices; however, they cannot absolve researchers of the responsibility for evaluating their meshes. The issue of mesh quality when such meshes are used is both a difficult practical issue and a misunderstood theoretical issue, see Berzins [3] for a discussion of existing metrics and their limitations. Often such metrics are based on purely geometric information such as the edge lengths and volume, for instance the PSUE environment of Weatherill [29], which provides visual information regarding these geometric quantities and statistical information regarding the mesh in both a visual and a report form. Given the lack of understanding of how to construct optimal unstructured meshes it is perhaps not surprising that they are rarely found or even sought. Given these constraints, the contemporary question is not whether a given mesh is optimal, but whether it is of sufficiently good quality to be used for computing a solution. Quality metrics are still an open question, as are the factors determining quality. For meshes of insufficient quality, a pertinent issue is whether and how the mesh can be repaired or improved.

The work described here addresses the theoretical question of quality metrics. It also addresses the practical issue of repairing meshes in that it isolates the local neighborhoods in need of repair. The approach to these questions in this work is to examine the geometry and quality of specific meshes in detail. Of particular interest to us is the relationship between mesh quality and a mesh element’s volume, shape, location within the mesh, vertex location, and orientation. This paper describes the mesh analysis tool designed for this purpose as well as several findings resulting from our observations of specific 16000-element meshes and quality metrics.

Mesh quality analysis may be defined as finding the “worst” elements of the mesh, and understanding why those elements are bad in terms of their shape and the solution behavior. In a complex unstructured mesh this process is made

easier if visualization techniques are employed to identify these elements and their dependence on the solution and the mesh geometry. The method described here also provides immediate insights into local and global geometry.

One critical measure of a finite element mesh is how close the approximate solution computed using the mesh comes to the true solution. This measure is obtained through error estimation. Error is reduced via mesh adaptation, a process in which, depending on the type of adaptation scheme, regions of the mesh are further refined or derefined, vertices are moved, edges or faces are swapped, h -adaptation, p -adaptation, or smoothing is performed, or higher order basis functions are used. The desired result of mesh adaptation is a mesh that allows for as close an approximation of the continuous problem as possible.

Examining mesh geometry and element quality in detail implies establishing an appropriate error metric over the set of elements and then stepping through the entire mesh and examining each element and adjacent neighbors in terms of their geometry and quality. This requires both detailed local information and the ability to relate the local information to a higher level understanding of the rest of the mesh so that factors such as relative volume and shape and location can be analyzed. Interaction and visualization techniques can be applied in concert to help manage the complexity and time-consuming nature of this undertaking.

It is hard to make sense of all the 10^5 or 10^6 mesh elements in a full three dimensional unstructured mesh display, particularly in regions densely packed with small elements, and it is difficult to comprehend fully the myriad of element shapes and volumes. Even for a small unstructured mesh, such as the nine element tetrahedral mesh shown in wire frame in Fig. 1, it becomes difficult to ascribe edges and vertices to elements and therefore to pick out individual elements; the difficulty is further compounded when one or more

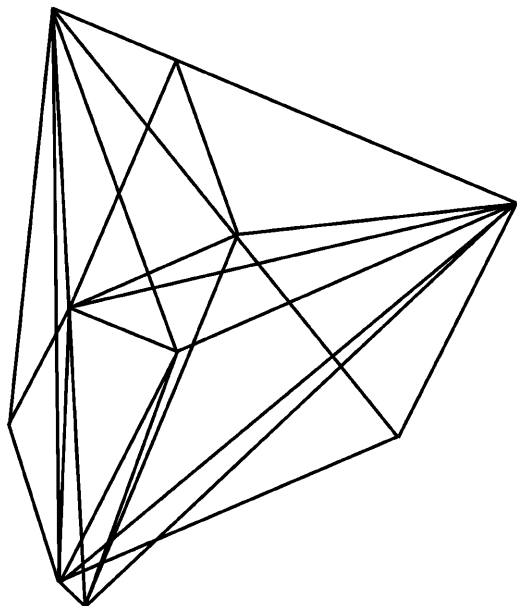


Fig. 1. Nine element mesh subregion displayed in wire frame. Wire frame visualization is not especially beneficial for investigating individual elements and their error: it is difficult to pick out the individual tetrahedra and to integrate this with the separate error information quickly, making comparisons among tetrahedra unreasonably time-consuming

scalar quantities are also displayed. Views of the full mesh, showing the outline of the elements in wire-frame mode or showing each element as a colored solid, are not useful for the investigation of individual mesh elements – or of quantities defined over the set of elements, like error estimates – because the granularity is wrong. Searching a full mesh display for bad elements is not unlike the proverbial search for a needle in a haystack: the sheer number of elements and their contiguity makes detailed investigation of a display containing all of them too time-consuming to be viable. A chief task in designing an interface is to control the visual complexity of the information presented to the user. Whereas in Fig. 1 it is difficult to determine what edges belong to which element, in a full wire frame mesh such as the one shown in Fig. 2, it is not only more difficult to pick out individual tetrahedra, but there is an additional problem of edges occluding one's view of those behind them. In effect, only the edges in sparse regions or at the front edge of the volume are discernable. For the problem at hand, the potential visual complexity is significantly heightened by the need to display additional information about element quality on a per-element basis.

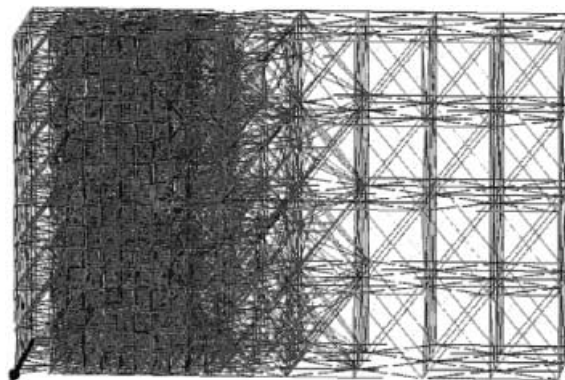


Fig. 2. Mesh for the simple advection example, displayed as a wire frame model. All 16000 elements are displayed; however, no error information is displayed. The area of heavy refinement represents the discontinuity

Unstructured mesh visualization tools that include ways to view information at the element level include Los Alamos National Lab's General Mesh Viewer gmV [18], Advanced Visual Systems Inc's AVS5 [1], which has also been customized into Sandia National Lab's FEAVR and MUSTAFA systems [24], and Lawrence Livermore Lab's MeshTV [16]. The usefulness of interactive and immersive environments in mesh viewing, for activities like finding illegal elements, was also explored by the Advanced Data Visualization and Exploration group at Sandia [23, 25].

Mesh quality analysis has been done to some extent by coloring regions of the mesh according to a mesh quality indicator see [6, 10, 11]. The main problem with using standard visualization techniques for viewing mesh quality indicators is that they are not conducive to viewing information on a per-tetrahedron or per-vertex basis. This is especially evident in unstructured three dimensional meshes. Finding bad tetrahedra or vertices in a visually complex mesh using standard mesh visualization techniques such as wire-frame views with clipping planes is a daunting task.

The goal of this research, then, is to provide an effective user interface to interact with three dimensional visualiza-

tions of mesh error indicators that can quickly cut through the huge amount of information to the key information: how bad is this mesh, where are the worst elements, and what is wrong with them? This user interface builds on some mesh visualization ideas from MeshView, [11], such as clipping surfaces, viewing subsets of elements, and viewing the neighborhood of a specific tetrahedron. MeshView is designed primarily for viewing a variety of metrics, all of which are calculated just from the mesh itself. The intention here is to display information interactively, using both haptic and graphic interfaces tailored to the user's investigation of the causes of poor mesh quality as measured by an error indicator, which is based upon solution and geometry information, see [4].

Such a system might be applied to the improvement of a mesh generation technique and equally to the deduction and analysis of per-element error estimation techniques. Although it is certainly easier for the developer to understand and use the approach, we believe that the simple visual approach to investigating the solution allows end-users to see where there are large errors and perhaps to experiment with different strategies for meshing to see if the errors can be reduced. The diagnostics and their ease of use also provide end-users with an intuitive grasp of how errors are distributed in the mesh.

2 Interface design

The basic design elements are visual feedback, force feedback, and user commands. Visual feedback is provided on a standard 21" desktop monitor. Force feedback is provided by a PHANTOM™, designed and manufactured by SensAble Technologies [19,26], which can apply forces to a pen at the end of a robotic arm. User commands are provided by a combination of a gestural interface, based on "pointing" the pen in three space, and standard GUI interfaces allowing the user to control a set of interface parameters such as the magnitude of the pen forces and the colors used in the display.

