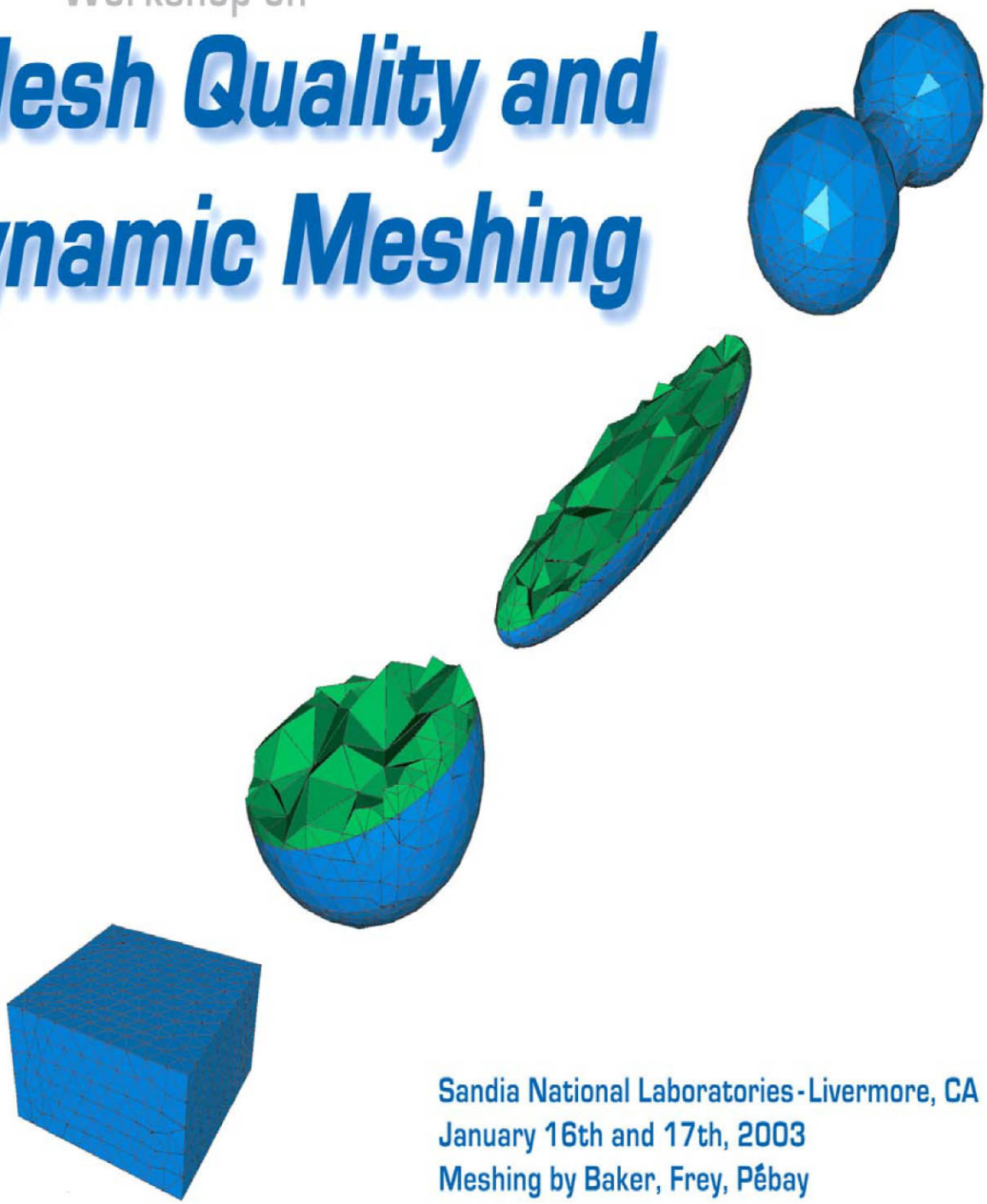
Workshop on

Mesh Quality and Dynamic Meshing



Sandia National Laboratories - Livermore, CA
January 16th and 17th, 2003
Meshing by Baker, Frey, Pébay

Workshop on

Mesh Quality and Dynamic Meshing

Table of Contains

UNTANGLING OF 2D TRIANGULAR AND QUADRILATERAL MESHES IN ALE SIMULATIONS

PAVEL VACHAL[¶], MIKHAIL J. SHASHKOV[†], & RAO V. GARIMELLA[‡]

[¶]Czech Technical University in Prague,
Czech Republic

[†‡]MS B284, T-7,
Los Alamos National Laboratory,
Los Alamos, NM USA

ABSTRACT: A procedure is presented to untangle unstructured 2D meshes containing inverted elements by node repositioning. The inverted elements may result from node movement in flow simulations and in large deformation problems such as metal forming. Meshes with inverted elements may also be created due to the limitations of mesh generation algorithms particularly for non-simplicial mesh generation. The untangling procedure uses a combination of direct node placement based on geometric computation of the feasible set and node repositioning driven by numerical optimization of an element area based objective function. It is shown that a combination of the feasible set based method and the optimization method achieves the best results in untangling the mesh. Preliminary results are also presented for untangling of 3D unstructured meshes by the same approach.

KEY WORDS: Mesh untangling, inverted elements, triangles, quadrilaterals

Introduction

Arbitrary Lagrangian-Eulerian or ALE methods are a popular class of methods for simulating flow problems and large deformation problems [1, 2]. ALE methods consists of a Lagrangian step in which the mesh nodes move according to the flow of the material, a rezone step in which the mesh is modified to improve its quality and the remapping step in which the solution is transferred from the old mesh to the new, improved mesh. In [3, 4, 5], methods were described to improve quality of the mesh while keeping it close to the original mesh. However, in order to improve the meshes by the methods described in [3, 4, 5], all elements of the starting mesh must be valid or non-inverted. Therefore, if the Lagrangian step of an ALE simulation causes the mesh to become tangled (i.e., it has some elements that become inverted), the mesh must be untangled before the mesh improvement procedures are applied to it.

The need for untangling meshes also exists when a mesh generation procedure is unable to create all valid elements in a mesh. This situation may be encountered in the generation of all hexahedral meshes and or general polyhedral meshes where there is no guaranteed method of directly generating a valid mesh [6]. It may also be encountered

[¶]vachal@lanl.gov
[†]shashkov@lanl.gov
[‡]rao@lanl.gov

in advancing front based mesh generation of tetrahedral and hexahedral [7, 8, 9] where it is possible to generate small cavities that cannot be filled with all positive volume elements. In these cases, it is very useful to have a tool that can untangle the mesh after the initial generation assuming that the right mesh connectivity has been generated.
Several researchers have recently begun focusing on the problem of untangling unstructured meshes by node repositioning [10, 11, 12, 13]. Freitag and Plassman [11, 14] untangle meshes by optimization of a local function based on maximizing the minimum element area at each mesh vertex. Knupp [12] performs a global optimization of the difference between the absolute and signed values of element volumes in order to optimize the mesh. Kovalev et.al. [13] visit each vertex connected to at least one invalid element and reposition the vertex directly to a point in its feasible set (or "kernel") to make all connected elements valid. They define the feasible set of a vertex to be the set of all locations of the vertex for which all elements connected to the vertex will be valid. The new location of the vertex in the feasible set is found by a clever use of the simplex method to find three corners of the feasible set and taking their mean. Most of these procedures are capable of successfully untangling many tangled mesh configurations. However, there are conditions under which they may fail to untangle the mesh or they may not be able to produce elements with sufficient positive volume so as to provide a good starting point for mesh improvement procedures.
In this paper, a method is presented for untangling unstructured 2D meshes containing triangles and quadrilaterals. The method uses a multi-step approach based on relocating vertices to points in its feasible set and on minimizing a function to untangle the mesh elements. The approach adopted here focuses only on ensuring that all elements of the mesh meet at least bare minimum validity criteria as defined by the simulation procedures that will use them. However, the methods can easily be combined with separate procedures for improving the quality of the untangled mesh like the ones described in [4, 5].

1 Definitions

In this study, an element is considered valid if each corner of the element is considered valid. An element corner is valid if the Jacobian determinant of its mapping to a right corner is positive.
The ***feasible set*** *for a vertex can be defined as the set of all positions of the vertex for which each connected element is valid at the corners affected by the position of the vertex*.
The feasible set for a vertex connected to a single triangle is a half-space as shown in the 2D example in Figure 1a. For a vertex connected to a general polygonal element, the feasible set of a vertex is defined by the intersection of three half-spaces as shown in Figure 1b for a pentagon. Figure 1c shows the feasible set for a vertex interior to a patch of elements. From the definition of the feasible set, it is clear that the feasible set is a convex polygon in 2D while it is a convex polyhedron in 3D. Note that the feasible set for a vertex can be empty depending on the geometry and configuration of the elements connected to the vertex. This implies that given the positions of the boundary vertices of the patch, there is no location for the vertex which will simultaneously make all the

elements valid.

2 Untangling by Finding the Feasible Set

The definition of a feasible set given above leads to a natural method of untangling meshes in which the feasible set of a vertex is determined and the vertex is positioned inside it. This is referred to here as the *feasible set method*. In this approach each vertex of the mesh is visited and its connected elements examined to see if any of them are invalid. If an invalid element exists among the elements connected to the vertex, the feasible set of the vertex is computed and the vertex is placed inside the feasible set. The method loops over the mesh until all elements are valid or no invalid element can be fixed.

The feasible set of a vertex can be found by computing the intersection of half-spaces representing the feasible region of the vertex with respect to each element. In 2D, this is accomplished by the intersection of pairs of lines demarcating the feasibility half-planes [15].

In Figure 2, an example of mesh untangling by the intersection based feasible set method is illustrated step by step. In the figure, the invalid quadrilaterals are shown shaded and nodes connected to at least one invalid quadrilateral are shown in black.

While the intersection based computation of the feasible set works well in 2D, finding the feasible set polyhedron by intersection of half-spaces is impractically complex in 3D. Therefore, an alternate method for positioning the vertex inside the feasible set is implemented based on the simplex method as described in [13]. In [13] the idea is proposed that the boundary lines of the half-planes forming the feasible set in 2D can be interpreted as inequality constraints on the minimization of an arbitrary function. If the function is a linear function then its minimum must occur at the intersection of two of these inequality constraints or in other words the corner of the feasible set polygon. Therefore, the corners of the feasible set polygon can be found by minimizing different linear functions along with the appropriate inequality constraints using the simplex method [16, 17].

This idea is further simplified by recognizing that for untangling, it is sufficient to find any one position for the vertex inside its feasible set to make all connected elements valid. This implies that it is unnecessary to find all corners of the feasible set polygon. Therefore, for untangling a patch of elements in 2D, it can be inferred that it is sufficient to find three distinct corners of the feasible set polygon and reposition the vertex to the center of the triangle formed by these corners. Therefore, the procedure minimizes simple linear functions such as $f(x,y) = x$, $f(x,y) = -x$ and $f(x,y) = y$ to find three distinct corners of the feasible set polygon. The vertex is then repositioned to the center of the triangle formed by these corners.

The feasible set approach to untangling is very useful because it is a direct way of fixing inverted elements affecting only those nodes connected to invalid elements. However, the shortcoming of this approach is that some elements cannot be fixed because the feasible set associated with each of their vertices is empty.

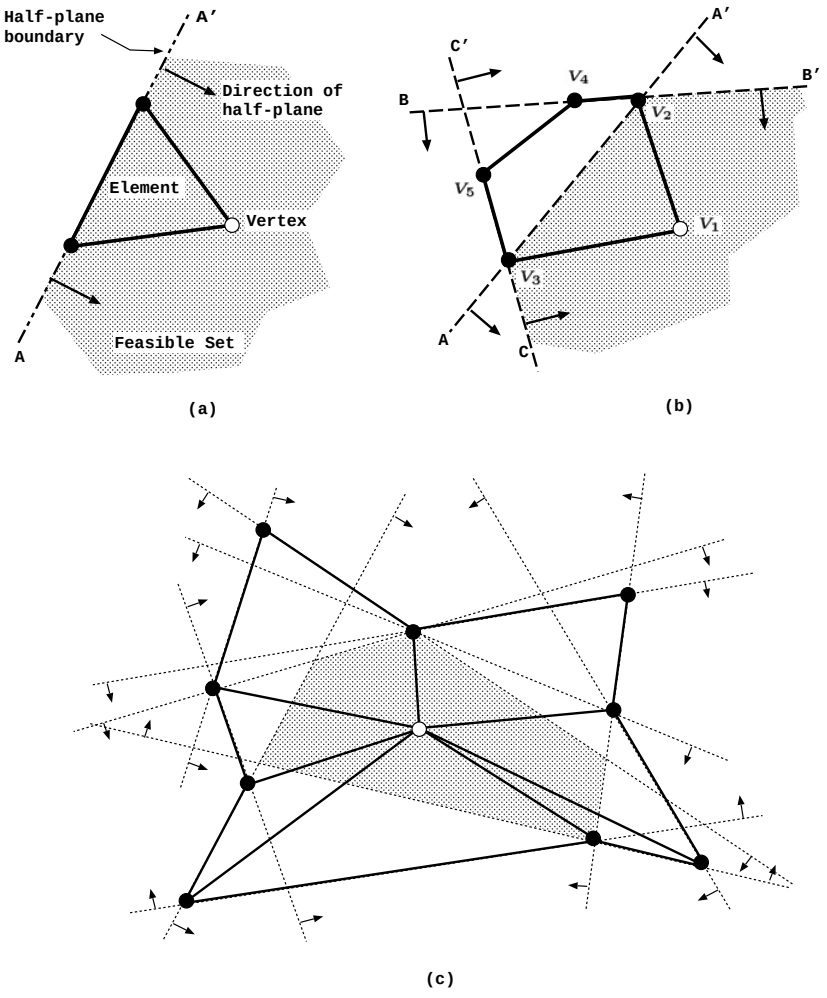


Figure 1: Illustration of Feasible Sets (a) Feasible set for vertex connected to single element (b) Feasible set for vertex connected to a pentagon (c) Polygonal feasible set for vertex connected to patch of triangular and quadrilateral elements.