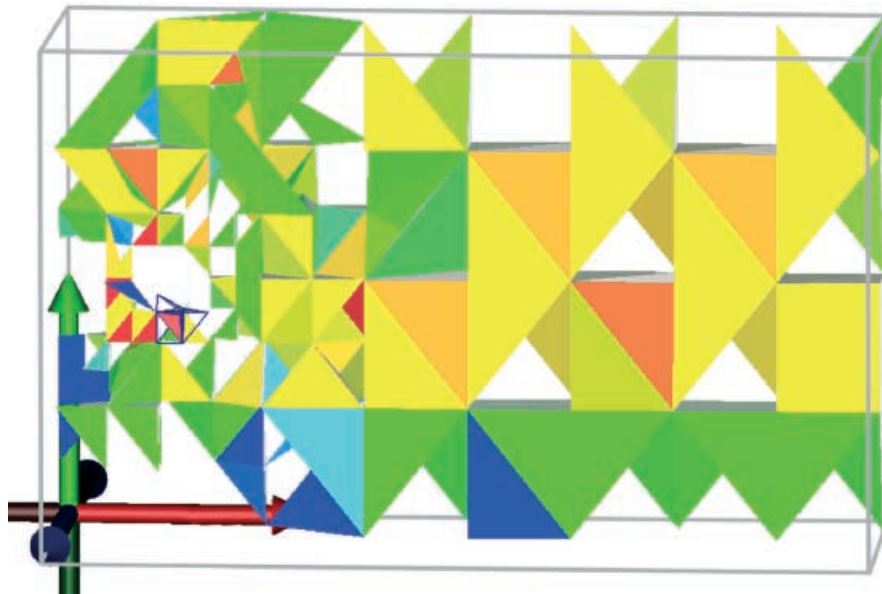


Fig. 3. Example of a use of the global view. It is used here to view a subset of the atmospheric model. The user controls what subset is displayed using either a histogram of error values or a cartesian range setter. The subset is displayed as solid tetrahedra, with lighting and shading effects. The color assigned to each tetrahedron corresponds with its relative error value. In this case, the view was restricted to the mesh elements with nonzero error values. The mesh is also filled in directly surrounding the user's current search point using a solid model of the "current" tetrahedron and wire frame for the adjacent elements. A small white sphere indicating the search point is also drawn. An axis at the origin is provided for orientation during rotations of the view

The chief design requirement was that the interface should provide brevity and clarity to the task of finding the bad mesh elements and deducing the reasons behind their poor quality. This breaks down naturally into two separate tasks, searching the mesh for bad elements and examining the mesh around each bad element. It also entails a third task, a comparison across mesh elements to determine what makes one element good and another bad.

Searching the mesh for bad elements is too time-consuming to be done interactively by the user. The notion of linking the display to user-controlled subset capabilities was introduced by Gitlin [10, 11]. The user specifies the criteria for inclusion or exclusion, and the software does the search and displays the resulting subset. This technique is used here to permit the user to define a set of mesh elements based on relative error value. It is incorporated into a particular view of the mesh, called the global view, to allow the user to restrict the display to a subset of mesh elements. A typical use of this feature is to display only the worst mesh elements in an otherwise empty mesh volume. The user then dynamically fills in parts of the volume gesturally, moving the pen to cause regions of the mesh to be filled in. A sample of the global view is provided in Fig. 3.

With the task of finding the bad elements thus minimized, one of the remaining tasks the user faces is examining the mesh around bad elements to determine the factors affecting the poor quality. Examination of the mesh around bad elements is facilitated by a local view of the mesh subregion that contains relevant information presented in a way that aids the user's inquiry. Figure 4 shows several examples of the local view.

The final task the user faces is developing a higher level understanding of what factors are relevant. This appears to again be a search problem, but one with potentially many iterations on search criteria during the process of elimination of irrelevant factors. Several rudimentary comparison aids were incorporated into the user interface to allow comparison of scalar values defined over the set of elements. Comparative information about the scalar field is displayed graphically

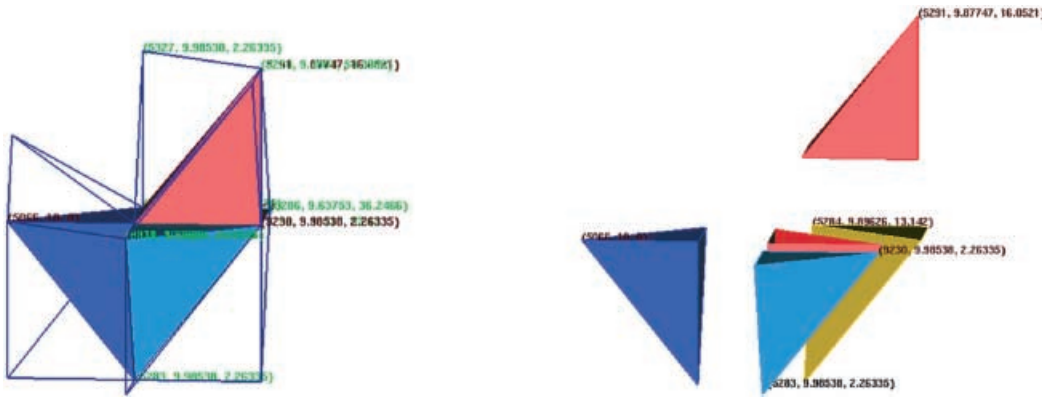


Fig. 4. Two examples of uses of the local view. The same region of the mesh is displayed in two distinct viewing modes. The local view is centered about a tetrahedron chosen by the user. That center tetrahedron is displayed as a shaded solid, colored by error value. Its immediate neighbors are also displayed as shaded, colored solids. The neighbors can be colored using several color schemes useful for analysis. One more level of face-neighbor adjacency, of lower relevance, is also displayed in wire frame. The view can be zoomed in or out, rotated about the central element, and negative space can be created between the elements so that their shapes and relationships can be seen from multiple vantage points. The graphical representations of each element can be annotated with textual information relevant to the user's analysis. In these images, the element number, error value, and solution value are displayed as a triplet attached to an element node

via color mapping, allowing the user to rank mesh elements' scalars by color. This information is also presented haptically via a mapping of the range of scalar values to the presence or absence of texture. Texture force feedback is available in three dimensions over the workspace of the PHANToM™ device, and can increase the overall bandwidth of information the user receives about relative scalar values by its deployment over a larger set of elements than can be displayed on the 2D visual display, where the more elements displayed, the more they overlap, obscuring each other and requiring rotations of the viewpoint. Texture was incorporated into the design to provide two pieces of information: it duplicated relative error value information shown graphically with color, and it provided resolution of the three dimensional location of bad elements not possible in the 2D visual display.

An additional feature was added to promote the user's higher level understanding of the relationship of scalar value (e.g., error value) to element position within the mesh. A useful sequence of subset displays is folded into an automated visualization, called "evaporation," that the user can perform on a mesh to gain a higher level understanding of the distribution of element error in the mesh. The visualization uses time as a display, applying an increasingly restrictive high-pass filter to the mesh elements at each time step. The result is an animated image in which the mesh appears to be eroded down to its worst elements. This visualization technique is described more completely in [7].

2.1 Interface implementation and use

A typical user session with a new mesh or error metric began with the high-level view of the distribution of error in the mesh (evaporation), followed by a presentation of the 10–20 worst mesh elements in the global view, color coded by utilizing the full color map to distinguish them from each other. The user then began to investigate the bad mesh elements one by one. The user located a problem element using the sparse global view, navigated to it and clicked on it to populate the local display, and then focused his/her attention on the information presented in the local view in order to deduce the

reasons behind the high error value. After analyzing one bad element, the user refocused his/her attention on the global view in order to pick a new bad element for review.

Onscreen, the local view displays the mesh elements significantly larger than they appear in the global view. The views are shown in separate windows which the user can position onscreen, as shown in Fig. 5, a snapshot of a user's screen. The relative volumes and shapes of the elements are readily apparent with these simultaneous global and local views, and the user interaction is sped up by obviating the need to zoom in and out.

Figure 5 shows what the screen typically looks like while the application is running. The SCIRun dataflow window is running in the background, and the global and local windows are displayed as large as possible. The size, location, and orientation of the graphics were preset so that they correspond to the location and orientation of the PHANToM™ workspace's principal axes. Up is up, left is left, and so on. For a right-handed user, the PHANToM™ is placed to the right of the monitor. The graphics are considered to be a simple translation, no rotation, of the haptic workspace. Most users of the system had no trouble understanding and using the correspondence between the displays, although several who were new to the PHANToM™ device described a learning period before they could use the PHANToM™ to move around the data volume to specific locations effectively.

An inverse rainbow coloring scheme (highest to lowest scalar value: red, orange, yellow, green, blue) is used. The color map contains 16 distinct colors rather than a smoothly graded function. These colors are intended to be easily distinguishable from each other. This set-level coloring allows the user to reduce the task to say "looking for all the red tetrahedra." Colors are remapped whenever the subset criteria are changed. This allows the user to regrade the coloring to highlight differences between a small set of tetrahedra, or to get a more coarse-grained indication of the relative value within a larger subset of the scalar field.