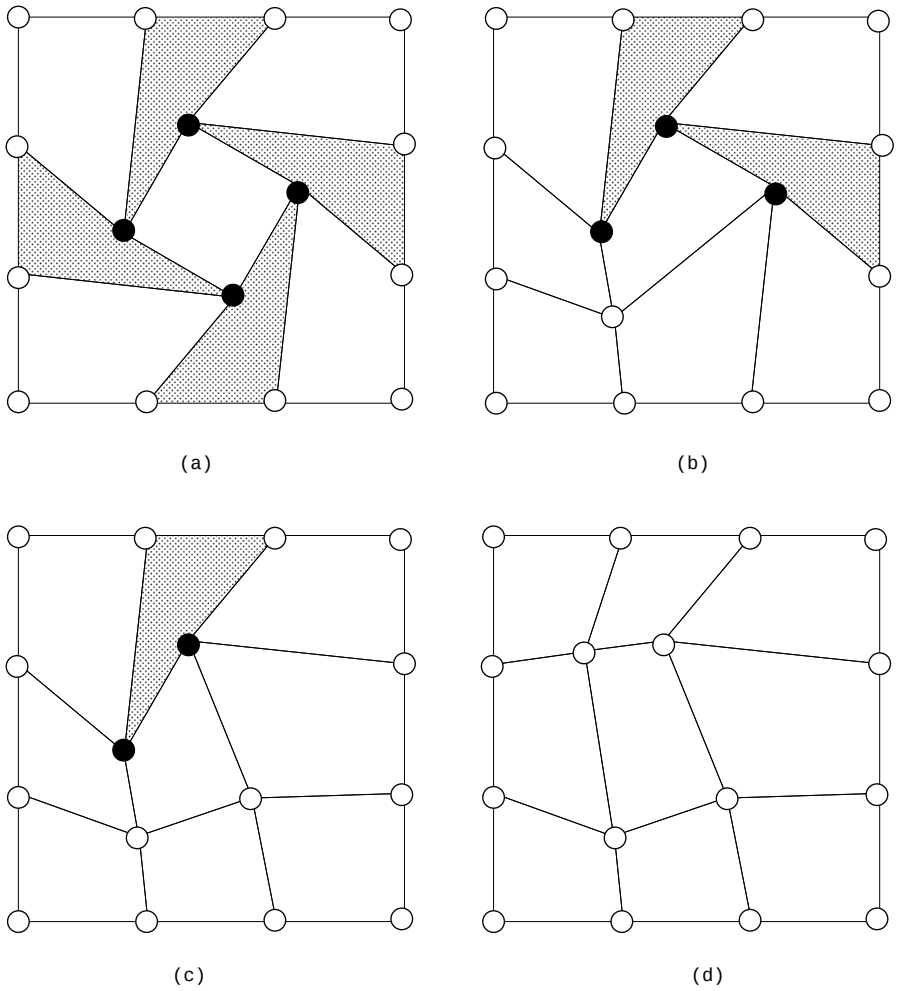


Figure 2: Untangling of mesh by feasible set method (a) Initial mesh (b),(c) Intermediate meshes (d) Final mesh. The shaded quadrilaterals indicate invalid elements and vertices represented in black are connected to at least one invalid quadrilateral.

3 Optimization Approach to Untangling

An alternative approach for untangling meshes is by minimizing an appropriate objective function so as to make all the invalid elements valid [11, 12]. Knupp [12] proposed optimization of a global objective function based on the diference between signed volumes of elements and their corresponding absolute values. If the area of the i'th element in a mesh is α_i, then the function to be minimized is

$$f(x) = \sum_{i}^{n}(\mid \alpha_i \mid -\alpha_i) \tag{3.1}$$

Minimization of this function can only bring elements to a zero area (volume) state which is still considered unusable in numerical simulations. Therefore, Knupp suggested that a user controlled parameter β be added to the function modifying it to be

$$f(x) = \sum_{i}^{n}(\mid \alpha_i - \beta \mid -(\alpha_i - \beta)) \tag{3.2}$$

The role of β is to force the function to reach a minimum when the elements have a small positive volume instead of zero volume.
In this work, the objective function proposed by Knupp in [12] has been modified so that it is quadratic and smooth as shown in Eq. 3.3 below.

$$f(x) = \sum_{i}^{n}(\mid \alpha_i - \beta \mid -(\alpha_i - \beta))^2 \tag{3.3}$$

This smooth function can then be minimized using a numerical optimization method such as the conjugate gradient method [16, 17]. It has been found that, in pratice, minimization of the quadratic form of the objective function untangles the mesh more reliably than the linear form.
The advantage of the optimization approach to untangling described above is that is guaranteed to make all elements at least non-negative. However, the method can have a non-local effect on the mesh since it may have to mean several nodes in a local neighborhood in order to fix an invalid element.
Figure 3 shows example of the mesh shown in Figure 2, untangled by the optimization procedure. Figure 3a shows the mesh optimized using the quadratic objective function without the use of β or $\beta = 0$. The procedure untangles the mesh but because β is zero, some of the valid elements are barely valid. The barely valid elements and the interior vertices connected to these elements are shaded in the figure. Figure 3b shows the same optimization but with a finite β, which is calculated as 10% of the problem size (diagonal of the bounding box) normalized by the number of elements in either the x or y directions and has the value of 0.106. As seen from the figure, the mesh is better in this case since all the elements are positively valid.
In using the optimization procedure for untangling meshes, the choice of β must be made carefully. Without β (i.e., $\beta = 0.0$), the optimization procedure can only make all the elements valid in the sense that the volume of every element is positive or at least zero. On the other hand, using an indiscrimanetely large β can be detrimental since the objective function minimum can become non-zero. This implies that for the

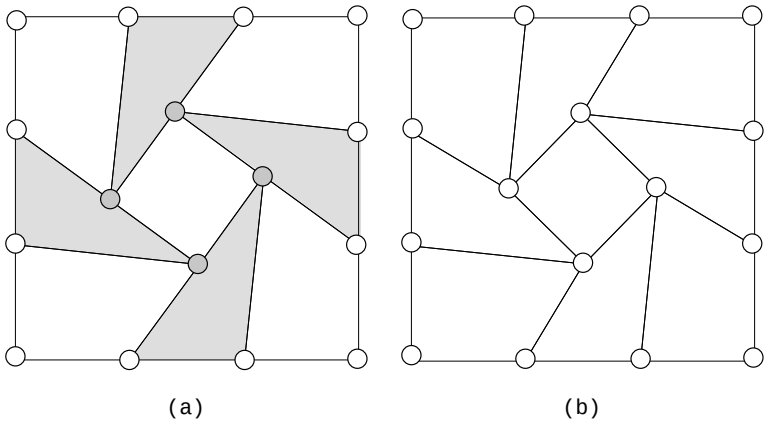


Figure 3: Invalid mesh of Figure 2 optimized using optimization with quadratic objective function (a) Without β (barely valid elements shown shaded) (b) With $\beta = 0.106 =$ 10% of bounding box diagonal normalized by number of elements in x or y directions

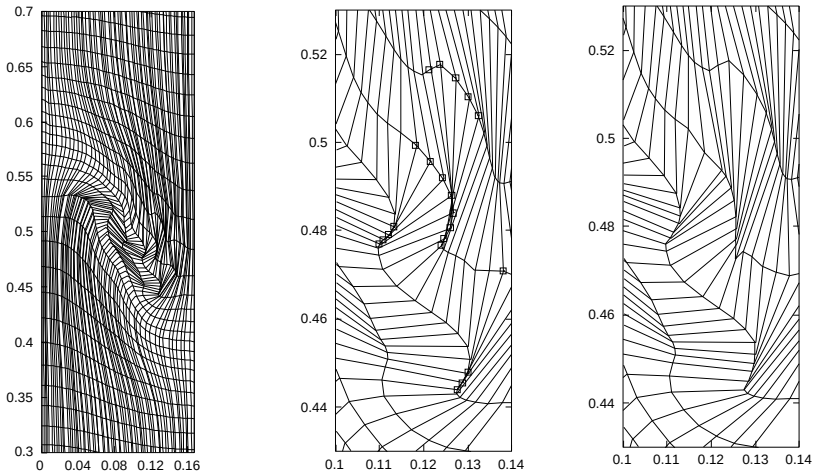
given boundary configuration it is not possible for all elements to achieve a volume equal to or greater than β. Therefore, the approach adopted here is to set β to the minimum acceptable area or volume of the elements with respect to the mesh optimization procedures that will subsequently improve the mesh.

4 Mesh Untangling by 3 Stage Procedure

In the previous sections, it was seen that the feasible set approach had a local effect on the mesh but could not always fix the mesh. On the other hand, the optimization approach, usually fixed the mesh by making all elements non-negative but could affect a larger number of nodes and resulted in barely valid elements. Therefore, the procedures have been combined here into a 3-step procedure for maximizing the possibility of untangling the mesh with minimal impact on the mesh.

The 3-step procedure for untangling the mesh first performs untangling by the feasible set method so as to fix as many elements as possible with minimal impact to the valid part of the mesh. The second step of the procedure performs a minimization of the quadratic objective function described earlier in order to fix any remaining invalid elements. The optimization procedure first performs a local or vertex-by-vertex optimization loop over the boundary vertices in order to try and fix as many elements as possible by their movement. The movement of the boundary vertices is constrained to the original discrete boundary by a local parameterization technique [5].

The third step of the procedure performs another round of untangling on barely valid elements by the feasible set approach to try increase their volume. This step requires redefinition of the feasible set boundaries to account for the desired element volumes. Once this is done, the mesh is usually suitable for use as input to a mesh quality im-



(a) (b) (c)

Figure 4: Untangling of Lagrangian mesh from Rayleigh-Taylor Simulation (a) Part of original tangled mesh (b) Zoom-in of tangled mesh (nodes shown by a square are connected to at least one invalid element) (c) Zoom-in of untangled mesh

provement procedure such as the one described in [4, 5].

5 Results

The first example shown in Figure 4 illustrates the untangling of a mesh arising from a Rayleigh-Taylor simulation. The mesh is made invalid during the Lagrangian step of an ALE simulation of the problem and must be fixed before the simulation can proceed. Figure 4a shows a part of the overall domain, Figures 4b shows the tangled portion of the mesh (with squares marking nodes connected to invalid elements). Figures 4c shows the untangled mesh corresponding to the tangled meshes shown in Figures 4a,b. The stages of untangling of this mesh are illustrated further in Figure 5. Figure 5a shows a zoom-in of the tangled portion of the original mesh. Figure 5b shows the mesh after the first untangling step by the feasible set method which is unable to fix all the elements. Figure 5c shows the mesh after untangling by optimization during which all elements were made at least barely valid. Finally, Figure 5d shows the mesh after the second round of untangling by the feasible set method during which all elements were brought to a positive volume state.

Figure 6 presents the example of untangling a mesh of the state of Texas. An originally valid mesh shown in Figure 6a was tangled by a random perturbation of a subset of the interior vertices to result in the mesh shown in Figure 6b. The maximum perturbation was 20% of the domain size. This mesh was then successfully untangled using the

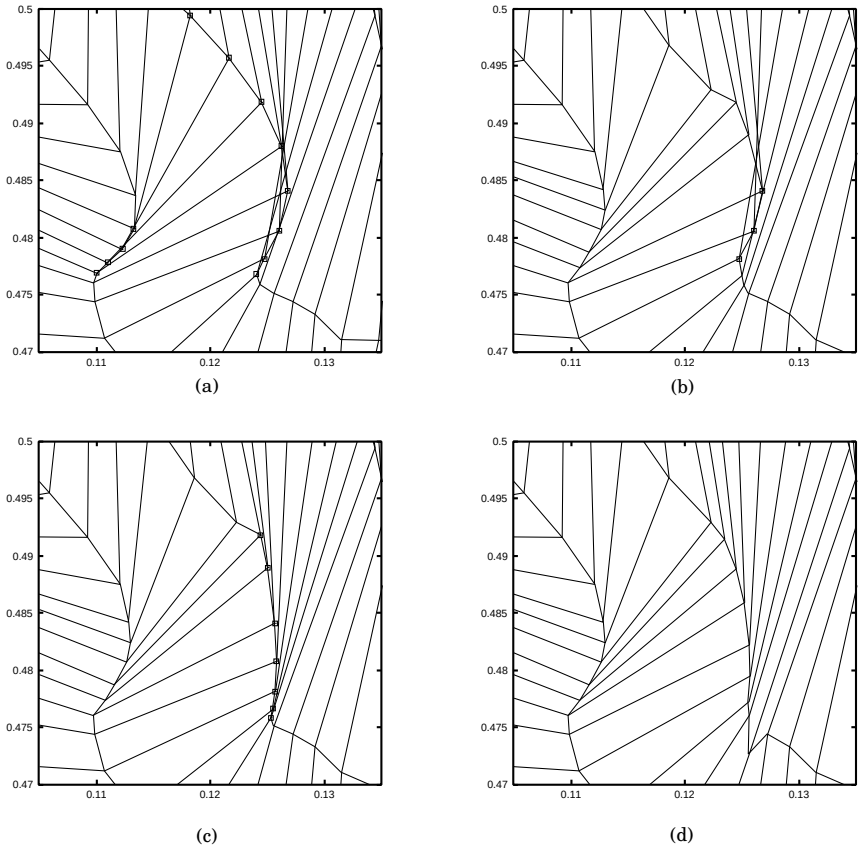


Figure 5: Stages of untangling of mesh from Rayleigh-Taylor simulation (a) Zoom-in of original mesh (nodes connected to invalid elements shown as squares) (b) Mesh after application of feasible set approach, some elements remain with less than a minimum volume (c) Mesh after optimization, some elements remain with zero volume (d) Mesh after second application of the feasible set approach, all elements have volume greater than required minimum.

3-step procedure to give the mesh shown in Figure 6c. For this example the feasible set method alone left behind 17 patches with negative area triangles and the optimization method alone left behind 6 zero area triangles. The value of β was zero for the optimization step. The result of mesh improvement on the untangled mesh is presented in Figure 6d to illustrate that good quality meshes can be obtained when the untangling procedure is combined with mesh improvement procedures.

Conclusions

A multi-step method for successful untangling of unstructured 2D meshes has been presented in this paper. The method uses a combination of the feasible set method and optimization method to achieve the greatest degree of success in untangling the mesh while keeping the mesh close to the original mesh as required for remapping in ALE simulations. The methods have shown a high degree of success in untangling complex 2D meshes. The formulation of the procedures allows easy extension to 3D problems and preliminary results are promising.

6 Acknowledgments

The work of the authors was performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.
The work of P. Vachal was also supported in part by the Czech Grant Agency grant 201/00/0586

References

[1] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997. Originally published in Journal of Computational Physics, 1974.

[2] D.J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:235–395, 1992.

[3] P. M. Knupp, L. G. Margolin, and M. J. Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176:93–128, 2002.

[4] M. J. Shashkov and P. M. Knupp. Optimization-based reference-matrix rezone strategies for arbitrary Lagrangian-Eulerian methods on unstructured grids. In *Proceedings of the Tenth Anniversary International Mesh-*

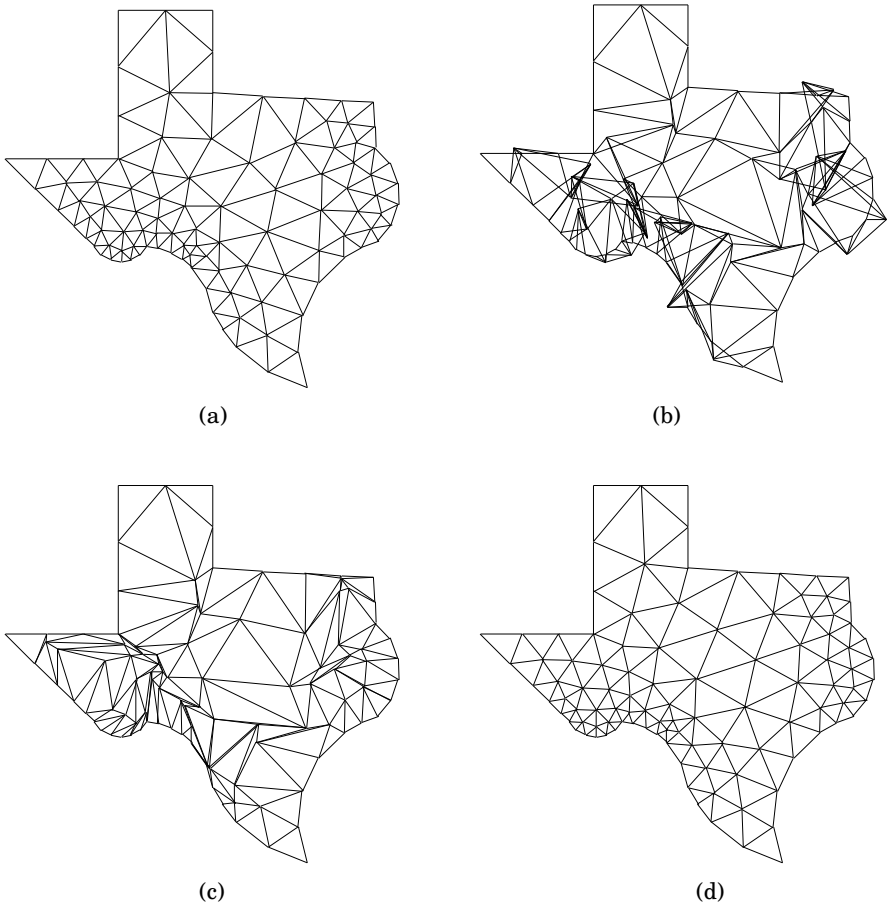


Figure 6: Untangling of triangular mesh of Texas outline (a) Original valid mesh (b) Tangled Mesh (random movement of interior vertices) (c) Untangled mesh with poor quality elements (d) Mesh improved based on optimization of element condition numbers.

ing Roundtable, pages 167–176, Newport Beach, CA, October 2001. http://math.lanl.gov/~shashkov.

[5] R. V. Garimella, M. J. Shashkov, and P. M. Knupp. Optimization of surface mesh quality using local parameterization. In *Proceedings of the Eleventh International Meshing Roundtable*, pages 41–52, Ithaca, NY, September 2002.

[6] T.J. Tautges, T. Blacker, and S.A. Mitchell. The Whisker Weaving Algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes. *International Journal of Numerical Methods in Engineering*, 39(19):3327–3349, 1996.

[7] T. D. Blacker and M. B. Stephenson. Seams and wedges in plastering: A 3D hexahedral mesh generation algorithm. *Engineering with Computers*, 9:83–93, 1993.

[8] R. Löhner and P. Parikh. Three-dimensional grid generation by the advancing front method. *International Journal for Numerical Methods in Engineering*, 8:1135–1149, 1988.

[9] B.K. Karamete, M.W. Beall, and M.S. Shephard. Triangulation of arbitrary polyhedra to support automatic mesh generators. *Internation Journal for Numerical Methods in Engineering*, 49(1):167–191, 2000.

[10] T.S. Li, Y.C. Wong, S.M. amd Hon, C.G. Armstrong, and R.M. McKeag. Smoothing by optimisation for a quadrilateral mesh with invalid elements. *Finite Elements in Analysis and Design*, 34:37–60, 2000.

[11] L. Freitag and P. Plassmann. Local optimization-based simplicial mesh untangling and improvement. *International Journal of Numerical Methods for Engineering*, (49):109–125, 2000.

[12] P.M. Knupp. Hexahedral and tetrahedral mesh untangling. *Engineering with Computers*, 17:261–268, 2001.

[13] K. Kovalev, M. Delanaye, and Ch. Hirsch. Untangling and optimization of unstructured hexahedral meshes. In S.A. Ivanenko and V.A. Garanzha, editors, *Proceedings of Workshop on Grid Generation: Theory and Applications*, Moscow, Russia, 2002. Russian Academy of Sciences.

[14] L. Freitag and P. Plassman. Local optimization-based untangling algorithms for quadrilateral meshes. In *Proceedings of the Tenth Anniversary International Meshing Roundtable*, pages 397–406, Newport Beach, CA, October 7-10 2001.

[15] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, Cambridge, UK, 1996.

[16] Reklaitis, Ravindran, and Ragsdell. *Engineering Optimization - Methods and Applications*. Wiley Interscience, 1983.

[17] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

PARALLEL GUARANTEED QUALITY DELAUNAY MESH GENERATION AND REFINEMENT: CURRENT STATUS

v1.0 4/4/2003

NIKOS CHRISOCHOIDES[¶]

[¶] Computer Science Department
College of William and Mary
Williamsburg, VA 23185

ABSTRACT: Unstructured mesh generation and refinement poses a number of difficult technical challenges, both in terms of algorithms and software. During the last five years we have directed our research effort on mesh quality and parallel mesh generation and refinement. This paper describes a summary of our results, open problems and future directions for parallel guaranteed quality Delaunay mesh generation and refinement.

KEY WORDS: Parallel, Mesh, Generation, Refinement, Quality, Delaunay.

Introduction

Parallel mesh generation methods decompose the original meshing problem into smaller subproblems that can be solved (i.e., meshed) in parallel. The requirements for the solution of the subproblems on distributed memory computers are: (1) *stability*, distributed meshes should retain the good quality of elements and decomposition properties of the sequentially generated and partitioned meshes, (2) *scalability* (3) *fault-tolerance*, and (4) *code re-use*, in order to leverage from the ever evolving basic sequential meshing techniques. The design and implementation of well tested and fine tuned adaptive mesh refinement codes is a very expensive and time consuming task. The same task becomes orders of magnitude more complex in the case of *stable, scalable and guaranteed-quality mesh generation*.

In order to develop cost-effective (i.e., high code re-use), stable and scalable guaranteed-quality parallel mesh generation codes we are experimenting with three different approaches: *(1) parallelization of existing sequential guaranteed quality Delaunay methods, (2) parallel Constrained Delaunay meshing techniques, and (3) parallel Domain Decoupling approaches.* In the rest of the paper we present a short overview on the current state-of-the-art parallel methods for mesh generation and then we will briefly describe the three approaches we developed at the College of William and Mary in collaboration with our colleagues at Cornell University in the context of crack propagation project [5].

[¶]nikos@cs.wm.edu

1 Parallel Mesh Generation: An Overview

In [15] parallel mesh generation methods, for distributed memory computers are classified in terms of the *way* and the *order* the artificial boundary surfaces (interfaces) of the subproblems are meshed. Specifically, parallel methods are classified in three large classes: (i) the methods that mesh simultaneously the interfaces as they mesh the individual subproblems [14, 7, 12], (ii) methods that first mesh (in parallel or sequentially [22]) the interfaces of the subproblems and then mesh in parallel the individual subproblems, and (iii) the methods that first solve the meshing problem in each of the subproblems in parallel and then mesh the interfaces [16] so that the global mesh is consistent. Other classifications are possible, for example in [19] Lohner et al. classify parallel mesh generation methods in terms of the basic sequential meshing techniques used to mesh the individual subproblems.
In [10] we evaluate and classify parallel mesh generation methods in terms of two fundamental issues: *concurrency and synchronization* that affect the stability, performance, and code re-use of parallel meshing methods. The level (coarse-grain or fine-grain) at which concurrency is explored and the type (local or global) synchronization is required, determine the degree code re-use is possible and vise-versa. For example, high code re-use of inherently tightly coupled methods, it reduces opportunities for concurrency at a fine-grain level [22] or requires frequent global synchronization [19]. Concurrency and synchronization are independent off implementation choices like communication paradigm (shared memory or message passing) and programming model (data-parallel or task-parallel) of parallel and distributed mesh generation codes. And therefore concurrency and synchronization can be used as key attributes to evaluate parallel mesh generation methods. Parallel mesh generation methods can explore concurrency in either a coarse-grain level (i.e., subdomain or submesh level of the geometry to be meshed) or a fine-grain level (node or element level of the mesh) and they synchronize either globally (all processors are involved) or locally (only small set of processors is involved).

2 Parallel Guaranteed Quality Delaunay Mesh Generation

Most of the traditional parallel mesh generation methods explore concurrency at a coarse-grain level using stop-and-repartition methods which are based on global synchronization —a major source of overhead for large-scale parallel machines (see Figure 1). Also, traditional parallel mesh generation methods solve the mesh generation and partitioning problems separately —this leads to an unnecessary and expensive memory access overheads, since parallel mesh generation is NOT a computation intensive application. Moreover the boundary (interfaces) of the subproblems are fixed —for some algorithms this affects the stability of the parallel mesh.
In [13, 20] we presented the first provable 3-dimensional (3D) parallel guaranteed quality Delaunay mesh (PGQDM) generation and refinement algorithm and software for polyhedral domains. The PGQDM mathematically guarantees the generation of tetrahedra with circumradius to shortest edge ratio less than 2, as long as the angle sepa-

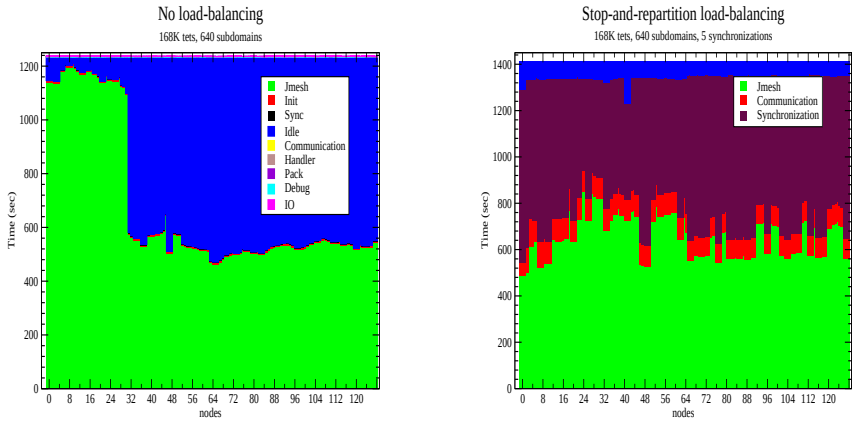


Figure 1: Performance data from parallel advance front technique [21] on 128 processor cluster. In the left graph, the blue color depicts the cycles lost due to work-load imbalance due to refinement. In the right graph, the red and magenta depict the cycles spend in communication (due to data movement) and global synchronization for traditional stop-and-repartition parallel mesh generation methods. The green color in both graphs depicts the actual computation time of the advance front mesh generation.

rating any two incident segments and/or facets is between 90^o and 270^o degrees. The PGQDM kernel is based on existing sequential algorithms [23, 9].

The PGQDM algorithm: (1) explores concurrency at both coarse-grain and fine-grain levels in order to tolerate communication and local synchronization latencies, (2) couples the mesh generation and partitioning problems into a single optimization problem in order to eliminate redundant memory operations (loads/stores) from and to cache, local & remote memory, and discs, (3) allows the submesh interfaces induced by an element–wise partitioning of an initial mesh of the domain to change as the mesh is refined in order to mathematically guarantee the quality of the elements.

Figure 2a depicts an initial decomposition of a coarse mesh which is used for data distribution purposes only. Figure 2b shows that the Bowyer-Watson [4, 25] cavity extend beyond the submesh interfaces in order to guarantee the quality of the mesh. The extension of the cavity beyond the interfaces is a source of intensive communication. However as Figure 2c shows we can tolerate almost 90% of the communication by concurrently refining other regions of the submeshes while we wait for remote data to arrive. Unfortunately, the concurrent refinement can create a number of inconsistencies in the mesh as the Figure 2d shows. These inconsistencies are resolved at the cost of setbacks and frequent message polling shown in Figure 2e. At a communication cost the PGQDM it can afford to be domain decomposition independent and it allows the distribution of the new elements as they are generated (Figure 2f). Both characteristics

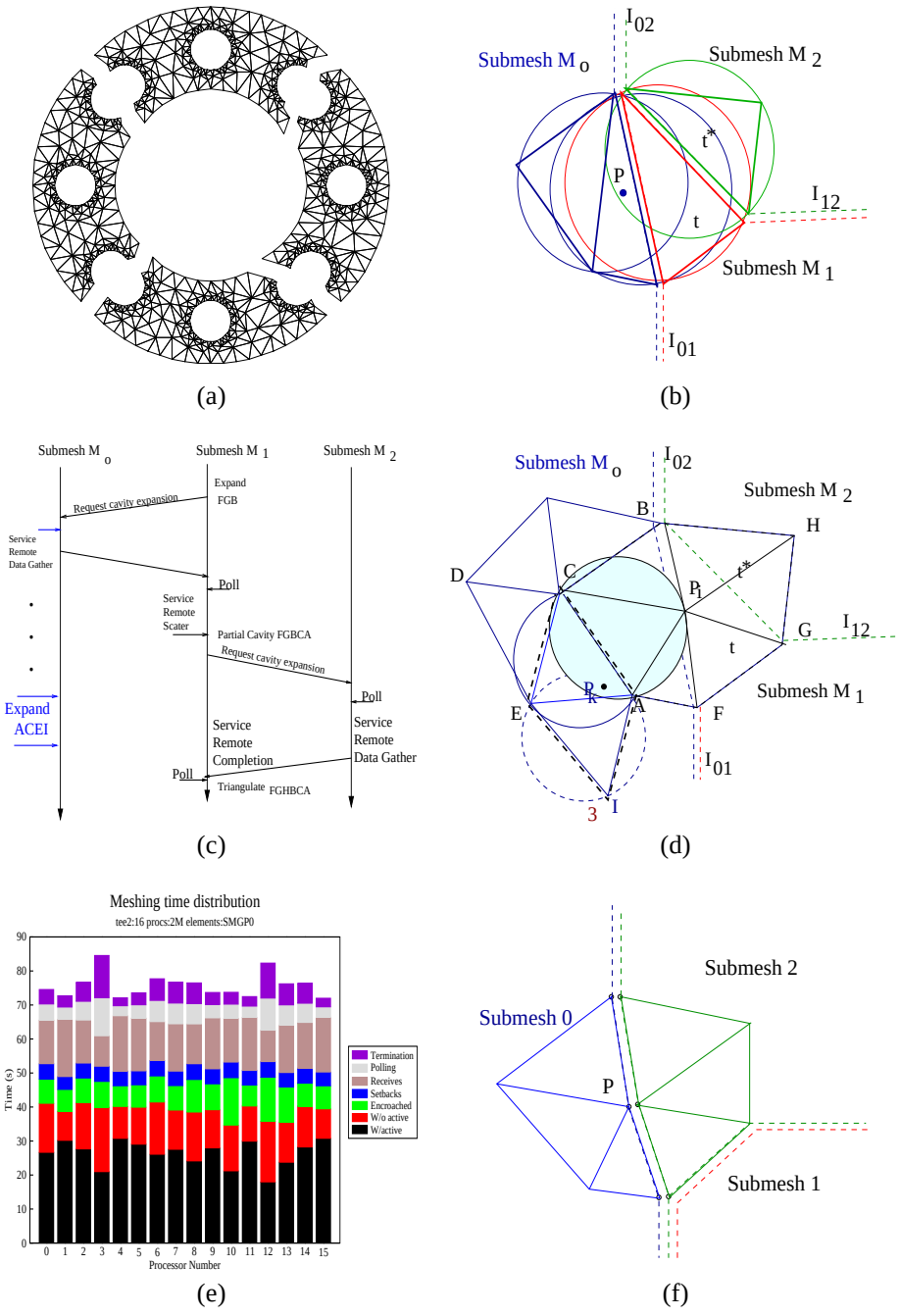


Figure 2: a) Initial data distribution, b) cavity extension beyond the submesh interfaces, c) time diagram with concurrent point insertion for tolerating communication latencies, (d) an example that depicts a source of mesh inconsistency (e) performance data that show setbacks due to elimination of mesh inconsistencies and finally (f) the refinement of a cavity with simultaneous distribution of the newly created elements.

eliminate the need to solve a challenging partitioning problem before, during or after parallel meshing.
Our performance data from a 16 node SP2 and 32 node Cluster of Sparc Workstations suggest that between 80% and 90% (in some instances even more) of the communication latency can be masked effectively at the cost of increasing communication overhead up to 20% of the total run time. Despite the increase in the communication overhead the latency-tolerant PGQDM can generate elements for parallel finite element PDE solvers many times (it depends on the size of the generated mesh) many times faster than the traditional approach which is based on the sequential generation and parallel partition and placements of elements.
In PGQDM although we mask most of the communication and synchronization latencies, the communication overhead (cycles for pushing and pulling messages from the network) is around 50% of the total runtime. In summary, PGQDM is stable, but it is communication intensive with limited code re-use. Next, I describe a stable method we developed which eliminates local synchronization and minimizes communication as it improves code re-use.