Force feedback is used to provide textures for the bad mesh elements. Good elements have no texture, while bad elements have a ridged texture that is generated by applying

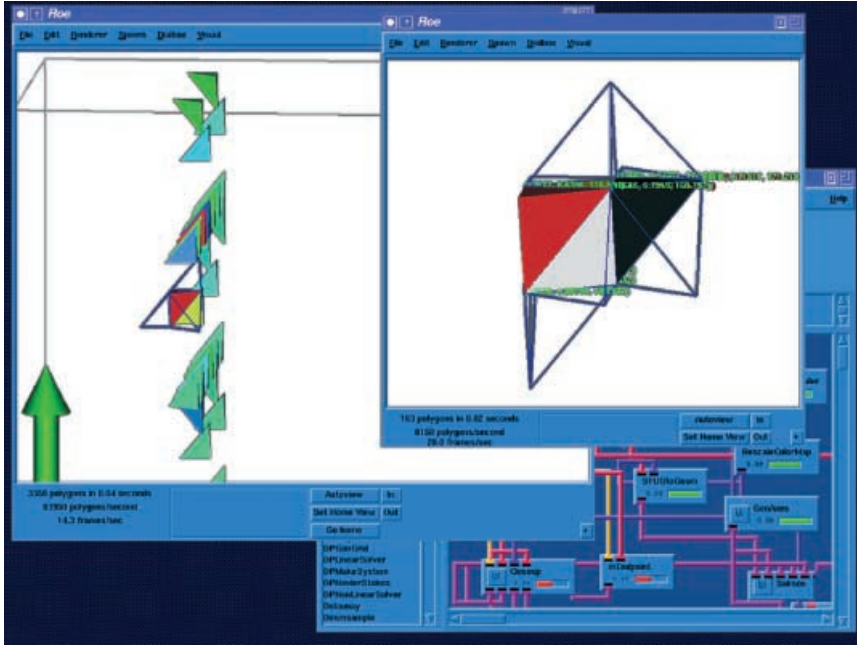


Fig. 5. The complete onscreen display. The global and local views occupy two separate X windows. The SCIRun user interface, used to start and direct the mesh analysis, is visible in the *lower right*

a small sinusoidal force field in the x - y plane. Details about the haptic interaction are provided in [9].

A prototype of the design was created. Part of the user interface was implemented within the SCIRun problem-solving environment as SCIRun modules; the PHANToMTM was controlled via a separate executable, and communication between the two executables was implemented with sockets. The full text describing the interface design, implementation, and evaluation can be found in [8].

The described user interface was used to investigate error values calculated for the two h -adapted unstructured tetrahedral finite element meshes within the problem class.

3 Numerical experiments

3.1 Problem class and solution techniques

In order to illustrate the interface for improving mesh quality analysis a simple finite volume solver will be applied to the class of problems given by the equation

$$U_t + [F(U)]_x + [G(U)]_y + [H(U)]_z = S(U) \quad (1)$$

for three space dimensions (x, y, z) and with time t . The variable $U(x, y, z, t)$ is the vector of conserved variables and the vector functions $F(U)$, $G(U)$ and $H(U)$ are the analytic fluxes and $S(U)$ is a source term. On account of the need to admit discontinuous solutions such as shock waves and contact surfaces, it should be understood that we investigate weak solutions of the integral form of these equations:

$$\begin{aligned} & \frac{\partial}{\partial t} \iiint_V U d\tau + \iint_{\partial V} (F\mathbf{i} + G\mathbf{j} + H\mathbf{k}) \cdot d\mathbf{S} \\ & = \iiint_V S(U) d\tau. \end{aligned} \quad (2)$$

Here V represents some fixed control volume with volume element $d\tau$ and surface ∂V with directed area element $d\mathbf{S}$, and where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the cartesian unit basis vectors. The numerical method employed is a first order accurate, conservative cell-centered finite volume scheme using Godunov's Riemann Problem (RP) approach. The numerical solution in some element i at time t^n is denoted by U_i^n , and is understood to be an approximation to the exact element averaged volume integral of the solution, that is:

$$U_i^n \approx \frac{1}{V_i} \iiint_{V_i} U(x, y, z, t^n) d\tau \quad (3)$$

where V_i is the volume of element i , and is usually regarded as being valued at the element centroid for cell centered schemes. Application of the integral conservation law (2) shows that the numerical solution at the next time level t^{n+1} may be written:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_k} \sum_{k=0}^3 A_k \mathbf{F}_k \cdot \mathbf{n}_k \quad (4)$$

where the sum is over the k faces of the element i . The \mathbf{n}_k are the outward face unit normal vectors and A_k the face areas. The fluxes \mathbf{F}_k represent the numerical flux function for each element face, termed the element face fluxes, and are determined by the scheme. In the case of the well known Godunov scheme these element face numerical fluxes are constructed from the solution of the local element Riemann Problem (RP) at each element face, see [27], based on the values which (U_l, U_m) represents the left and right element data values on either side of a particular face.

3.2 Adaptive meshes and error estimates

The h -adaptive meshes for this scheme were created using the TETRAD mesh adaptation software developed by [27].

TETRAD output was previously partially integrated with SCIRun so that the resulting meshes and solutions could be visualized [12]. SCIRun is a problem-solving environment for scientific computing [20, 21].

The development of reliable error estimates for finite volume unstructured mesh solvers is an ongoing activity. In this work we make use of two independent ideas. The first is that although for relatively simple problems there are reliable estimates such as those of Kroner and Ohlberger [15] there are no such estimates for problems with complex source terms. The consequence of this is that we are forced to rely instead on local error indicators such as those proposed by Berzins [3]. The essential idea being to estimate the local growth in time of the spatial error. This estimate assumes that the spatial error is zero at any one time and then attempts to estimate its growth over the next timestep. For problems without source terms the estimate of Kroner and Ohlberger may be adapted to estimate this error. Let $\hat{e}(t)$ be the local in time spatial error computed on a timestep then combining the estimates of Corollary(2.14) of [15] and the ideas of Berzins [3, 4] gives

$$\iiint_V \hat{e} d\tau = a \delta t h^2 Q + 2\sqrt{bc} \delta t h^2 Q \quad (5)$$

where a, b, c are constants, see [15] and for an evenly spaced mesh with spacing h and timestep δt the value of Q is

$$Q = \sum_{j \in NT} h |u_j^{n+1} - u_j^n| + L \sum_{E \in NT} (\delta t + h) |u_j^n - u_l^n|$$

where u_j^n is the solution value associated with the j th tetrahedron out of a mesh of NT tetrahedra with edges $E \in NT$ at time t_n . The important feature of this error estimator is that apart from the constants the only solution information used consists of solution jumps across faces i.e. $u_j^n - u_l^n$ and solution changes in time $u_j^{n+1} - u_j^n$ on a particular tetrahedron. For each face of each tetrahedron in the mesh, TETRAD's adaptation code computes the flux across the face by comparing the solution for this tetrahedron to that of the tetrahedron which shares this face (called a face neighbor). The difference indicates the flux across the face. The actual error indicator used on the i th element is then given by

$$E_i^n = \sum_{j \in NT_i} |u_j^n - u_l^n| \quad (6)$$

where u_j^n is the solution value associated with the j th tetrahedron out of a mesh of NT_i tetrahedra whose faces adjoin the faces of tetrahedron i at time t_n . The important feature of this error estimator is that it only takes account of the spatial part of the error estimator and only takes into account the local growth in the spatial error at the present time. For any individual tetrahedron it is thus the individual solution jumps that are important. The error value assigned to the entire tetrahedron is thus a simple function of the fluxes across the faces. In using this estimator here we have used only the spatial component of the error and excluded the temporal terms, by using a time step so that the spatial error dominates, following Berzins [3]. The value calculated using this error indicator will be referred to in the following discussion as the error value. The set of error values defined over the set of mesh elements is referred to as the scalar field.

3.3 A simple advection equation example

The first mesh investigated was an advection of a simple one-dimensional discontinuity in a three dimensional channel. The equation being solved is

$$U_t + U_x = 0 \quad (7)$$

A typical example of a three dimensional unstructured mesh at a particular time step is shown in Fig. 2. The mesh is shown in wire frame, with all the nodes and their attachments shown. Although it is impossible to see all the elements, such a depiction allows one to get a rough idea of the resolution. The higher resolution area is the current location of the edge of the wave.

As time progresses, the high value propagates from left to right throughout the region. A typical problem introduced by numerical methods is the blurring of the discontinuity. Unstructured adaptive mesh solvers try to use a much finer resolution mesh at the boundary between the high and low values in order to capture the discontinuity. The only substantial change required to the error estimation procedure is that each face difference is divided by the face average in an attempt to preserve the magnitude of the difference irrespective of the magnitude of the solution values. For instance, a difference of 8 between the numbers 108 and 100 is not as significant as a difference of 8 between the numbers 18 and 10, and the error indicator calculation reflects this. In other words, the face values are (locally) normalized so that the magnitude of the face error values does not overshadow their difference.

4 Initial results

4.1 Success of design

The first question is whether the user interface design is satisfactory for our endeavor, that is, does it provide us with the ability to analyse the relative contributions of local geometry and solution behavior to poor element quality. A formal evaluation of this interface was not determined to be necessary to evaluate whether the system design provided an interface that furthered the authors' understanding of element quality and geometrical considerations. However, informal qualitative and quantitative evidence was gathered from a slightly larger pool of potential users, to determine whether the prototype system was usable. Of interest is the user's ability to understand and make use of the displays, as well as whether the interactivity exhibits sluggishness.

Nine volunteers tested the system. The volunteers were either members of the SCI Institute or students from an introductory Virtual Reality class within the Computer Science department at the University of Utah. Novice users required a training period on the system ranging from 10 to 30 minutes, after which they reported they were able to make use of the haptic and visual information and interface presented. Most users had prior experience using the SCIRun software system, and all but one had limited prior experience using the PHANToMTM haptic device as well. Four users had also previously used the combined SCIRun/PHANToMTM flow field display [9]. After the training period, users familiar with the

research domain used the system to investigate element quality in the sample meshes.

In Fig. 6, the global view is used to isolate the 11 worst elements in the simple advection mesh. Imposed on top of the global view is an ordering of the mesh elements by quality that the user determined by relative element color. The color map used is displayed on the left of the figure. The user's ranking was shown to be correct by comparison to the actual error values of the elements. The user's ability to isolate bad mesh elements using the graphical display is evidenced by the fact that a user can generate the image shown in Fig. 6 and successfully rank the elements by relative error value. This result provides evidence that users can cause the system to display the tetrahedra they are interested in, and that users can reduce these to an entirely manageable, comparable number using the system's subset definition interface.

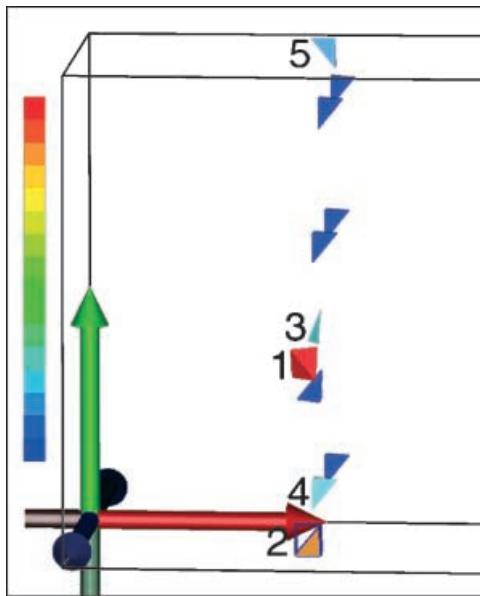


Fig. 6. The worst elements in the simple advection mesh, as ordered by a user using element color. The numerals were added to represent the ordering the user assigned. For clarity the original gray background is shown as white, and the right side of the mesh volume is truncated

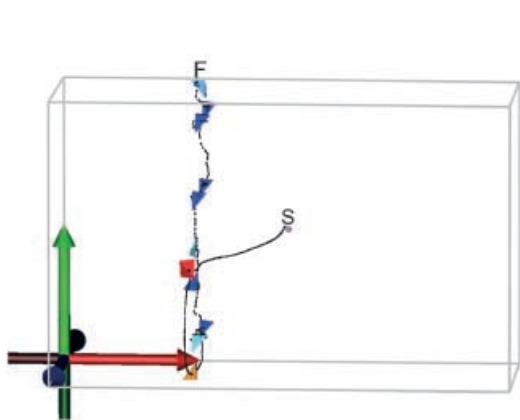


Fig. 7. The path a user took to bring the tip of the PHANTOM™ pen to each displayed element, shown in profile from the front and also from the side. The path is displayed as a series of black dots, its beginning marked 'S' and end marked 'F'. The user was asked to traverse the volume from the given starting point and to contact each element in the image once. The path was sampled, recorded, and then displayed in place

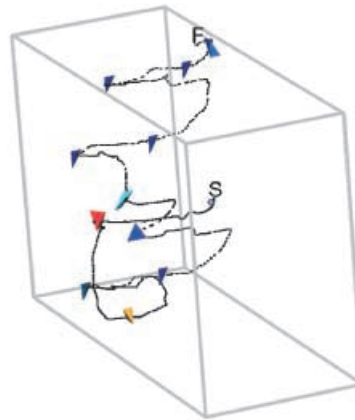
Figure 7 shows the path a trained user traced with the PHANTOM™ pen when asked to navigate to each element in the display. This navigation skill is necessary to populate the local view for analysis of a particular element. The path was generated by adding points to the display for each (x, y, z) position to which the user brought the endpoint. This shows that, with the information and pen-based gestural interfaces provided, the user can interpret and navigate the graphical and haptic displays. There does not appear to be much confusion, as evidenced by the fact that the user took a fairly efficient path from one to the other. This result also provides information about the navigation techniques used by users trained on the system. One new user reported difficulty in attaining this skill.

It is important to add that, although the authors considered ourselves to be quite familiar with the sample meshes, the system described here permitted us a much better and quicker visual insight into the meshing and mesh quality than we had before using the system.

4.2 Interactive rates

Interactive rates are necessary for this interface design. Experiments to determine the refresh rates for the visual and haptic displays were performed on a 195 MHz SGI Octane with two R10 000 processors. The test mesh contained 16 000 elements. The same endpoint path and rate was used for all experiments described in this section. Use of a fixed path negates the variability in rates which would otherwise result from algorithmic dependencies on location within the mesh and allows separate experiments to be compared. The same configuration of the dataflow diagram and the same compiler flags were also used across all experiments. Specific details about the system used for these experiments is provided in the table below. The machine was dedicated solely to the experiment.

Ten trial runs were performed, with a total of 50 000 cycle time measurements (5000 obtained each run). A haptic refresh cycle is defined as the time taken for one complete update of the haptic forces and is measured by the frequency with which the haptic loop was executed. A graphic refresh



cycle is defined similarly and captured by inserting a loop into the graphical update of the largest graphical object, thereby turning the normally event-based SCIRun graphical updates into a loop that could be timed while the experimental user session was run.

The first 100 cycle times and the last cycle time for each run were thrown out to avoid startup and shutdown effects. For the remaining data, the largest and smallest 10% of the numbers was also thrown out to decrease the variability. The average haptic refresh rate for this mesh, using a fixed endpoint path and rate, and using the volume bricking location scheme described in [8], was 1367 Hz. The average cycle time was 731 μ s, with a standard deviation of 470 μ s. In particular, no trial ran under 1 kHz, the required refresh rate for the device. This is evidence that the prototype implementation can read positions at the haptic refresh rate specified for the PHANToM™ device. The average graphic update rate for the same set of conditions is 455 Hz, well above the 30 Hz refresh rate required by human vision. The average cycle time for the graphics was 2197 μ s, with a standard deviation of 1136 μ s. Note that in SCIRun, the graphical scene is constructed from multiple independent objects produced by any number of distinct modules and provided to the graphical viewer module. Thus, these results do not tell the overall graphical update rate for all graphical objects: rather, these results can be taken to indicate that, during normal system operation (i.e., no parts of the system were suppressed or deactivated during timing), for the largest and most time-intensive of the graphical objects, the graphical refresh rate is more than sufficient. The high standard deviation for both the haptic and graphical update rates is attributable to timeouts for thread management and resource sharing within SCIRun (version 5/98). Although the system speediness provides a large margin for error, it cannot guarantee the consistent update rates of a real-time controller, and this would be a consideration in the construction of future versions of the mesh quality interface.

It is also noteworthy to determine how often the user is provided with the wrong information on account of system

lag. Given the average haptic update rate and the tetrahedron volumes from the experiment described above, it can be determined that the frequency of applying a texture to the wrong point in space (i.e., “drawing outside the lines”), is 0.4% for the average-sized tetrahedron, and 0.5% for the smallest tetrahedron (i.e., the worst case), corresponding to a position error of 0.037 mm at a hand movement rate of 50 mm/s.

4.3 Scalability

The degree to which the system design scales to handle larger meshes was also investigated. Tetrahedral meshes were generated by varying the resolution of a structured grid to construct five similar tetrahedral meshes containing approximately 1000, 10 000, 50 000, 100 000, and 500 000 elements. Beyond 500 000, the memory limits of the target platform, the SGI Octane described above, became problematic and limited the mesh sizes possible for the test. The approximate refresh rate for the entire system was obtained by timing the slowest part of the system, the redrawing of the contents of the global view. The maximum refresh rate for this slowest piece was obtained by setting the system up in a continual loop that redrew the global view as fast as possible. Mesh size was varied from 10 000 to 500 000, but the subset size displayed in the global view was kept constant at 84 elements. One trial run was performed for each of the five meshes, and 1000 cycle time measurements were collected from each run. The first 20 data points were thrown out to avoid startup effects; the largest and smallest 10% of the data were thrown out before analysis to decrease the variability. The refresh rate for the global view held constant at 60 Hz average cycle rate for each mesh size. The average cycle time was 16.7 ms, with a standard deviation of 55 μ s. The constant refresh rate observed for meshes ranging from 10 000 to 500 000 indicates that the refresh rate is directly proportional to the subset size, not the mesh size, within this SCIRun-based user interface implementation. This is not surprising, given that the system was designed to take advantage of the feature of SCIRun that modules are executed only when needed. This result suggests that the design is highly scalable: users can control the size of the subset, and refresh rate is proportional to the subset size the user picks.