3 Parallel Constrained Delaunay Mesh Generation

Contrary to PGQDM which ignores the interface edges or faces, with the Parallel Constrained Delaunay Mesh (PCDM) generation method [7] each submesh is treated as a mesh defined by external boundary and constrained edges (or faces in 3D) which are the edges of the interfaces between any pair of adjacent submeshes. PCDM simplifies and minimizes the communication required for the parallel generation of unstructured meshes.
Intuitively, the Constrained Delaunay Mesh (CDM) is as close as one can get to the Delaunay triangulation given that one needs to handle certain (constrained) edges. This idea can be described mathematically [8] and leads to Delaunay-based mesh generators that allow internal boundaries. It has be shown [8] that these internal boundaries do not affect the quality of the resulting mesh for 2-dimensional (2D) planar domains. Although an observant user might infer the presence of such boundaries in the resulting mesh by noting the way in which triangle edges are aligned. Using the idea of a CDT, we can introduce in the mesh artificial constrained (interface) edges which decompose the mesh into submeshes that can be meshed independently.
The interfaces between submeshes must be created during a preprocessing (domain decomposition) step. The domain decomposition problem is not trivial since the interfaces must satisfy certain properties:

- The interfaces should not create any new features like segments and angles which are smaller in size than the features of the original geometry.

- The interfaces should minimize surface to volume ratio and connectivity between the submeshes.

The interfaces do not have to be simple line segments. The boundaries can be polylines or even curves, although line segments are probably sufficient for most situations. For

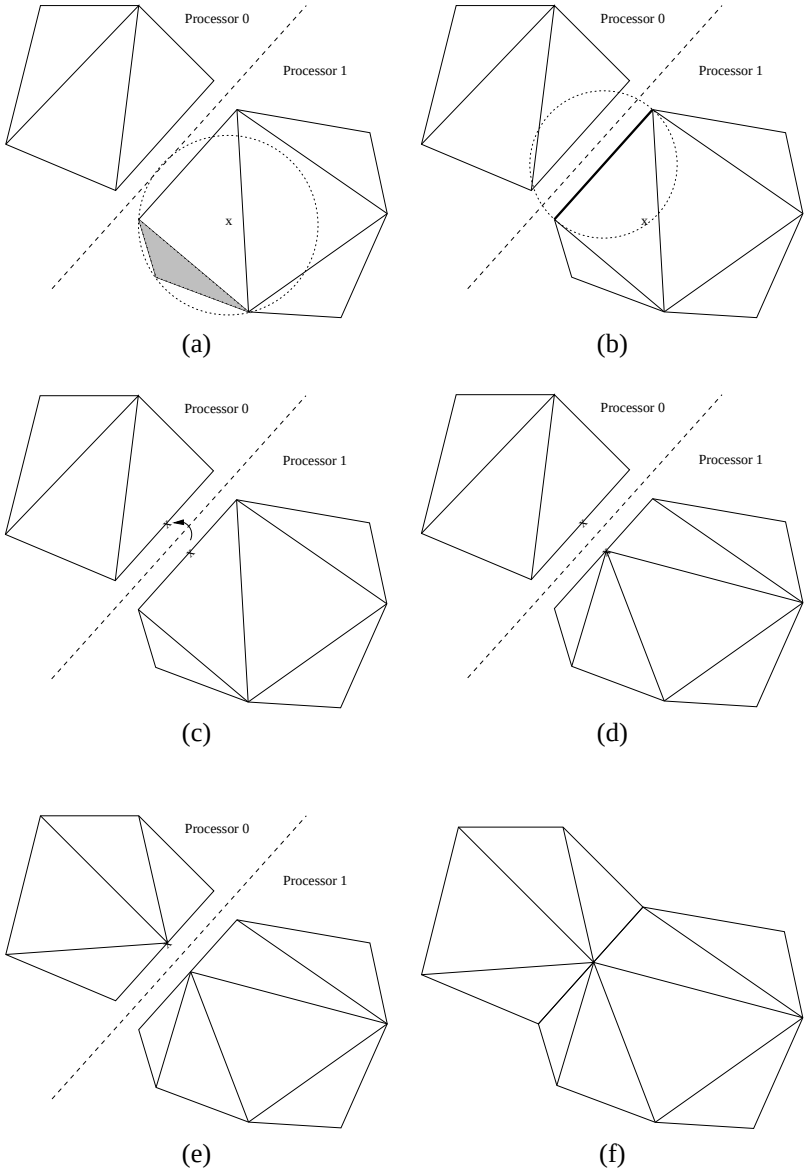


Figure 3: Processor P1 inserts a new point (a) which is encroaching upon an interface edge (b). Then P1 discards the new point and inserts the midpoint of the encroached edge(c) while at the same time it sends a request to split the same interface edge on processor P0. Processor P0 computes the cavity of the midpoint (d). The triangulation of the cavities (d) and (e) of the midpoint of the interface edge result into a new consistent and distributed Delaunay (in the CDT sense) triangulation which guarantees the quality of the elements.

planar geometries we can achieve such decomposition using Medial Axis Transform (MAT). Figure 3 depicts the MAT of the cross section of a pipe and the corresponding decomposition (see Figure 3c).

Note that, by the definition of the CDM, points on one side of the interfaces have no effect on triangles on the other side; thus, no synchronization is required during the element creation process. In addition, inter-process communication is tremendously simplified: the only message between adjacent processes is of the form, "Split this interface (i.e., constrained) edge" if a newly inserted point encroaches upon an interface edge (i.e., violates the Delaunay property of a Delaunay edge). Since interface edges always split exactly in half, no additional information needs to be communicated.

Although PCDM eliminates the synchronization and minimizes the communication of parallel guaranteed quality Delaunay mesh generation, the PCDM for 3D domains it is still an open problem. The code re-use for PCDM is not as high as we would like it to be.

4 Parallel Domain Decoupling Delaunay Mesh Generation

In order to maximize code re-use and be able to leverage from the ever evolving and mature technologies in sequential meshing we developed the Parallel Domain Decoupling Delaunay (PD^3) method [18]. PD^3 works for 2D domains and we are working for its extension on 3D domains.

The PD^3 based on the idea of decoupling the individual submeshes so that they can be meshed independently with zero communication and synchronization. In the past similar attempts to parallelize Delaunay triangulations and implement Delaunay based mesh generation presented in [3, 17]. However, in [18] we solve some of the drawbacks and improve upon the previously published methods.

The two-dimensional, divide-and-conquer Delaunay triangulation (DCDT) algorithm and its parallel implementation presented in [3] assumes that all points are known at the beginning of the triangulation. This is not a practical assumption for mesh generation and refinement, because new points are inserted on demand in order to improve element quality, perform mesh refinement, and recovery of boundary edges and faces.

The Parallel Projective Delaunay Meshing (P^2DM) method presented in [17] requires pre-processing of the domain in order to compute sequentially the proper Delaunay separators. This by itself is not a major problem since many methods require some pre-processing before the parallel mesh generation. However the PD^2M method after that pre-processing might suffer setbacks and start all-over again. The setbacks are in the form of discarding the mesh created because the separators do not always guarantee the Delaunay triangulation (i.e., they are not always Delaunay admissible) of the domain as new points are inserted [17]. The second problem is that the dynamic load balancing of the parallel mesh is based on prediction of the computational load of the processors. However, the computation of Delaunay meshing is variable and the workload prediction wihtout major corrections (during the mesh refinement) is very difficult and inaccurate.

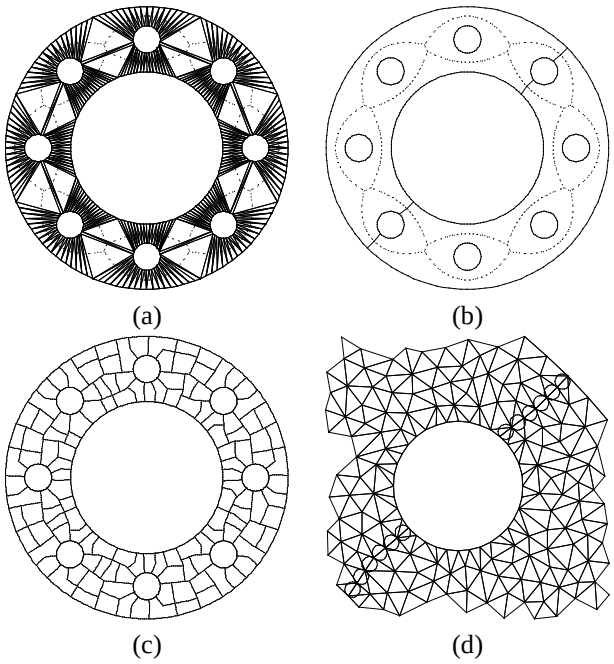


Figure 4: a) Initial boundary conforming mesh used to compute the Medial Axis Transformation (b) which in turn is used to achieve high quality domain decomposition (c). For PD^3 the interfaces of the subdomains are refined (d) in a pre-processing step.

In summary both methods have two main problems, proper decomposition of the domain in the context of adaptivity and dynamic load balancing. The construction of decompositions that can decouple the mesh and the proof that these decompositions lead to stable parallel meshes is a challenging problem. We use a domain decomposition method which is based on the partition of a weighted graph and an approximation of the MAT. Before the construction of the graph, the initial mesh is pre-processed in order to prevent the creation of "bad" angles [18] between either the interfaces themselves or between the interfaces and the external boundary of the domain. Then the pre-processed mesh is used to construct a dual weighted graph of the sides of the triangles. This graph is simply connected and its partition provides a decomposition of the domain.

After the decomposition of the domain, the *PD*^{3} constructs a "zone" around the interfaces of the submeshes. In [18] we prove that Delaunay meshers will not insert any new points within this zone i.e., the sequential Delaunay mesh on the individual submeshes it can terminate without inserting any new points on the interfaces and thus eliminate communication and code modifications of the sequential codes. So the problem of parallel meshing is reduced into a "proper" domain decomposition and a discretization of interfaces. Of course one has to show that: (1) the domain is not over-refined because of a predefined discretization and (2) the quality of the elements is maintained.

Table 1: Preliminary data on the number of elements and execution time (in seconds) for generating very large meshes using 1 to 128 processors.

Procs	1	8	16	32	48	64	128
Domains	12	96	192	384	576	768	1200
# elems	19M	150M	300M	600M	934M	1.2B	1.6B
Dec. Time	0.20	0.31	0.35	0.54	0.90	1.27	3.08
Mesh Time	124.18	131.57	132.57	135.04	136.78	135.81	106.31
Total time	124.38	131.88	132.92	135.58	137.68	137.08	109.39

Our preliminary experimental data indicate that over-refined is not a major factor of inefficiency in the geometries we tested our method. Similarly the high quality of the elements is preserved.
At the end of the parallel mesh generation process some communication is required to compute the global topology of the mesh. Finally, the PD^3 achieves 100% code re-use.

5 Parallel Implementation

Providing adequate software support for adaptive mesh refinement codes is a challenging task even for sequential implementations. The program complexity, for parallel mesh generation codes that are scalable and stable, increases by an order of magnitude, due their dynamic and irregular computation, communication and synchronization requirements. In order to handle the software complexity of managing task parallelism we developed a Portable Runtime Environment for Multiprocessor Applications (PREMA).

5.1 Parallel Runtime Environment for Multiprocessor Applications

PREMA's design objective is to assist mesh generation practitioners to develop parallel meshing codes in an evolutionary (incremental) way starting from well tested and fine-tuned sequential codes (preferable written in C or C++ programming languages). PREMA supports a Simple Mesh Partitioning Library (SMPL) that runs on a single processor and gets as input a mesh which is defined in the standard format that most of the public domain Finite Element Analysis codes use. SMPL generates as output the submeshes using the same format, but at the end appends adjacency information at the subdomain level.
The next two layers handle communication [2, 11]. The *Data Movement and Control Substrate* (DMCS) implements one-sided low-latency communication. DMCS is built on top of low-level, vendor-specific communication subsystems that bypass the operating system and access the network interface directly in order to minimize communication startup overheads. Also, it is implemented on widely available libraries like MPI for clusters of workstations and PCs. DMCS adds a small overhead (less

than 10%) to the communication operations provided by the underline communication system. In return DMCS provides a flexible and easy to understand application program interface for one-sided communication operations including *put-ops* and *remote procedure calls*. Furthermore, DMCS is designed so that it can be easily ported and maintained by non-experts. The second communication component is the Mobile Object Layer (MOL). MOL supports one-sided communication in the context of data (submesh) or object mobility. Specifically, MOL uses global logical name space and supports automatic message forwarding in order to ease the implementation of dynamic load balancing methods for adaptive applications on distributed memory machines. MOL is implemented on top of DMCS and adds another 10% to 15% overhead. In return the application programmer can perform data (submesh) or object migration, for load balancing, without changing a single line of code.

Finally, the third layer implements the dynamic Load Balancing Library (LBL) on top of DMCS and MOL. LBL supports both explicit and implicit load balancing methods. LBL uses the abstraction of *Schedulable Object* (which sometimes is called Work Unit) to represent application-defined data objects like the submeshes, for the case of discrete DD methods. The Schedulable Object (SO) is the smallest unit of granularity that is managed by the runtime system. Balancing processors workload or memory (in the case of adaptive refinement) results in moving one or more Schedulable Objects. The SOs are migrated automatically by the *Load Migrator* which is responsible for un-installing, moving, and installing SOs using MOL so that messages to them will automatically forwarded. The decision making for which SO and where to migrate is handled by the *Scheduler*. The Scheduler supports a wide spectrum of load-balancing methods and it is described in detail in [1]. In the next version of the LBL tools like *Load Monitor* and *System Monitor* will provide to the scheduler all the systems (processors and network) information it needs for effective dynamic load balancing.