The system spends time up front running through the mesh to obtain the subset; however, a new subset is calculated once or twice per session, and any user interaction with the subset in the global view is then relatively fast. This aspect of the design appears to be critical to interactive rates: comparative refresh information was obtained for wire frame views of the mesh, a typical mesh visualization technique. Under oth-

Table 1. System setup for frame rate determination

System Component	Details
Computer Model	SGI Octane
Operating System	Irix 6.5.3
CPUs	dual 195 MHz IP30 Processors
CPU type	MIPS R10000, Chip Revision 2.7
FPU type	MIPS R10000, Floating Point Chip Revision 0.0
Main Memory Size	128 Mbytes
Instruction Cache Size	32 Kbytes
Data Cache Size	32 Kbytes
Secondary Unified Cache Size	1 Mbyte
Graphics Board	IMPACTSR, MXI, resolution 1280 × 1024, 72.24 Hz
Bus Type	SCSI controller Version QL1040B revision 2
Haptic Device	SensAble PHANToM™, Model Classic 1.5, with high-resolution stylus encoders
SensAble Card Type	PCI
SensAble libraries	GHOST SDK version 2.0
C++ Compiler	MIPSpro Compilers, Version 7.2.1.3m
SCIRun version	source tree taken from 5/98

Table 2. Refresh rate as a function of Mesh Size Mesh size, Subset size, Global View Display frame rate, Wire Frame Display frame rate, and Subset construction time

Mesh	Subset	G.V. (Hz)	W.F. (Hz)	Subset (s)
1000	84	NR	5	1/5 s
10 000	84	60	2	1/2 s
50 000	84	60	1	1 s
100 000	84	60	1	1 s
500 000	84	60	NR	N/A

erwise identical testing conditions, the refresh rates for wire frame views were noninteractive, even for small meshes. The results for both the Global View experiments and the wire frame experiments are tabularized below. The estimated time spent to obtain the subset is also presented.

The time spent to obtain the subset is estimated based on the assumption that the time to construct the subset consists of one read through the entire mesh and one display of the subset. The display of the subset is known from the Global View update cycle rate, and the read through the entire mesh is no longer, and probably shorter, than the update cycle rate for the wire frame display, which does a more complex set of operations for each mesh element than does the subset calculation module. Therefore, the subset construction can be conservatively estimated as the time taken for one wire frame display cycle plus one subset display cycle. This number is on par with Wire Frame display rates and grows with mesh size. It is encountered only when the system initializes and whenever the subset is changed; once this penalty has been paid, the update rate for the Global View again applies.

Note that it is important to implement the Evaporation visualization technique such that the frame rate remains interactive as mesh size grows. No special effort was made to do so in the present prototype system, since the meshes we were working with were interactively displayed using the given prototype on the given machine. However, because the evaporation presents static information, the sequences can be computed offline and simply displayed in order to achieve interactive rates.

4.4 Analysis of simple advection example

The interface was used to analyse the meshes generated for the simple advection example. The error formula in (6) was used. The performance of this error estimator is examined separately [5]. Given the error formula in (6) the spatial distribution of error was determined by application of the evaporation technique of [7], i.e., successive applications of an increasingly restrictive high-pass filter. Not surprisingly, error was concentrated along the area adapted to represent the advancing edge. The worst elements in the mesh are displayed in Fig. 6. They appear to be roughly aligned with one another and are located along the leading edge of the heavily refined mesh area of Fig. 2.

The worst tetrahedron in the mesh, denoted by Tetrahedron 1, is singled out for analysis. Its neighborhood is shown in Fig. 8. The view is centered about Tetrahedron 1. As in-

dicated by the colors and annotations in the figure, the error value for this tetrahedron is quite high, the highest in the entire mesh, while its neighbors' error values are zero. The fact that all surrounding tetrahedra have low error values relative to Tetrahedron 1 implies that the poor quality of Tetrahedron 1 is a function of the tetrahedron's shape and its orientation with respect to the discontinuity.

In Fig. 8 the tetrahedron appears to be wedge-shaped. Rotations of the viewpoint are provided in Fig. 9, which shows the same neighborhood from multiple angles or vantage points. An axis is provided to indicate the orientation of the image. The orientation of a tetrahedron is described relative to the direction of flux. In general, faces which run parallel to the flux are "best" because they introduce no error, while ones that are broadside to the flux are "worst" because they introduce large errors into the finite element or finite volume method. Wide faces tend to spread or diffuse values, which lowers the accuracy of the numerical solution. The orientation of a face is defined as its dot product with the direction of flux. The direction of flux for this mesh is strictly along the x axis. The axes are shown within the images in Fig. 9, indicating the orientation of the worst element with respect to the x axis. The images show that Tetrahedron 1 has two faces which are somewhat close to perpendicular to the x axis, face 1 shown in the middle image is wide in the $y-z$ plane and slender along x , and face 2 visible in the right image is similar.

It appears that the error value for this tetrahedron is high as a result of these two faces. The information displayed in the local view indicates that the worst tetrahedron has two relatively large faces whose dot product with the direction of flux is relatively low. The two main contributors to its high error value seem to be the orientation of the element, which causes two faces to be close to perpendicular to the flux, and the wedge shape of the element, which causes these two faces to be relatively wide.

5 Problem 2: Atmospheric diffusion model

A more realistic example to illustrate the difficulties in investigating mesh quality is the following three dimensional advection reaction problem, which is taken from a model of atmospheric dispersion from a power station plume – a concentrated source of NO_x emissions [28]. The photo-chemical reaction of this NO_x with polluted air leads to the generation of ozone at large distances downwind from the source. An accurate description of the distribution of pollutant concentra-

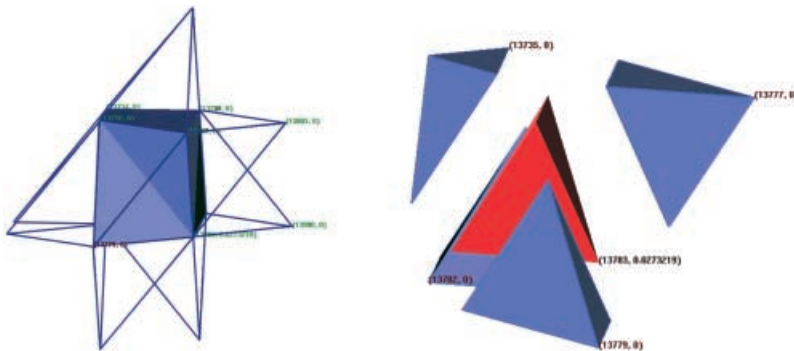


Fig. 8. Local view of the simple advection mesh around Tetrahedron 1 in two distinct views. Color indicates error value, blue lowest and red highest. The *lefthand image* shows the geometry of the mesh region and two levels of face-neighbor adjacency, the outermost in *wire frame* and the inner as *solids*. Tetrahedron 1 is not visible in this view; however, it is visible in the *righthand image* where the elements have been exploded outward from the center. The wedge shape of Tetrahedron 1 is apparent, as is the large jump in error value indicated by color

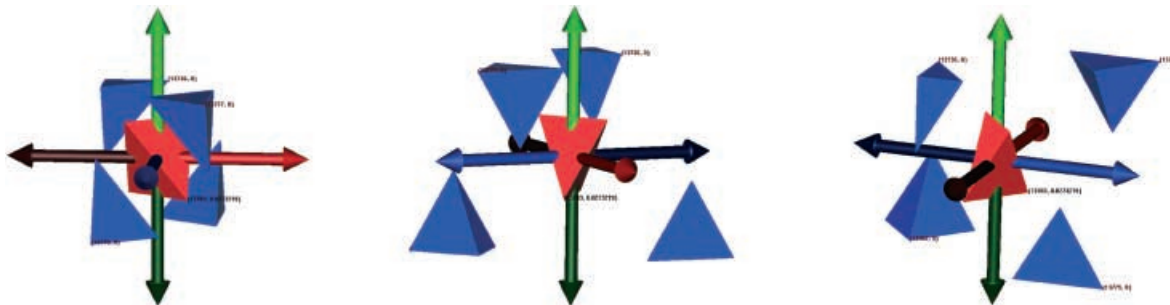


Fig. 9. Rotations of the local view for Tetrahedron 1. Axes added to provide orientation ($x+$ bright red and horizontal, $y+$ bright green and vertical, $z+$ bright blue and out of the plane). The positive x axis is the direction of flux and is shown in all pictures as the horizontal, red axis. The middle view is a -35 degree rotation of the left view about the y axis, and the right view is rotated an additional -20 degrees. Face 1 is visible in the middle view; face 2 is the oblique face visible in the right view

tions is needed over large spatial regions in order to compare with field measurement calculations. The complex chemical kinetics in the atmospheric model gives rise to abrupt and sudden changes in the concentration of the chemical species in both space and time. These changes must be matched by changes in the spatial mesh and the timesteps if high resolution is required, [28]. This application is modeled by the atmospheric diffusion equation in three space dimensions given by:

$$\frac{\partial c_s}{\partial t} + \frac{\partial u c_s}{\partial x} + \frac{\partial v c_s}{\partial y} + \frac{\partial w c_s}{\partial z} = D + R_s + E_s - \kappa_s c_s, \quad (8)$$

where c_s is the concentration of the s 'th compound, u , v and w , are wind velocities and κ_s is the sum of the wet and dry deposition velocities. E_s describes the distribution of emission sources for s 'th compound and R_s is the chemical reaction term which may contain nonlinear terms in c_s . D is the diffusion term set to zero here. For n chemical species a set of n partial differential equations (p.d.e's) is formed where each is coupled to the others through the nonlinear chemical reaction terms.