By using PREMA parallel mesh practitioners can be assisted in many different levels. First, they can customize to their code requirements one-sided message passing operations like *put_op*, where op is user defined function. We found that this functionality simplifies code complexity substantially, since the application programmer does not have to concern about what to do when a message arrives. Second, they know that at some point they have to implement mesh refinement i.e., they know that they will have to move submeshes (object) among the processors for load balancing purposes, they can use MOL messages which find the object independently off its physical location. Third, the programmer can get load balancing (and in the future out-of-core implementation), for free, by using the LBL's *Scheduler*.

Finally once the parallel meshing code is build and it is ready for production, the mesh practitioners can fine tune the performance by directly using *Load Migrator* and for performing explicit load balancing and swapping of SOs (i.e., submeshes). This incremental or evolutionary approach is described in great detail in [1], for parallel Constrained Delaunay meshing. Figure 5 depicts some initial performance data from balancing a parallel advance front method on 32 processors. More and better methods need to be developed for larger processors configurations. Fortunately the plug-and-play software design of PREMA makes this task very easy. From the last 15 years of experience we have seen that there is no single dynamic load balancing method for all applications or even parallel mesh generation techniques.

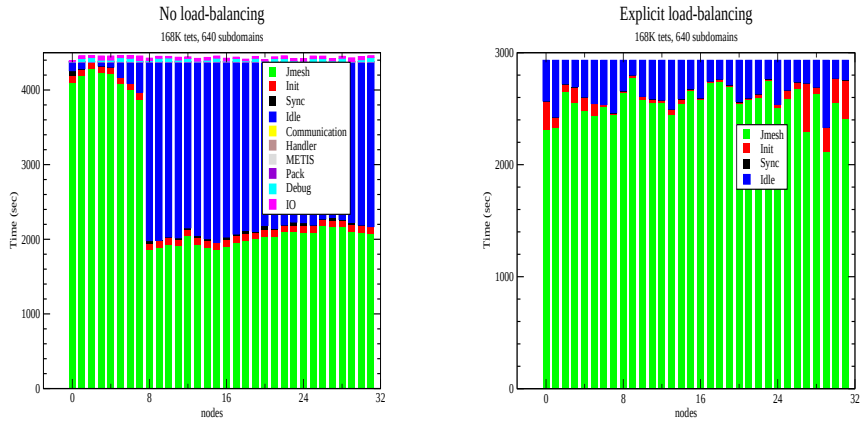


Figure 5: Performance data from parallel advance front technique [21] on 32 processor Sun cluster. The explicit method is based on work-stealing [1] and does not require global synchronization.

Conclusions

We have described three parallel guaranteed quality Delaunay mesh generation methods; the PGQDM for 3D polyhedral domains and two methods (PCDM and PD^3) for 2D planar geometries. In addition we briefly described a parallel runtime system, PREMA, we used to implement these methods. PREMA can be used to solve the dynamic load balancing problem. Thus PREMA allows us to focus on yet another very important and still open problem for parallel meshing, the domain decomposition problem.

The PGQDM is the first parallel provable 3D meshing method and it is independent of the quality of the domain decomposition. Its disadvantage are: (1) it involves complex data structures for maintaining the consistency and Delaunay properties of the distributed mesh, (2) it requires high communication which affects reduces its fixed speedup to O(logP), P is the number of processors and (3) it has zero code re-use. However, despite these difficulties it is the only method that does not dependent on domain decomposition and thus we are working [24] to alleviate some of the problems we mentioned above and extended it for 3D geometries with curved boundary.

Contrary to PGQDM the PCDM is easier to implement with an order of magnitude less communication and with linear speedup, but it depends on high quality domain decompositions. Current techniques for domain decomposition are not capable to always provide domain decompositions suitable for PCDM. We have solved the domain decomposition problem for 2D planar geometries and we are working [6] to solve this problem for 3D geometries.

Finally, the PD^3 method we believe is the “holy grail” for parallel meshing. It requires

zero communication and synchronization with 100% code re-use. Unfortunately, like PCDM requires special domain decompositions which are very challenging to generate for general geometries. For 2D planar geometries we have been very successful to generate such decompositions. We have a very good progress in applying the same techniques for 3D polyhedral geometries, but for 3D domains with curved boundary the problem is still open.

6 Acknowledgments

Many results that appear in this paper are due to my work with my students Andrey Chernikov, Andriy Fedorov, Demian Nave, Leonidas Linardakis and Chaman Verma, as well as my colleagues from Cornell University especially Dr. Paul Chew. This research has been funded by NSF CISE Challenge #EIA-9726388, Career Award CCR-9876179, Research Infrastructure grant #EIA-9972853, ITR grant #ACI-0085969 and NGS grant EIA-0203974.

References

[1] K. Barker, N. Chrisochoides, A. Chernikov, and K. Pingali. Architecture and evaluation of a load balancing framework for adaptive and asynchronous applications. *IEEE Trans. Parallel and Distributed Systems,*, Under revision, 2003.

[2] Kevin Barker, Nikos Chrisochoides, Jeff Dobbelaer, Démian Nave, and Keshav Pingali. Data movement and control substrate for parallel, adaptive applications. *Concurrency and Computation Practice and Experience*, 14, 2002.

[3] G. Blelloch, J. Hardwick, G. Miller, and D. Talmor. Design and implementation of a practical parallel delaunay algorithm. *Algorithmica*, 24:243–269, 1999.

[4] Adrian Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.

[5] Bruce Carter, Chuin-Shan Chen, L. Paul Chew, Nikos Chrisochoides, Guang R. Gao, Gerd Heber, Antony R. Ingraffea, Chris Myers Roland Krause and, Démian Nave, Keshav Pingali, Paul Stodghill, Stephen Vavasis, and Paul A. Wawrzynek. Parallel fem simulation of crack propagation – challenges, status. In *Lecture Notes in Computer Science*, volume 1800, pages 443–449. Springer-Verlag, 2000.

[6] Andrey Chernikov. *Domain Decomposition for Guaranteed Quality Parallel Mesh Generation*. PhD thesis, College of William and Mary, 2005.

[7] P. Chew, N. Chrisochoides, and F. Sukup. Parallel constrained delaunay meshing. In *Special Symposium on Trends in Unstructured Mesh Generation*. ASME/ASCE/SES, 1997.

[8] Paul Chew. Constrained delaunay triangulations. *Algorithmica*, 4:97–108, 1989.

[9] Paul Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proc. 9th Annu. ACM Sympos. Cornput. Geom.*, pages 274–280, 1993.

[10] Nikos Chrisochoides. A survey of parallel unstructured mesh generation methods. *SIAM Review*, To be submitted, 2003.

[11] Nikos Chrisochoides, Kevin Barker, Demian Nave, and Chris Hawblitzel. Mobile object layer: A runtime substrate for parallel adaptive and irregular computations. *Advances in Engineering Software*, 31(8-9):621–637, 2000.

[12] Nikos Chrisochoides and Demian Nave. Simultaneous mesh generation and partitioning. *Mathematics and Computers in Simulation*, 54(4-5):321–339, 2000.

[13] Nikos Chrisochoides and Démian Nave. Parallel delaunay mesh generation kernel. *IJNME*, To appear in 2003.

[14] Nikos Chrisochoides and Florian Sukup. Task parallel implementation of the BOWYER-WATSON algorithm. In *Proceedings of Fifth International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, 1996.

[15] H. de Cougny and M. Shephard. *CRC Handbook of Grid Generation*, chapter Parallel unstructured grid generation, pages 24.1–24.18. CRC Press, Inc., 1999.

[16] H. de Cougny and M. Shephard. Parallel volume meshing using face removals and hierarchical repartitioning. *Comp. Meth. Appl. Mech. Engng.*, 174(3-4):275–298, 1999.

[17] J. Galtier and P. L. George. Prepartitioning as a way to mesh subdomains in parallel. In *Special Symposium on Trends in Unstructured Mesh Generation*. ASME/ASCE/SES, 1997.

[18] L. Linardakis and N. Chrisochoides. Parallel delaunay domain decoupling method for planar geometries. *Algorithmica*, To be submitted in 2003.

[19] R. Lohner and J. Cebral. Parallel advancing front grid generation. In *International Meshing Roundtable*. Sandia National Labs, 1999.

[20] D. Nave, N. Chrisochoides, and P. Chew. Guaranteed-quality parallel delaunay refinement for restricted polyhedral domains. In *Proceedings of 8th ACM Symposium on Computational Geometry*, pages 135–144, 2002.

[21] J. B. Cavalcante Neto, P. A. Wawrzynek, M. T. M. Carvalho, L. F. Martha, and A. R. Ingraffea. An algorithm for three-dimensional mesh generation for arbitrary regions with cracks. *EwC*, 17(1):75–91, 2001.

[22] R. Said, N. Weatherill, K. Morgan, and N. Verhoeven. Distributed parallel delaunay mesh generation. *Comp. Methods Appl. Mech. Engrg.*, 177:109–125, 1999.

[23] Jonathan Richard Shewchuk. *Delaunay Refinement Mesh Generation*. PhD thesis, Carnegie Mellon University, School of Computer Science, May 1997. Available as Technical Report CMU-CS-97-137.

[24] Chaman Verma. Parallel guaranteed quality delaunay mesh generation for 3d geometries with curved boundaries. Master's thesis, College of William and Mary, 2003.

[25] David F. Watson. Computing the n-dimensional delaunay tessellation with application to voronoi polytopes. *The Computer Journal*, 24(2):167–172, 1981.

AMOVING GRIDS FOR HYPERBOLIC PROBLEMS

v1.1 3/28/2003

KONSTANTIN LIPNIKOV[¶], MIKHAIL SHASHKOV[¶]

[¶]Los Alamos National Laboratory
MS B284, Los Alamos, NM, 87545, U.S.A.

ABSTRACT: A few methods for minimizing error functionals via redistribution of mesh points are considered. Some equivalence relations between these methods are analyzed both theoretically and numerically. Effects of the data interpolation, grid smoothing, and time integration for 1D Burgers equation are studied in details.

KEY WORDS: moving grids, Burgers equation.

Introduction

In practical applications of finite difference methods to numerical solution of PDEs, an appropriately chosen mesh will produce sufficiently accurate results with less mesh elements. Nowadays, the great number of engineering computations rely on ability to construct grids adapted to the solution peculiarities. In this paper we consider an adaptive strategy based on a redistribution of the mesh nodes (r-adaptation). This strategy is most appealing in a numerical solution of time dependent problems where the adaptive grid changes slightly during one step of a time integration.

A natural way to define an adaptive grid is via a minimization problem for an error functional. Given an initial conformal mesh, we minimize the error functional by moving the mesh points without breaking the mesh topology. In many cases, a uniform error distribution is achieved on the resulting grid.

The idea of linking the constrained minimization problem to the principle of error equidistribution has been considered to some extend by many researchers. In [9], Pereyra and Sewell equidistributed a local truncation error of finite different methods in order to improve error estimates for these methods. An appropriate grading of a mesh for piecewise polynomial interpolation problem has been considered by Carey and Dinh in [3]. They approach has been further developed in [4]. Steinberg and Roache have shown in [11] that a constrained minimization problem for a certain quadratic functional has an equidistribution solution. Baines has shown in [2] that the least square minimization of the residual of the divergence of a vector field is equivalent to an equidistribution of the L_2-norm of the residual. The general discussion of mesh generation algorithms based on the methods mentioned above can be also found in [1, 8].

The exact evaluation of error functionals is either impossible or quite costly. As the result, adaptive grids are generated in such a way to resolve steep gradients of discrete

[¶]lipnikov@t7.lanl.gov
[¶]shashkov@lanl.gov

solutions. This approach is based on a monitor function involving a few problem dependent parameters (see e.g. [7, 12] and references therein). Since the mesh points are mainly concentrated in regions where sharp changes of the solutions occur, the adaptive grids may not be appropriate for solving time dependent problems due to strong disproportionality of neighboring mesh cells resulting in essential lose of accuracy. The problem was resolved by Dorfi and Drury in [6]. The authors proposed an algorithm for guaranteed spatial and temporal smoothing of monitor functions.

In this paper we advocate the method based on a sufficiently accurate evaluation of error functionals followed by the guaranteed smoothing of the associated monitor function. The connection between the error functional and the smoothed monitor function is established by Proposition 1.1. We shall prove that under certain conditions, errors introduced by both data interpolation algorithms and time integration schemes are of higher order than the approximation error.

The outline of the paper is as follows. In Section 1 we describe the principle of the error equidistribution in the case of piecewise constant approximations and compare it theoretically and numerically with direct minimization algorithms. In Section 2 we study effects of a monitor smoothing on accuracy of the solution. In Section 3 we consider the one-dimensional Burgers equation and show that the temporal error introduced by a time integration scheme can be ignored in mesh adaptation algorihms.

1 Piecewise constant approximations

Let $f(x)$ be a given sufficiently smooth function over the unit interval $[0, 1]$. We consider a conformal grid $\{x_i\}$:

$$0 = x_0 < x_1 < \ldots < x_{M+1} = 1$$

and define $x_{i+1/2} = (x_{i+1} + x_i)/2$ and $h_{i+1/2} = x_{i+1} - x_i$. Let $f^h(x)$ be a piecewise constant function given by

$$f^h(x) = a_{i+1/2}, \qquad x \in [x_i, x_{i+1}], \tag{1.1}$$

where $i = 0, \ldots, M$. We introduce an error functional

$$\Phi(\{x_i\}, \{a_{i+1/2}\}) = \int_0^1 \Big(f(x) - f^h(x)\Big)^2 \,\mathrm{d}x = \sum_{i=0}^{M} \int_{x_i}^{x_{i+1}} \big(f(x) - a_{i+1/2}\big)^2 \,\mathrm{d}x. \tag{1.2}$$

For a given grid $\{x_i\}$, a piecewise constant function $f^h(x)$ minimizing the functional Φ has to satisfy the following conditions: $\partial\Phi/\partial a_{i+1/2} = 0$ for $i = 0, \ldots, M$. The straightforward calculations give

$$a_{i+1/2} = \bar{f}_{i+1/2} \equiv \frac{1}{h_{i+1/2}} \int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x \tag{1.3}$$

where $\bar{f}_{i+1/2}$ denotes the integral average of function $f(x)$ on interval $[x_i, x_{i+1}]$. Additivity of the integration implies that the discrete function $f^h(x)$ defined by (1.1) and (1.3) preserves the integral value of the continuous function $f(x)$, i.e.

$$\int_0^1 f^h(x)\,\mathrm{d}x = \int_0^1 f(x)\,\mathrm{d}x.$$

The above formulas allow us to reduce the original minimization problem (1.2) to a more simple one of minimizing only with respect to mesh points:

$$\min_{x_1,\ldots,x_M} F_{ex}(\{x_i\}), \qquad F_{ex}(\{x_i\}) = \sum_{i=0}^{M} \int_{x_i}^{x_{i+1}} \left(f(x) - \bar{f}_{i+1/2}\right)^2 \mathrm{d}x. \tag{1.4}$$

Using the Taylor expansion with the Lagrange remainder, it is easy to show that the minimization problem (1.4) is equivalent to the following one:

$$\min_{x_1,\ldots,x_M} F_{ex}(\{x_i\}), \qquad F_{ex}(\{x_i\}) = \frac{1}{12}\sum_{i=0}^{M} \left(\left.\frac{\partial f}{\partial x}\right|_{x^*_{i+1/2}}\right)^2 h^3_{i+1/2} \tag{1.5}$$

where $x^*_{i+1/2}$ is a point from interval (x_i, x_{i+1}).
If all integrals in (1.4) can be computed exactly, the problem can be solved by methods from [1] (see references therein). We shall show below that in the one-dimensional case the minimization problem (1.4) is reduced to a nonlinear elliptic problem. The solution of this problem will equally distribute the L_2-norm of the error $f - f^h$ over the mesh cells.