The test case model is based on that used by [28] and covers a region of 300×500 km. The chemical mechanism contains only 7 species but still represents the main features

of a tropospheric mechanism, namely the competition of the fast inorganic reactions, [28], with the chemistry of volatile organic compounds (VOC's), which occurs on a much slower time-scale. The power station is taken to be the only source of NO_x and hence the initial grid will contain more elements close to this concentrated emission source.

Figure 10 shows a SCIRun visualization of the plume developing with the adaptive mesh clustered around the developed portion of the solution. The main area of mesh refinement is along the plume edges close to the chimney. Using the adaptive mesh, we can clearly see the plume edges and can easily identify areas of high concentrations. The effects of the plume on ozone concentrations also provides some interesting results. Close to the plume the concentration of O_3 is much lower than that in the background. Due to the high NO_x concentrations the inorganic chemistry is dominant in this region and ozone is consumed. As the plume travels downwind and the NO_x levels decrease, the plume gradually picks up emissions of VOC's and leads to the production of NO_2 which in turn causes levels of ozone to rise above the background levels at quite large distances downwind from the source of NO_x .

For this atmospheric diffusion model, the meshes and means of obtaining them are more fully described in [12]. The scalar field representing error values was again derived

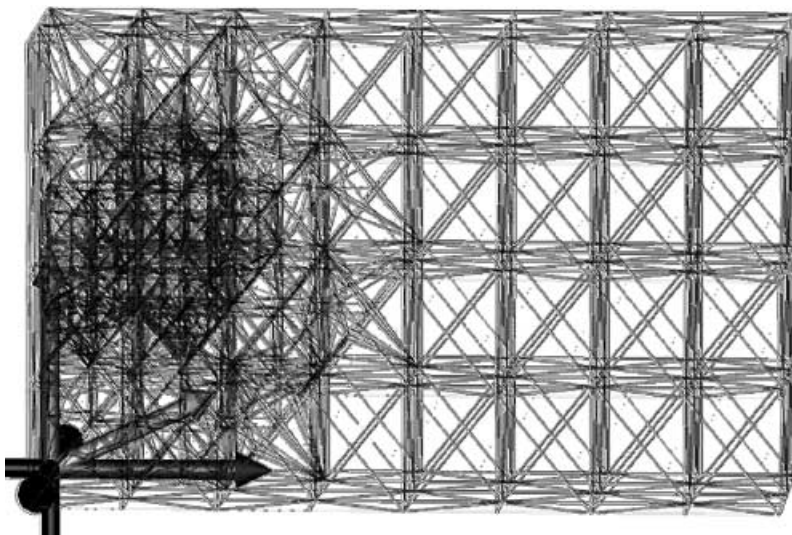


Fig. 10. Mesh for the atmospheric model example, displayed as a wire frame model. No error information is displayed. The area of heavy refinement reflects the top of the power station chimney, a point source of several of the chemicals in the model

from a simple first-order calculation based on gradients. The same formula (6) was applied to obtain separate error values for each separate chemical species as well as for a sum of the main NO_x species.

The issue of whether the mesh is appropriate for this application is somewhat more complex than for a simple linear problem. Strong local variations in solution component values make it difficult to assess the quality of the mesh for each component without somehow incorporating solution behavior. Results are presented from interactively investigating the differences among the error indicators and from analyzing one particularly bad mesh element. Since wind direction is important, it was added to the principal axes provided for orientation and appears in yellow in the figures.

5.1 Analysis of atmospheric example

Not surprisingly, the set of bad elements depends on the error indicator used. The superlative “worst” is conferred onto different elements by the different indicators, as shown in the righthand image of Fig. 11. The figure displays the union of the worst element(s) flagged by each indicator. At a broader level, the distribution of error, and its spatial distribution, also varies depending on the error indicator. This is illustrated in Fig. 12, which contrasts the sets of imperfect elements from three different indicators, those for NO, SNGN and a simple nonreacting tracer showing the wind direction. Generating

these images from the user interface was straightforward, and the results give the user immediate evidence of the effect of choice of indicator. Which indicator to pay attention to is unclear, but the potential impact of ignoring certain chemicals can be hypothesized and tested by going back to the mesh adaptation software and recalculating solution and error information, then visualizing the new results. For instance, it may be beneficial to remesh regions where any of the indicators flag elements, e.g., the regions around all of the elements in the righthand image of Fig. 11.

One element near the chimney (the red element in the left image in Fig. 11) was flagged by several indicators. It is shown in several local view modes from several vantage points in Fig. 13. As visible in the figure by color comparison, this poor quality element is surrounded by good quality elements, as was true in the simple advection example. This implies that its error is not a simple function of a single face or vertex; otherwise, one neighbor would also have a nonzero error value. Comparison of the text tags associated with the elements permitted the discovery of a large jump in the solution value across one face. The jump occurs between the central element and a single one of its neighbors.

Abrupt changes in solution component values are expected with this atmospheric model. In that sense, the location of this element puts it at risk. Its volume is slightly above average for the subregion of the mesh around the chimney; however, its error value would obviously be reduced by further refinement. The orientation, shape, and size of the of-

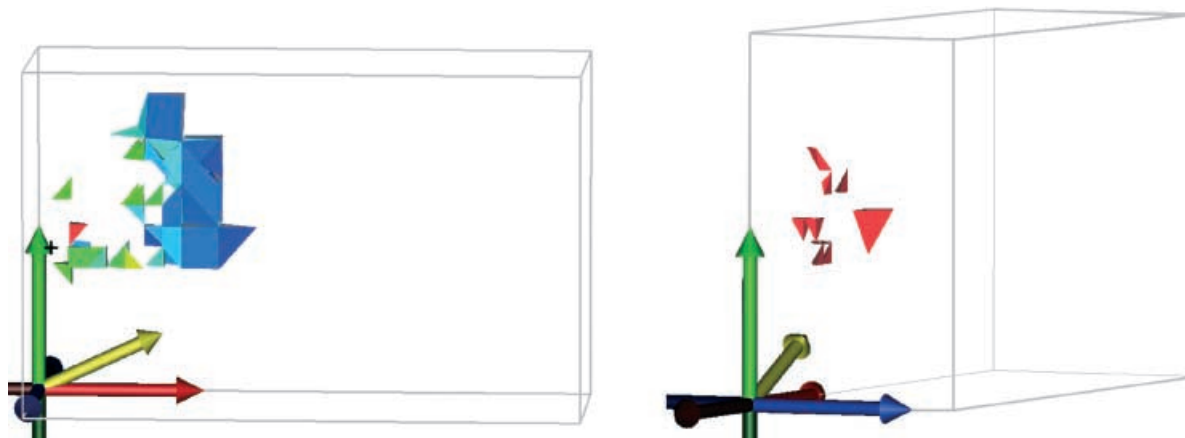


Fig. 11. *Left:* global view showing elements with error values within 85% of the maximum error from the atmospheric model example, as flagged by a specific indicator. The plus sign on the left of the mesh indicates the initial boundary of NO_x chemicals at the top of the chimney. *Right:* visualization of the set of all “worst” elements flagged by the various indicators used to analyze this mesh. Initial NO_x boundary indicated by a rectangular box near the cluster of elements



Fig. 12. Comparison of the elements flagged by three different indicators (for NO, SNGN and a tracer) used for the atmospheric model. In each figure, all elements with error values within 98% of the maximum error value are drawn. The same color map is used for all ranges and is applied to the entire scalar range in each case. The spatial distribution of error is visibly different, as is the numeric distribution of error values apparent in color patterns