Proposition 1.1. *Let $e_{i+1/2}(\cdot,\cdot)\colon \Re^2 \to \Re$, $i = 0,\ldots,M$, be a set of functions defined by*

$$e_{i+1/2}(x_i, x_{i+1}) = \int_{x_i}^{x_{i+1}} g(x)\,\mathrm{d}x, \qquad 0 \le x_i \le x_{i+1} \le 1,$$

where $g(x) \ge 0$ is an arbitrary bounded function. Then, for any positive integer p,

$$\min_{x_1,\ldots x_M} \sum_{i=0}^{M} e^p_{i+1/2}(x_i, x_{i+1}) = \frac{\mathcal{E}^p}{(M+1)^{p-1}}$$

where

$$\mathcal{E} = \sum_{i=0}^{M} e_{i+1/2}(x_i, x_{i+1}) = \int_0^1 g(x)\,\mathrm{d}x.$$

Moreover, the minimum is achieved when $e_{i+1/2}(x_i, x_{i+1}) = \mathcal{E}/(M+1)$, $i = 0,\ldots,M$, i.e. $e_{i+1/2}$ is constant.

The proof of Proposition 1.1 is given in Appendix. Let us apply the assertion of this proposition to minimization problem (1.5). We introduce additional notations:

$$\hat{\omega}_{i+1/2} = \left(\left. \frac{\partial f}{\partial x} \right|_{x^*_{i+1/2}} \right)^{2/3} \qquad \text{and} \qquad \hat{e}_{i+1/2} = \hat{\omega}_{i+1/2} h_{i+1/2}$$

where $i = 0, \ldots, M$. Using the above notations, we can rewrite the functional F_{ex} as follows:

$$F_{ex}(\{x_i\}) = \sum_{i=0}^{M} \hat{e}^3_{i+1/2} = \sum_{i=0}^{M} \hat{\omega}^3_{i+1/2} h^3_{i+1/2}. \tag{1.6}$$

It is obvious that

$$\hat{e}_{i+1/2} \to \int_{x_i}^{x_{i+1}} \left| \frac{\partial f}{\partial x} \right|^{2/3} \mathrm{d}x \qquad \text{and} \qquad \sum_{i=0}^{M} \hat{e}_{i+1/2} \to \int_0^1 \left| \frac{\partial f}{\partial x} \right|^{2/3} \mathrm{d}x$$

as $\max\limits_i h_{i+1/2} \to 0$. Thus, the functions $\hat{e}_{i+1/2} = \hat{e}_{i+1/2}(x_i, x_{i+1})$ satisfy approximately the requirements of Proposition 1.1 with $g(x) = |\partial f/\partial x|^{2/3}$. As shown by numerous numerical experiments, this discrepancy can be neglected in many applications. Then, Proposition 1.1 implies that the grid satisfying

$$\hat{e}_{i+1/2} = \hat{e}_{i-1/2}, \qquad i = 1, \ldots, M, \tag{1.7}$$

gives value of the functional F_{ex} close to its minimal value. However, the corresponding grids may be quite different, especially coarse ones.

The formula (1.7) is known as the principle of the error equidistribution. Let us rewrite it as follows:

$$\hat{\omega}_{i+1/2}(x_{i+1} - x_i) - \hat{\omega}_{i-1/2}(x_i - x_{i-1}) = 0. \tag{1.8}$$

It is easy to observe that (1.8) can be interpreted as a discretization of a non-linear elliptic equation

$$\frac{\partial}{\partial \xi} \left(\omega(x) \frac{\partial x}{\partial \xi} \right) = 0, \qquad x(0) = 0, \quad x(1) = 1, \tag{1.9}$$

on uniform grid $\xi_i = i\Delta\xi$, $\Delta\xi = 1/(M+1)$, in the reference coordinate $\xi \in [0, 1]$. According to (1.5) the coefficient $\omega(x)$ is given by

$$\omega(x) = \left| \frac{\partial f}{\partial x} \right|^{2/3}. \tag{1.10}$$

In order to guarantee the well posedness of problem (1.9), $\omega(x)$ should be regularized. In the sequel we assume that $\omega(x)$ is replaced by $\omega(x) + \varepsilon$ where ε is a small positive number, e.g. $\varepsilon = 10^{-12} \max\limits_x \omega(x)$. Following [7, 12], we shall call the resulting $\omega(x)$ as the monitor function.

Remark 1.1. We want to emphasize here that small positive number ε is not an adjustable parameter. In contrast to [6, 7, 12], it is used only to avoid numerical degeneracy of the elliptic equation.

It is not surprising that some discrete analog of (1.9) can be directly derived from $\nabla F_{ex} = 0$, the necessary condition on extrema of functional (1.4). Indeed, the straightforward differentiation of F_{ex} with respect to x_i gives

$$\frac{\partial F_{ex}}{\partial x_i} = 2f(x_i) - \bar{f}_{i-1/2} - \bar{f}_{i+1/2} = 0. \tag{1.11}$$

Using the Taylor expansions at point x_i and neglecting terms of third order and higher, we get

$$\left.\frac{\partial f}{\partial x}\right|_{x_i} \left(h_{i+1/2} - h_{i-1/2}\right) + \frac{1}{3}\left.\frac{\partial^2 f}{\partial x^2}\right|_{x_i} \left(h_{i+1/2}^2 + h_{i-1/2}^2\right) = 0.$$

Since $h_{i+1/2} \approx (\partial x/\partial\xi)_{i+1/2}\Delta\xi$, the last equation is a second order approximation of the equation

$$\frac{\partial f}{\partial x}\frac{\partial^2 x}{\partial \xi^2} + \frac{2}{3}\frac{\partial^2 f}{\partial x^2}\left(\frac{\partial x}{\partial \xi}\right)^2 = 0, \qquad x(0) = 0, \quad x(1) = 1, \tag{1.12}$$

which coincides with (1.9), (1.10) on a subset of $(0, 1)$ where $\partial f/\partial x \neq 0$.

Remark 1.2. The conditions similar to (1.11) can be found for two dimensional problems providing a natural way for deriving a system of partial differential equations for the mesh deformation. This will be the topic of our future research.

1.1 Approximate error functional

The above considerations assume the knowledge of either a continuous function $f(x)$ or its derivatives. However, the input data are usually given in a form of a discrete piecewise constant function $\{\bar{f}^0_{i+1/2}\}$ corresponding to the grid $\{x^0_i\}$. In this case, the grid adaptation requires data interpolation and estimates for derivatives. Since the interpolation procedure introduces an additional error, the optimization problem (1.4) has to be replaced by a more general problem:

$$\min_{x_1,\ldots,x_M} \int_0^1 \left| f(x) - \left[R_h(f^{h,0})\right](x)\right|^2 \mathrm{d}x \tag{1.13}$$

where R_h is an interpolation operator from the initial grid $\{x^0_i\}$ to a final grid $\{x_i\}$ and $f^{h,0}$ is the given piecewise constant function:

$$f^{h,0}(x) = \bar{f}^0_{i+1/2} = \frac{1}{h^0_{i+1/2}} \int_{x^0_i}^{x^0_{i+1}} f(x)\,\mathrm{d}x, \qquad x \in [x^0_i, x^0_{i+1}], \qquad i = 0, \ldots, M.$$

In order to define the interpolation operator R_h, we consider a piecewise linear function $g^{h,0}(x)$ reconstructed from $f^{h,0}(x)$:

$$g^{h,0}(x) = \bar{f}^0_{i+1/2} + s^0_{i+1/2}(x - x^0_{i+1/2}), \qquad x \in [x^0_i, x^0_{i+1}], \qquad i = 0, \ldots, M.$$

We define the limiter function

$$\operatorname{minmod}(a,b) = \begin{cases} 0 & \text{if} \quad ab \le 0, \\ \operatorname{sign}(a)\min(a,b) & \text{otherwise} \end{cases}$$

and use it to select slopes $s^0_{i+1/2}$ for $g^{h,0}(x)$:

$$s^0_{i+1/2} = \operatorname{minmod}\left(\frac{\bar{f}^0_{i+3/2} - \bar{f}^0_{i+1/2}}{x^0_{i+3/2} - x^0_{i+1/2}}, \quad \frac{\bar{f}^0_{i+1/2} - \bar{f}^0_{i-1/2}}{x^0_{i+1/2} - x^0_{i-1/2}} \right).$$

Let $\tilde{f}^h(x) = \left[R_h(f^{h,0})\right](x)$ be a piecewise constant function given by

$$\tilde{f}^h(x) = \tilde{f}_{i+1/2} = \frac{1}{h_{i+1/2}} \int_{x_i}^{x_{i+1}} g^{h,0}(x)\,\mathrm{d}x, \qquad x \in [x_i, x_{i+1}], \tag{1.14}$$

where $i = 0, \ldots, M$. It is obvious that the interpolation operator defined by the above formulas is conservative, i.e. it preserves the integral value of function $f^{h,0}(x)$. Since R_h is exact for linear functions, we have the following estimate:

$$\tilde{f}_{i+1/2} = \bar{f}_{i+1/2} + O(h^2_{i+1/2}) \tag{1.15}$$

where $\bar{f}_{i+1/2}$ is the unknown exact integral average of function $f(x)$ on the grid $\{x_i\}$. Therefore

$$\int_0^1 \left(f(x) - \left[R_h(f^{h,0})\right](x) \right)^2 \mathrm{d}x = \sum_{i=0}^{M} \int_{x_i}^{x_{i+1}} \left(f(x) - \bar{f}_{i+1/2} + O(h^2_{i+1/2}) \right)^2 \mathrm{d}x$$

$$= \sum_{i=0}^{M} \left[\frac{1}{12} \left(\left. \frac{\partial f}{\partial x} \right|_{x^*_{i+1/2}} \right)^2 h^3_{i+1/2} + O(h^4_{i+1/2}) \right].$$

Neglecting high order terms in the last formula, we get the functional F_{ex} from (1.5). Thus, the error introduced by the interpolation operator R_h can be ignored. In the sequel we shall analyze the minimization problem (1.5) with the functional given by (1.6). Since the precise computation of coefficients $\hat{\omega}_{i+1/2}$ is usually impossible, they are replaced by computable coefficients $\omega_{i+1/2}$ such that $\omega_{i+1/2} \approx \hat{\omega}_{i+1/2}$, $i = 0, \ldots, M$. The Taylor expansion for $f - f^h$ on a grid being intersection of the initial $\{x^0_i\}$ and current $\{x_i\}$ grids gives

$$\omega_{i+1/2} = \frac{1}{h^3_{i+1/2}} \sum_{k=0}^{M} \int_{\check{x}_{ik}}^{\hat{x}_{ik}} \left| \bar{f}^0_{k+1/2} + \left[\frac{\delta f^{h,0}}{\delta x} \right]_{k+1/2} (x - x_{k+1/2}) - \tilde{f}^0_{i+1/2} \right|^2 \mathrm{d}x \tag{1.16}$$

where $[\hat{x}_{ik}, \check{x}_{ik}] = [x_i, x_{i+1}] \cap [x^0_k, x^0_{k+1}]$ and $\delta f^{h,0}/\delta x$ is an approximation of the first derivative of function $f(x)$. The formula (1.16) implies that a first order approximation

of $\partial f/\partial x$ is sufficient. However, for numerical computations we use the second order approximation of $\partial f/\partial x$ given by

$$\left[\frac{\delta f}{\delta x}\right]_{i+1/2} = \frac{1}{h_{i+1/2}}\left(\left[\frac{\delta f}{\delta x}\right]_{i+1}\left(\frac{h_{i+1/2}}{6}+\frac{h_{i-1/2}}{3}\right)+\left[\frac{\delta f}{\delta x}\right]_{i}\left(\frac{h_{i+1/2}}{6}+\frac{h_{i+3/2}}{3}\right)\right)$$

where

$$\bar{h}_{i+1/2} = \frac{1}{3}(h_{i-1/2}+h_{i+1/2}+h_{i+3/2})$$

and

$$\left[\frac{\delta f}{\delta x}\right]_{i} = \frac{\bar{f}_{i+1/2}-\bar{f}_{i-1/2}}{(h_{i+1/2}+h_{i-1/2})/2}. \tag{1.17}$$

Note that we lose one order of accuracy, if we replace the exact values $\bar{f}_{i+1/2}$ by their second order approximations $\tilde{f}_{i+1/2}$.
The formula (1.16) is one of the possible quadrature rules for approximating the exact error. It implies that the original (exact) minimization problem (1.5) is replaced by an approximate one:

$$\min_{x_1,\ldots,x_M} F_{ap}(\{x_i\}), \qquad F_{ap}(\{x_i\}) = \sum_{i=0}^{M} \omega_{i+1/2}^3 h_{i+1/2}^3 \tag{1.18}$$

where both $\omega_{i+1/2}$ and $h_{i+1/2}$ depend on the grid. The formula (1.16) is the most natural approximation of the exact coefficients $\hat{\omega}_{i+1/2}$. We assume implicitly that the initial grid is fine enough to resolve the behavior of function $f(x)$. However, for completeness of the numerical analysis, we consider another way to approximate the exact coefficients. Assuming that the function $f(x)$ is known, we set

$$\bar{\omega}_{i+1/2} = \left|\frac{\partial f}{\partial x}(x_{i+1/2})\right|^{2/3} \qquad \text{and} \qquad \bar{F}_{ap}(\{x_i\}) = \sum_{i=0}^{M} \bar{\omega}_{i+1/2}^3 h_{i+1/2}^3.$$

1.2 Minimization algorithms

In the numerical experiments we use a steepest descent method for the direct minimization of (1.4). The method is based on representation (1.6) with very precise computation of coefficients $\hat{\omega}_{i+1/2}$ (see Algorithm 1). The same algorithm may used for minimization problem (1.18).