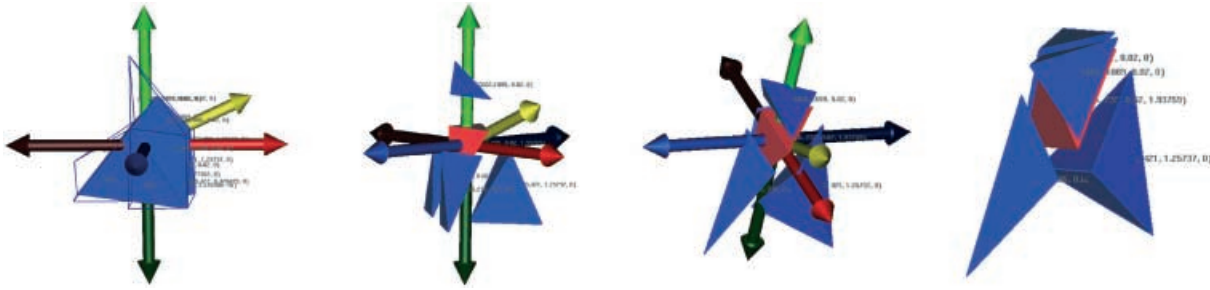


Fig. 13. Local view of the neighborhood of a bad tetrahedron in the atmospheric model. The views from *left to right* are simple translations of viewpoint; 2 levels of face-neighbor adjacency are shown in the *first image* only. An axis is provided for orientation as well as information about wind direction ($x+$ bright red, $y+$ bright green, $z+$ bright blue, wind yellow). Solids are colored by error value. Although difficult to see at this size, the text displays element index number, solution, and error value at each element's centroid. The jump in solution value occurs between the central element and the rightmost blue element

fending face appear to be factors in the solution jump, and therefore factors in this element's poor quality.

6 Conclusion

The user interface combines solution-based error indicators with geometric information about the meshes in a way that allows the user to analyze individual elements in the context of their positions within the larger mesh as well as their relative volume, shape, orientation, and vertex locations. A prototype implementation of the interface was constructed and used to examine two meshes in detail. This was done to determine the feasibility of this approach to mesh quality analysis. Informal quantitative and qualitative information was also gathered about system usability.

The interface appears to facilitate the investigation of mesh element quality, and appears to provide a possible approach with some merit. Of particular importance was the fact that this type of interface provided us with access to the mesh geometry integrated with the error information. The integration simplified the task of identifying poor (and high) quality elements and provided us with several quick insights into the meshing and its shortcomings. We were able to spend a short amount of time using the Global View and Evaporation visualization to pinpoint problem areas, and were then able to hone in on a small subregion of the mesh in the Local View. By manipulating the visualization parameters and viewpoint of the Local View we were able to determine the primary contributors to poor quality among those that interested us (element position within the larger mesh, relative element volume, shape, orientation, and vertex locations).

For the two meshes analyzed, the interface identifies the problem elements and provides several means for viewing the spatial distribution of error across the mesh. It provides the user with a simple interface for selecting a particular mesh element and presents relevant geometric and scalar information about a specific element in a manipulable form that can be used to make reasonable guesses as to the relative contributions of element location, volume, shape, orientation, and the solution behavior in the region. It also appears to be useful for comparing two different error indicators to each other: their differences are immediately apparent in the global view.

The complexity of the geometric and error information was greatly reduced through interactive display and navigation of the mesh. Interactivity was used to reduce the com-

plexity of the information displayed visually down to a mesh subregion around the user's search point. The user controlled the complexity of the visual information by controlling the size of the subset displayed. The full-mesh display was generally used to display just the worst elements in the mesh as a sort of road map to guide the user's analysis of mesh subregions. A gestural interface provided in three space allowed the user to unambiguously specify mesh regions and elements in three dimensions. Haptic textures reinforce the mapping from the pen's three dimensional workspace to the two dimensional graphic display and notify the user of contact with bad elements.

Although interactive navigation permitted the user to precisely specify a mesh region for detailed analysis, interactive navigation appears to be fairly time intensive interface for the task of developing an intuition for what factors make for good and bad elements in a specific mesh. A search point-based display appears to be a good approach for the gross comparisons that occur during the user's initial investigation, but it seems the wrong approach for detailed comparisons because it requires comparisons to be made across time and relies on the user's recall. Of the two methods implemented for comparisons, the better method for in-depth analysis appears to be defining a scalar value across mesh elements and then viewing subsets of mesh elements in a spatial representation of the mesh. It appears that, during the search for correlations between element quality and such features as element shape, volume, orientation and location, both a spatial representation of the mesh and a nonspatial ordered-by-value display would be beneficial. The interface described here would be greatly augmented by more flexible subset definition that allows a user to define precise criteria interactively. More powerful features are needed within the interface for doing comparisons and correlations across mesh elements. With the prototype, correlations between factors can be computed to arrive at a dataset containing one scalar per element, which is then used to control display parameters such as element color.

The graphics hardware used in this prototype was unsatisfactory. The 21" monitor limits the size of the windows in which the global and local views can be displayed and is too small for displaying the windows side by side. Another problem with using a monitor is that the three dimensional mesh is presented visually in two dimensions, and this requires the user to rotate the view to determine the elements' true locations or guess as to the location of elements along the collapsed z axis. Although it does not comprise a formal

study, the profile view shown in Fig. 7 shows some interesting navigation patterns which suggest that navigation is affected by the flatness of the display, and that users compensate for it by aligning their position in the (x, y) plane at some distance from the expected tetrahedron position in z , then traveling forward or backwards in z until they hit the tetrahedron. This suggests that users are using occlusion as a depth cue, and that they use it by positioning the endpoint cursor such that it occludes, or is occluded by, the tetrahedron-of-interest followed by a linear traversal to reverse the occlusion. Undesirable flatness mainly affects the global view: the local view is sufficiently shallow in depth to be little impacted by the display technique. Stereo visual feedback might improve the visual display and help the user determine the three dimensional positions of elements. A much larger display is also desirable such as those provided by a responsive workbench or CUBE, in conjunction with stereo display.

The visual display would also be improved by the introduction of a second lighting/shading model, one that provided more information about element shape. The interface defaults to use of high ambient light so that colors are truer, making color comparisons more reliable; however, this also results in minimization of shape information. Two lighting models the user could easily toggle between would solve this problem.

It would also be beneficial to use the haptic force feedback to correct for the user's hand tremor, the natural tendency to have difficulty keeping one's hand still. Since tetrahedra can be quite small within the haptic device's volume, the hand tremor itself can be enough to prevent the user from being able to "stay on" a particular tetrahedron. Here we minimized the effect of this by allowing the user to click on a tetrahedron to populate the local view, an interface that required only a momentary precision from the user. An inertia-based model might be employed to counter the effect, or perhaps a "snap to tetrahedra," analogous to the "snap to grid" mouse interface technique, which requires a significant perturbation before moving to an adjacent element. Force feedback could also be used to draw the user's hand toward bad tetrahedra. This may be advantageous force feedback over the textures used here, or it may be useful in conjunction with texturing. The technique for generating flow fields from [9]¹ could be used here to define a flow field with the worst tetrahedra as sinks to which the field flows. This would turn the pen into an active mouse, one that is naturally attracted to bad tetrahedra nearby. This may facilitate the finding of bad tetrahedra, and it may also have the effect of keeping the endpoint on the bad tetrahedron, but it would require the user to actively resist in order to keep the endpoint on a non-sink tetrahedron or require that the user be given the ability to "turn off" specific sinks or lower the forces to avoid them.

The prototype implementation, run on the given hardware, achieved interactive refresh rates for the meshes we investigated. Judging from the constant update rate as mesh size was varied, the design also appears to scale well to meshes up to 500 000 elements. Larger than 500 000 we cannot say for certain, because the given hardware was insufficient for testing

larger meshes; however, the fact that, regardless of mesh size, the interface permits the user to limit the display to a manageable number of elements bodes well for scalability. The interface design appears to scale well to larger meshes because it does not display all the mesh, just a subset, and much of the interaction occurs at a local level, where interactive rates are easy to achieve, and the rate becomes limited more by the user's analysis process than the data itself.

7 Future work

This work could be extended in a number of ways, and our experiences may be useful in providing direction for future work.

We continue to investigate the relationship between mesh quality and geometry. We will investigate larger and more complex meshes. The meshes investigated here were somewhat smaller than meshes in some applications areas. It is reasonable to expect that use of faster hardware platforms will also compensate in part for the difference in mesh sizes as we make the transition.

Other factors contributing to overall mesh quality should also be investigated, as well as the interrelationships between factors. It is perfectly possible within our environment to use different quality criteria. In essence all that is required is a way of providing each element with a value that fits on a scale of some sort. This presumes that interesting phenomena occur at the element level, which is not the case for all mesh quality metrics. Our work was motivated by trying to find those elements for which there are large errors.

There is much work to be done to translate this prototype implementation and the lessons learned from it into a general mesh analysis tool. Potential improvements to the interface include use of improved graphics hardware, redesign and deployment in a more immersive environment, and improved haptic force feedback such as hand tremor filters and creating force fields that draw the user's hand toward bad mesh elements. Better hardware would permit the display of larger mesh subregions at interactive rates. A tight coupling between the mesh analysis and the mesh structure or definition could permit the user to fix bad elements once they were identified and analyzed. Automation of useful interactions and comparisons would further decrease the amount of time the user spends doing the analysis. Investing the interface with the ability to simultaneously display two or more values defined over the mesh may help the user make correlations. Extension of this work to a more generalized class of elements, including hexahedral or mixed tet. and hex. meshes would also be beneficial.