Algorithm 1. **`[Direct Minimization]`** For $k = 1,\ldots,K_{max}$ do

1. For the given grid $\{x_i^k\}$ compute values $\omega_{i+1/2}^k$, $i = 0,\ldots,M$, using formulas (1.16).
2. Perform one Gauss-Seidel sweep

$$\min_{x_i^{k+1}} \left\{ \left[\hat{\omega}_{i+1/2}^{k+1}(x_{i+1}^k - x_i^{k+1})\right]^3 + \left[\hat{\omega}_{i-1/2}^{k+1}(x_i^{k+1} - x_{i-1}^{k+1})\right]^3 \right\},$$

where $i = 1,\ldots,M$, R_h is the interpolation operator from grid $\{x_i^k\}$ to grid $\{x_i^{k+1}\}$, and $\hat{\omega}^{h,k+1} = R_h(\omega^{h,k})$.

3. Stop iterations if $\max_i |x_i^k - x_i^{k+1}| \le TOL$ where TOL is the user given tolerance.

We tried different numerical schemes for the direct minimization of our functionals. The most stable results were obtained when we explicitly enforced the constrain

$$\sum_{i=0}^{M} \omega_{i+1/2}^k h_{i+1/2}^k = \sum_{i=0}^{M} \hat{\omega}_{i+1/2}^{k+1} h_{i+1/2}^{k+1}, \qquad \hat{\omega}^{h,k+1} = R_h(\omega^{h,k}),$$

in the second step of Algorithm 1 using the conservative interpolation operator R_h. Proposition 1.1 implies that instead of the direct minimization of the above functionals, we can use formulation (1.9) and numerical algorithms for solving non-linear elliptic problems. In the numerical experiments we use Algorithm 2.

Algorithm 2. **`[Equidistribution Principle]`** For $k = 1, \dots, K_{max}$ do

1. For the given grid $\{x_i^k\}$ compute values $\omega_{i+1/2}^k$, $i = 0, \dots, M$, using formulas (1.16).
2. Perform one Gauss-Seidel sweep

$$\omega_{i+1/2}^k (x_{i+1}^k - x_i^{k+1}) - \omega_{i-1/2}^k (x_i^{k+1} - x_{i-1}^{k+1}) = 0, \qquad i = 1, \dots, M.$$

3. Stop iterations if $\max_i |x_i^k - x_i^{k+1}| \le TOL$ where TOL is the user given tolerance.

1.3 Numerical experiments

Let us consider a test function $f(x)$ given by

$$f(x) = 1 - \frac{9r_1 + 5r_1^5}{10(r_1 + r_1^5 + r_2)}, \quad r_1 = \exp\frac{1/2 - x}{20\varepsilon}, \quad r_2 = \exp\frac{3/8 - x}{2\varepsilon}, \qquad (1.19)$$

with $\varepsilon = 0.005$. The solutions of minimization problems formulated above are shown in Fig. 1 and 2. We use notation

$$E(\{x_i\}) = \sqrt{F_{ex}(\{x_i\})}$$

for the L_2 norm of the error $f(x) - f^h(x)$ on grid $\{x_i\}$ (see Table 1). Let $\{x_i^{ex}\}$, $\{x_i^{ap}\}$ and $\{\bar{x}_i^{ap}\}$ be the grids minimizing functionals F_{ex}, F_{ap} and $\bar{F}_{ap}$, respectively. Since the steepest descent and equidistribution methods produce different grids, we use additionally superscripts 'st' and 'eq', respectively, for the resulting grids. Finally, the superscript 'un' stands for uniform grids. We shall use similar notations for the corresponding piecewise constant functions, e.g. $f^{h,ex}(x)$ denotes the best approximation to $f(x)$ on the grid $\{x_i^{ex}\}$.

It is obvious that in the case of piecewise constant approximation, the L_2 norm of the error is of the first order. However, optimal distribution of mesh points gives us 3 times smaller error. The solutions of problems (1.4) and (1.18) are compared on Fig. 1. On

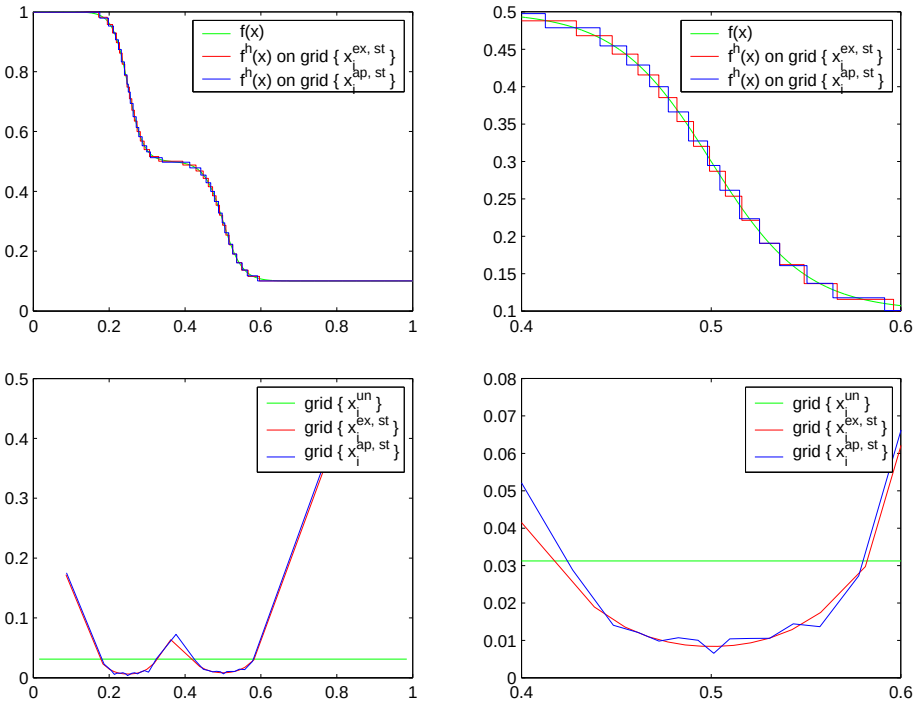


Figure 1: Minimization of exact and approximate functionals for $M = 32$: functions (on the top), mesh steps $h_{i+1/2}$ as the function of $x_{i+1/2}$ (on the bottom).

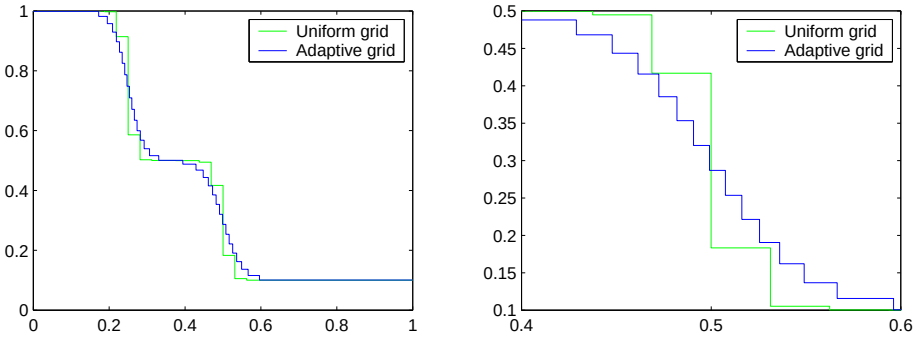


Figure 2: Piecewise constant approximations on uniform and adaptive grids for $M = 32$.

M	$E(\{x_i^{un}\})$	$E(\{x_i^{ex,st}\})$	$E(\{x_i^{ap,st}\})$	$E(\{x_i^{ap,eq}\})$
16	2.99e-2	1.01e-2	1.19e-2	1.19e-2
32	1.59e-2	4.99e-3	5.22e-3	5.22e-3
64	7.99e-3	2.48e-3	2.51e-3	2.51e-3
128	4.00e-3	1.24e-3	1.24e-3	1.24e-3

Table 1: Comparison of different minimization strategies.

the top, we plot the function $f(x)$ and its piecewise constant approximations. On the bottom, we plot the mesh steps $h_{i+1/2}$ as the function of $x_{i+1/2}$. The principle of the error equidistribution results in a solution which visually coincides with the solution of problem (1.18).
We observe that the distribution of mesh nodes is not a smooth function for the solution of problem (1.18). The deviation from the exact solution is due to high order terms which were neglected in derivation of quadrature rule (1.16). This problem can be solved either by mesh refinement or by using a better quadrature rule.
In order to emphasize importance of the adaptation, we compare solutions on adaptive grids with similar solutions on uniform grids (see Fig. 2).
The Algorithm 1 diverged for minimization problems with functional $\bar{F}_{ap}$. The main reason is that the replacement of $x^*_{i+1/2}$ by $x_{i+1/2}$ is not justified for grids with very big intervals (see Fig 1).

2 Grid smoothing

The algorithms described above produce grids optimal from approximation view point. The mesh points are concentrated in regions where sharp changes of the function occur. However, these grids may not be appropriate for solving time dependent problems due to strong disproportionality of neighboring mesh cells. Stability considerations usually require smooth change in concentration of mesh points. The remedy can be found in a guaranteed spatial smoothing of the monitor function $\omega(x)$ which will indirectly smooth the resulting grid due to relation (1.7). The objective here is to smooth the optimal grid while preserving its main features.
In this paper we follow the idea of the guaranteed smoothing proposed in [6]. Let α be a positive number. We define a smoothed grid as the grid whose mesh steps satisfy the following condition:

$$\frac{\alpha}{\alpha+1} \le \frac{h_{i-1/2}}{h_{i+1/2}} \le \frac{\alpha+1}{\alpha}, \qquad i = 1,\dots,M. \tag{2.1}$$

Recall that in the equidistribution principle ratio of neighboring mesh steps equals to ratio of values of the monitor function, i.e. $h_{i-1/2}/h_{i+1/2} = \omega_{i+1/2}/\omega_{i-1/2}$.

Proposition 2.1. *Let $\omega_{i+1/2}$, $i = 0,\dots,M$, be given values of a monitor function. The*

values $\tilde{\omega}_{i+1/2}$, $i = 0,\dots,M$, *of a smoothed monitor function satisfying*

$$\frac{\alpha}{\alpha+1} \le \frac{\tilde{\omega}_{i+1/2}}{\tilde{\omega}_{i-1/2}} \le \frac{\alpha+1}{\alpha}, \qquad i = 1,\dots,M,$$

can be computed by solving the system of $M+1$ *linear equations:*

$$\tilde{\omega}_{i+1/2} - \alpha(\alpha+1)(\tilde{\omega}_{i+3/2} - 2\tilde{\omega}_{i+1/2} + \tilde{\omega}_{i-1/2}) = \omega_{i+1/2}, \tag{2.2}$$

where $\tilde{\omega}_{-1/2} = \omega_{1/2}$ *and* $\tilde{\omega}_{M+3/2} = \omega_{M+1/2}$.

The proof of Proposition 2.1 is given in Appendix. The proposition provides a simple way for merging the functional minimization with the grid smoothing. Recall that Proposition 1.1 establish equivalence between solution of the system of nonlinear equations (1.9) and the minimization of functional F_{ex}. In other words, for a given monitor function, we may write down a functional minimizing which we get a grid with desired properties. The functional is the result of smoothing of coefficient in formula (1.16):

$$\min_{x_1,\dots,x_M} \tilde{F}_{ap}(\{x_i\}), \qquad \tilde{F}_{ap}(\{x_i\}) = \sum_{i=0}^{M} \tilde{\omega}^3_{i+1/2} h^3_{i+1/2} \tag{2.3}$$

Note that the smoothing algorithm may be generalized to higher dimensions.

Algorithm 3. **`[Direct minimization with smoothing]`** For $k = 1,\dots,K_{max}$ do

1. For the given grid $\{x_i^k\}$ compute values $\omega^k_{i+1/2}$, $i = 0,\dots,M$, using formulas (1.16).
2. Compute the smoothed values $\tilde{\omega}^k_{i+1/2}$, $i = 0,\dots,M$, by solving system (2.2).
3. Perform one Gauss-Seidel sweep

$$\min_{x_i^{k+1}} \left\{ \left[\hat{\omega}^{k+1}_{i+1/2} (x^k_{i+1} - x_i^{k+1}) \right]^3 + \left[\hat{\omega}^{k+1}_{i-1/2} (x_i^{k+1} - x^{k+1}_{i-1}) \right]^3 \right\},$$

   where $i = 1,\dots,M$, R_h is the interpolation operator from grid $\{x_i^k\}$ to grid $\{x_i^{k+1}\}$, and $\hat{\omega}^{h,k+1} = R_h(\tilde{\omega}^{h,k})$.
4. Stop iterations if $\max_i |x_i^k - x_i^{k+1}| \le TOL$ where TOL is the user given tolerance.

Proposition 2.1 allows us to derive an estimate for the minimal mesh step $h_{i+1/2}$. Let us consider the extreme case when the grid is geometrically refined to the point $x = 1$ in such a ways that

$$h_{i+1/2} = h_{1/2} \left(\frac{\alpha}{\alpha+1} \right)^i, \qquad i = 0,\dots,M.$$

This immediately implies that the minimal mesh step is given by

$$h_{M+1/2} = \frac{\alpha^M}{(1+\alpha)^{M+1} - \alpha^{M+1}}.$$

In order to study the effect of smoothing, we consider again the test function (1.19). The results of numerical experiments are shown in Table 2 and in Fig. 3. We use superscript '*sm*' for grids generated by Algorithm 3. Note that the loss in the accuracy of the solution is about 3%. In Fig. 3 we compare results of the exact minimization problem (1.4) with similar results for the approximate problem (2.3). The neighboring mesh steps satisfy inequality (2.1) with $\alpha = 1$. Observe that the original and smoothed monitors are close to each other. However, the oscillations in function $h_{i+1/2}(x)$ clearly observed on Fig. 1 have been dumped by the smoothing procedure.

It is pertinent to note that on a continuous level the smoothing and differentiation operations are commute. Therefore, instead of smoothing the coefficients $\omega_{i+1/2}$ on each loop of Algorithm 1, we may smooth once the function $f^{h,0}$ given on the initial grid $\{x_i^0\}$. The results of numerical experiments are shown in the last column of Table 2. Unfortunately, this approach provides less control over distribution of mesh points; in particular, inequality (2.1) is no longer valid since the discrete smoothing and differentiation operations are not commute.

3 Burgers equation

Let us consider the one-dimensional viscous Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \varepsilon\frac{\partial^2 u}{\partial x^2}, \qquad x \in (0,1), \qquad t \in (0,T), \tag{3.1}$$

subject to initial condition

$$u(x,0) = u_0(x)$$

and some boundary conditions specified later.

In order to solve problem (3.1) numerically, we use the explicit time integration schemes and grids moving according to algorithms from previous sections. Let Δt^n be the time step, $t^n = \sum_{i=1}^{n} \Delta t^i$, and $\bar{u}_{i+1/2}^n$ be the integral average of $u(x,t^n)$ on interval (x_i^n, x_{i+1}^n).