This research simply begs the question of the ideal human/computer interface for in-depth analysis of mesh quality. Here we used search point-based display of the mesh because it suited our rather unspecific ideas as to what we were looking for within the mesh. Formal user studies of a range of possible interface designs are needed to accurately compare methods; however, based on this work, it seems that what is needed to aid researchers in mesh quality analysis is a rich command language, tightly coupled with display commands, followed by some degree of interactive visualization of the results. For instance, the user should be able to define a shape function, have all mesh elements analyzed with it, display all elements where shape fits a user defined set of

¹ The technique for generating the force field is not explicitly stated in the paper. It is to define the $[x, y, z]$ location of a number of sinks (or sources) and, for each point in the flow field, to compute the net force as a function of distance to/from all sinks/sources.

criteria, and then work with the results to form a useful visualization, and in some cases also interact with the results to glean information and form opinions. The Qviz data query and visualization framework developed at Los Alamos may provide a good platform for development of a good command interface and visualization interface, since it fits this investigative model and can handle very large datasets [14] and appears to be extensible to the kinds of data and queries common to finite element meshes and quality metrics. In some cases a simple visualization is sufficient; in others, an interactive visualization environment can be of assistance. Given that mesh quality metrics are not yet well understood, users will likely be using the system to determine good metrics, and using the visualizations in part to guide their investigation of the metrics, which may be somewhat of a backward, or at least iterative, use of the Qviz framework.

Providing the user with means to further manipulate this element set in ways tangential to the original shape function is also likely to be useful, such as providing ways for the user to look at the regions around each element. Depending on the mesh generation and refinement strategies used and the types of information the user cares about, these goals may only be possible if their implementation is tightly intertwined with the mesh generation or refinement packages as a window into their actions and a way to influence their outcomes.

Acknowledgements. The authors would like to thank C.R. Johnson, J.M. Hollerbach, P. Shirley, N.J. Macias, and the reviewers for their helpful comments and L. Zhukov, N.J. Macias, D.M. Weinstein, R. Freier, P.P.J. Sloan, S.G. Parker, K. Zimmerman, and R. Cummins for providing code or assistance with technical issues.

References

- Advanced Visual Systems Inc: AVS – Products – AVS5 Overview. Advanced Visual Systems Inc., 23 Feb. 2000. <http://www.avs.com/products/AVS5/avs5.htm>
- Bern, M., Epstein, D.: Mesh Generation and Optimal Triangulation. Report CSL 92-1, Xerox Corporation, Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304
- Berzins, M.: Temporal Error Control in the Method of Lines for Convection Dominated Equations. *SIAM J. Sci. Comput.* 16, 558–580 (1995)
- Berzins, M.: Mesh Quality: A Function of Geometry, Error Estimates or Both? *Engineering with Computers* 15, 236–247 (1999)
- Berzins, M., Durbeck, L.J.K.: Mesh Quality for Unstructured Mesh Riemann Solver Based Methods Applied to Hyperbolic PDEs with Source Terms. In: Toro, E. (Ed.): *Godunov Methods: Theory and Applications* (Proc. of Godunov Conf. October 18–22, 1999, Oxford UK), pp. 117–124. Dordrecht: Kluwer Academic/Plenum 2001
- D. Deitz: Designing with CFD. *Mech. Eng.* 118.3, 90–94 (1996)
- Durbeck, L.J.K.: Evaporation: a Technique for Visualizing Mesh Quality. Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, 11–13 October 1999, pp. 259–265
- Durbeck, L.J.K.: Contrast Displays: A Haptic and Visual Interface Designed Specifically for Mesh Quality Analysis. M.Sc. Thesis, University of Utah 1999
- Durbeck, L.J.K., Macias, N.J., Weinstein, D.M., Johnson, C.R., Hollerbach, J.M.: SCIRun/Haptic Display for Flow Fields. In: Salisbury, J.K., Srinivasan, M.A. (Eds.): *Proceedings of the Third PHAN-ToM Users Group Workshop*, Dedham, 3–6 October 1998: AI Lab Technical Report No. 1643 and RLE Technical Report No. 624, December 1998, pp. 71–75. Cambridge: MIT 1998
- Gitlin, C.S.: Techniques for Visualizing Three-Dimensional Unstructured Meshes. M.Sc. Thesis. University of Utah 1995
- Gitlin, C.S., Johnson, C.R.: MeshView: A Tool for Exploring 3D Unstructured tetrahedral meshes. Proceedings of the 5th International Meshing Roundtable, 1996, pp. 333–345
- Johnson, C.R., Parker, S.G.: Applications in Computational Medicine using SCIRun: A Computational Steering Programming Environment. In: Meuer, H.W. (Ed.): *Proceedings of Supercomputer '95*. Mannheim, 22–27 June 1995, pp. 2–19. Mannheim 1995
- Johnson, C.R., Berzins, M., Zhukov, L., Coffey, R.: SCIRun: Applications to Atmospheric Diffusion Using Unstructured Meshes. In: Baines, M.J. (Ed.): *Numerical Methods for Fluid Dynamics VI*, pp. 111–122. Oxford: Oxford University Press 1998
- Keahey, T.A., McCormick, P., Ahrens, J., Keahey, K.: Qviz: a Framework for Querying and Visualizing Data. Proceedings of SPIE Vol. 4302 – Visual Data and Exploration and Analysis VIII, January 2001
- Kroner, D., Ohlberger, M.: A posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi-dimensions. *Math. Comput.* 69, 25–39 (2000)
- Lawrence Livermore National Laboratories: MeshTV: Scientific Visualization and Graphical Analysis Software. Lawrence Livermore National Laboratories, 20 June 1999. <http://www.llnl.gov/bdiv/meshtv/>
- Liu, A., Joe, B.: Relationship between tetrahedron shape Measures. *BIT* 34, 268–287 (1994)
- Los Alamos National Laboratory: GMV Home Page. Los Alamos National Lab., 23 Feb. 2000. <http://www-xdiv.lanl.gov:80/XCM/gmv/GMVHome.html>
- Massie, T.H.: Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction. Thesis. M.I.T. 1996
- Parker, S.G., Weinstein, D.M., Johnson, C.R.: The SCIRun Computational Steering Software System. In: Arge, E., Bruaset, A.M., Langtangen, H.P. (Eds.): *Modern Software Tools in Scientific Computing*, pp. 1–44. Basel: Birkhauser Press 1997
- Parker, S.G., Johnson, C.R.: SCIRun: A Scientific Programming Environment for Computations Steering. Proceedings of the 1995 ACM/IEEE Supercomputing Conference. San Diego, 3–8 Dec. 1995
- Parthasarathy, V.N., Griachen, C.M., Hathaway, A.F.: A comparison of tetrahedron quality measures. *Finite Element Analysis and Design* 15, 255–261 (1993)
- Pavlakos, C., Jones, J., Mitchell, S.: An Immersive Environment for the Exploration of CUBIT Meshes. Proceedings of the 6th International Meshing Roundtable, October 1997, pp. 47–48
- Sandia National Laboratories: Unstructured Grid Visualization Tools. Sandia National Laboratories, 23 Feb. 2000. <http://www.cs.sandia.gov/VIS/unstruct.html>
- Sandia National Laboratories: VR-assisted Mesh Generation. Sandia National Laboratories, 23 Feb. 2000. <http://www.cs.sandia.gov/VIS/cubitvr.html>
- SensAble Technologies: Product Specs – Netscape. SensAble Technologies, 3 May 1999. <http://www.sensable.com/products>
- Speares, W., Berzins, M.: A 3D Unstructured Mesh Adaptation Algorithm for Time-Dependent Shock-dominated Problems. *Int. J. Numer. Methods Fluids* 25, 81–104 (1997)
- Tomlin, A.S., Ghorai, S., Hart, G., Berzins, M.: The Use of 3-D Adaptive Unstructured Meshes in Air Pollution Modelling. In: Zlatev, Z. et al. (Eds.): *Proceedings of Nato Workshop on Air Pollution Modelling*, Sofia, 1998. Large Scale Computations in Pollution Modelling, pp. 339–348. Dordrecht: Kluwer Academic Publishers 1998
- Weatherill, N.P., Hassan, O., Morgan, K., Marchant, M.J.: Large scale computations on unstructured grids, In: Benkhaldoun, F., Vilsmeier, R. (Eds.): *Proceedings of the Conference on Finite Volumes for Complex Applications*, pp. 77–98. Paris: Hermès 1996
- Weatherill, N.P., Marchant, M.J., Hassan, O.: Unstructured Grid Generation and Adaptation for a transport Aircraft Configuration. Paper presented at 1993 European Forum on Recent Developments and Applications in Aeronautical Computational Fluid Dynamics. Held at Bristol UK, 1–3 September 1993