We assume that the data associated with the grid $\{x_i^n\}$ are given and exact. Let $\bar{u}^{n+1} = \mathcal{L}^n(\bar{u}^n)$ be an explicit time integration scheme. A possible error accumulation due to errors in the initial data will be addressed in the future publications. A new grid may be built in such a way to minimize an error functional F_{ex}^n representing the L_2-norm of the error $u(x,t^{n+1}) - \bar{u}^{n+1}$. As we have mentioned above, this grid may result in unstable computations and should be smoothed. The methodology elaborated in previous sections suggests to replace the functional F_{ex}^n by an approximate functional F_{ap}^n via smoothing of the monitor function ω^n associated with F_{ex}^n. Let the grid $\{x_i^{ap,n}\}$ minimize the functional F_{ap}^n and R_h be the interpolation operator R_h from $\{x_i^n\}$ onto $\{x_i^{ap,n}\}$ defined in Section 1.1. Then, the solution at time step t^{n+1} is computed as $\bar{u}^{n+1} = \mathcal{L}^n(R_h(\bar{u}^n))$. The time integration step is completed by setting $\{x_i^{n+1}\} \equiv \{\tilde{x}_i^n\}$.

At the end of this section, we analyze the error functional associated with the simplest

M	$E(\{x_i^{ap,st}\})$	$E(\{\tilde{x}_i^{ap,st,sm}\})$	$E(\{\tilde{x}_i^{ap,st,sm}\})$
16	1.19e-2	1.62e-2	1.68e-2
32	5.22e-3	6.01e-3	6.50e-3
64	2.51e-3	2.68e-3	2.66e-3
128	1.24e-3	1.28e-3	1.25e-3

Table 2: Effects of smoothing strategies.

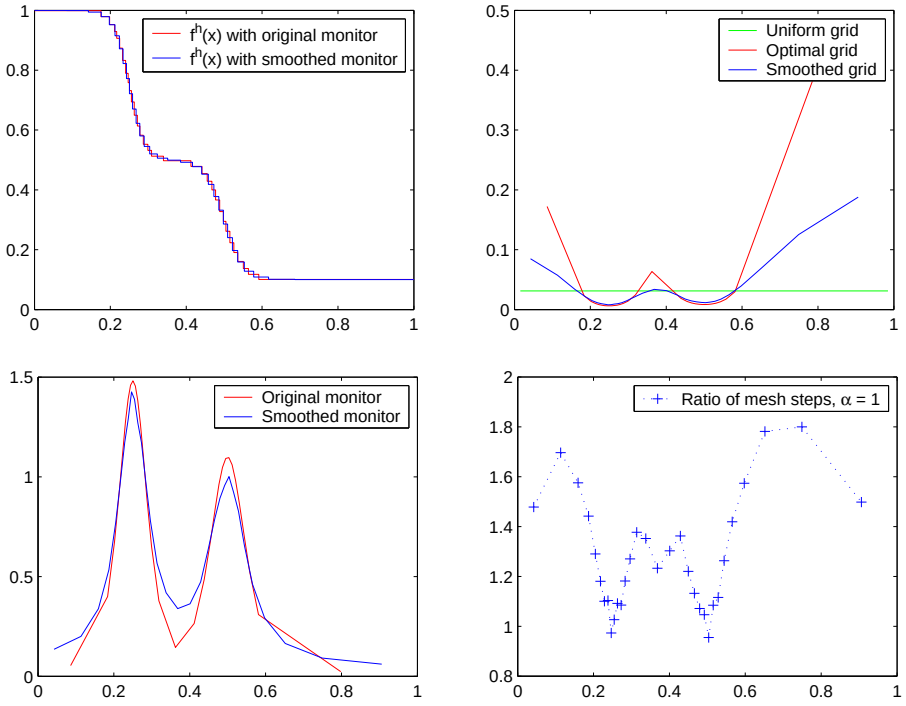


Figure 3: Steepest descent methods with and without smoothing for $M = 32$ and $\alpha = 1$: solutions (top-left), mesh steps $h_{i+1/2}$ as the function of $x_{i+1/2}$ (top-right), monitors $\omega(x)$ and $\tilde{\omega}(x)$ (bottom-left), and ratio of neighboring mesh steps (bottom-right).

donor scheme:

$$\begin{aligned}\bar{u}^{n+1}_{i+1/2} &= \mathcal{L}^n_{i+1/2}(\bar{u}^n)\\ &= \bar{u}^n_{i+1/2} - \frac{\Delta t^n}{h^n_{i+1/2}}(f^n_{i+1} - f^n_i) + \frac{\varepsilon \Delta t^n}{h^n_{i+1/2}}\left(\left[\frac{\delta u^n}{\delta x}\right]_{i+1} - \left[\frac{\delta u^n}{\delta x}\right]_i\right)\end{aligned} \tag{3.2}$$

where $\delta u^n/\delta x$ is defined by (1.17) and f^n_i denotes the flux at point x^n_i,

$$f^n_i = \frac{1}{2}\begin{cases} (\bar{u}^n_{i-1/2})^2 & \text{if } \bar{u}^n_{i+1/2} + \bar{u}^n_{i-1/2} \geq 0,\\ (\bar{u}^n_{i+1/2})^2 & \text{otherwise.}\end{cases}$$

The stability conditions [10] impose limitations on the time step,

$$\min_i \Delta t^n \left(\frac{\bar{u}^n_{i+1/2}}{h^n_{i+1/2}} + \frac{2\varepsilon}{(h^n_{i+1/2})^2}\right) < 1. \tag{3.3}$$

Without loss of generality we assume that $\bar{u}^n_{i+1/2} + \bar{u}^n_{i-1/2} \geq 0$ at all mesh points and consider the following minimization problem

$$\min_{x^n_1,\ldots,x^n_M} F^n_{ex}(\{x^n_i\}) \tag{3.4}$$

where

$$\begin{aligned}F^n_{ex}(\{x^n_i\}) &= \sum_{i=0}^{M} \int_{x^n_i}^{x^n_{i+1}} (u(x, t^{n+1}) - \bar{u}^{n+1}_{i+1/2})^2 \,\mathrm{d}x\\ &= \sum_{i=0}^{M} \int_{x^n_i}^{x^n_{i+1}} (u(x, t^{n+1}) - \mathcal{L}^n_{i+1/2}(\bar{u}^n))^2 \,\mathrm{d}x\end{aligned}$$

The stability condition (3.3) implies that $\bar{u}^n_{i+1/2}\Delta t^n < h^n_{i+1/2}$ and $\varepsilon \Delta t^n < (h^n_{i+1/2})^2/2$. Using the Taylor expansions at point x^n_i and neglecting terms of second order and higher in $h^n_{i+1/2}$, we get

$$\begin{aligned}&\int_{x^n_i}^{x^n_{i+1}} \left(u(x, t^{n+1}) - \mathcal{L}^n(\{\bar{u}^n_{i+1/2}\})\right)^2 \mathrm{d}x = \\ &\left(\left.\frac{\partial u}{\partial x}\right|_{x^n_{i+1/2}}\right)^2 \left[\frac{(h^n_{i+1/2})^3}{12} + \frac{(\Delta t^n u^n_{i+1/2})^2}{4}\frac{(h^n_{i+1/2} - h^n_{i-1/2})^2}{h^n_{i+1/2}}\right] + O(h^n_{i+1/2})^4.\end{aligned} \tag{3.5}$$

Recall that in the case when the mesh step $h^n_{i+1/2}$ is sufficiently big, the last formula should be replaced by a more accurate quadrature. The non-linear elliptic equation (1.9) implies that

$$h_{i+1/2} = \Delta\xi \left.\frac{\partial x}{\partial \xi}\right|_{i+1/2} + O(\Delta\xi^2) = \frac{c\Delta\xi}{\omega(x_{i+1/2})} + O(\Delta\xi^2)$$

where c is a constant depending on the boundary conditions and $\Delta\xi = 1/(M+1)$. Thus, for the case of smooth monitor functions, the difference between $h_{i+1/2}$ and $h_{i-1/2}$ is of order $\Delta\xi^2$. Since similar speculations may be applied to any grid obtained via a smooth mapping, the first term in the right hand side of (3.5) is the dominating term. We performed many numerical experiments and all of them confirmed this conclusion. After omitting the second term, we obtain a minimization problem (1.18) for function $u(x, t^n)$ and grid $\{x_i^n\}$. Thus, even for lower order time integration schemes, the temporal error introduced by the numerical scheme can be ignored.

3.1 Numerical experiments

In this section we study numerically efficiency of adaptive moving grids.

In the first set of experiments we consider the viscous Burgers equation. Let $T = 0.9$ and $\varepsilon = 0.005$ in equation (3.1). The problem has an exact solution when the initial condition $u_0(x)$ is given by the function (1.19). In Fig. 4 we plot the accuracy

$$E(\{x_i^N\}) = \left[\sum_{i=0}^{M} \int_{x_i^N}^{x_{i+1}^N} (u(x,T) - \bar{u}_{i+1/2}^N)^2 \, dx \right]^{1/2}$$

at the final time $T = t^N$ versus the problem size (left picture) and trajectories of the mesh points for $M = 32$ (right picture). Note that for the given accuracy 10^{-3} we need 10 times less mesh points in the adaptive grid than in the uniform one. This difference grows when ε tends to zero, i.e., when the solution profile at time $t = T$ becomes more sharp (see Fig. 5).

It is typical for one-dimensional hyperbolic problems that the arithmetical complexity for a fixed accuracy is less for uniform grids than for adaptive ones. In two and three dimensions it is highly possible that the solution of a hyperbolic problem is anisotropic and the directions of the anisotropy are not aligned with axes of the Cartesian coordinate system. In this case the convergence rates for uniform and adaptive grids will be different. In other words the slops of functions in Fig. 4 and 5 will be different resulting in computational efficiency of adaptive grids.

In the second set of experiments we consider the inviscous Burgers equation, i.e. $\varepsilon = 0$. Let $T = 0.5$ and the initial condition be the periodic function

$$u_0(x) = 0.5 + \sin(2\pi x).$$

A shock wave is developed with time and reaches its maximal strength at $t_0 = 0.25$. The computed solution is compared with the exact one in Fig. 6. Note that the donor scheme (3.2) smoothes the discrete solution before and after the shock. A better resolution of the shock may be obtained with higher order time integration schemes. The trajectories of mesh points shown in Fig. 6 confirm the theoretically know result that the shock has a constant velocity 0.5.

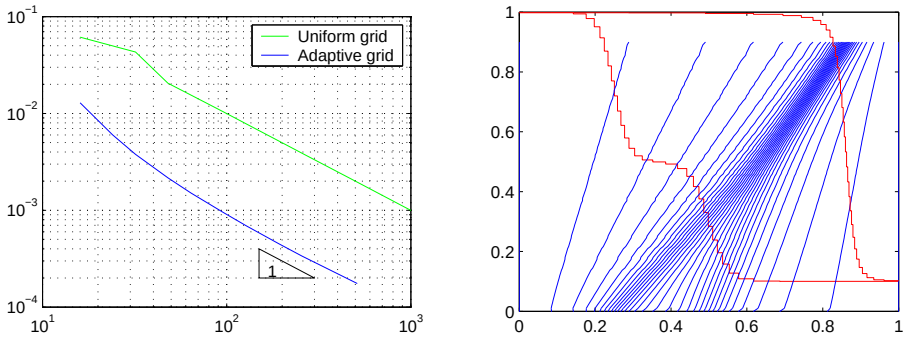


Figure 4: Viscous Burgers equation with $\varepsilon = 0.005$: convergence rate (on the left), trajectories of mesh points for the smoothing parameter $\alpha = 1$ and discrete solutions at times $t = 0$ and $t = T$ (on the right).

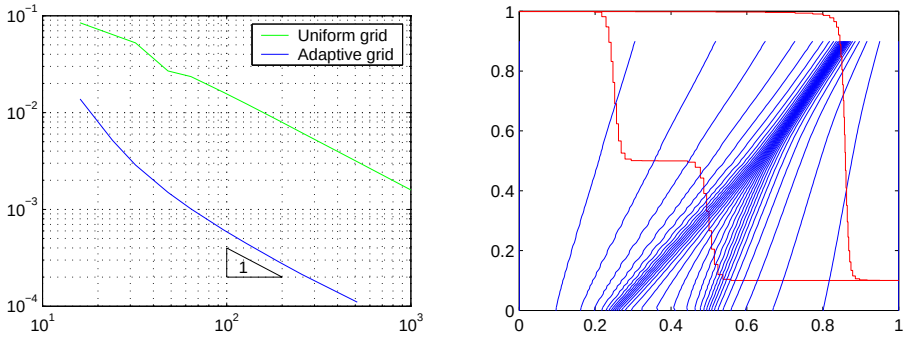


Figure 5: Viscous Burgers equation with $\varepsilon = 0.002$: convergence rate (on the left), trajectories of mesh points for the smoothing parameter $\alpha = 1$ and discrete solutions at times $t = 0$ and $t = T$ (on the right).

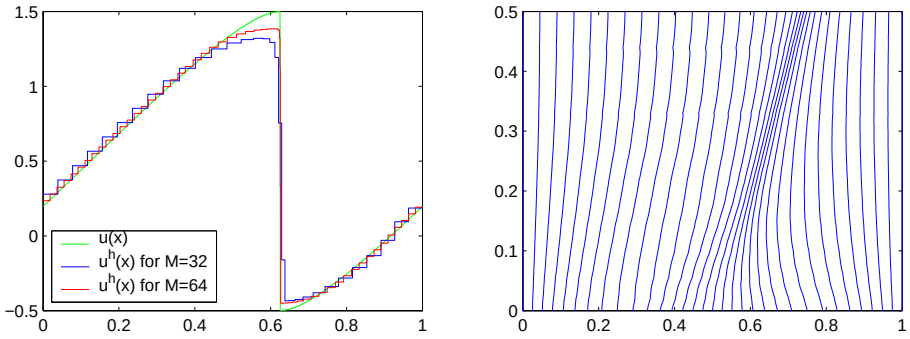


Figure 6: Inviscous Burgers equation: the discrete and exact solutions at $t = t_0$ (on the left), trajectories of mesh points for the smoothing parameter $\alpha = 2$ and $M = 32$ (on the right).

References

[1] M. BAINES *Moving Finite Elements,* Oxford University Press, New York, 1994.

[2] M. BAINES Least squares and approximate equidistribution in multidimensions. *Numer. Meth. P.D.E.*, 15:605–615, 1999.

[3] G. CAREY & H. DINH Grading functions and mesh redistribution. *SIAM J. Numer. Anal.*, 22(5):1028–1040, 1985.

[4] K. CHEN Error equidistribution and mesh adaptation. *SIAM J. Sci. Comp.*, 15:798–818, 1994.

[5] C. CONCUS, G. GOLUB & G. MEURANT Block preconditioning for the conjugate gradient method. *SIAM J. Sci. Statist.*, 6:220–252, 1985.

[6] E. DORFI & L. DRURY Simple adaptive grids for 1-d initial value problems. *J. Comput. Phys.*, 69:175–195, 1987.

[7] W. HUANG Variational mesh adaptation: isotropy and equdistribution. *J. Comput. Phys.*, 174:903–924, 2001.

[8] V. LISEIKIN *Grid Generation Methods*. Springer, Berlin, Heidelberg, New York, 1999.

[9] V. PEREYRA & E. SEWELL Mesh selection for discrete solution of boundary problems in ordinary diferential equations. *Numerische Mathematik*, 23:261–268, 1975.

[10] A. SAMARSKII *The Theory of FDifference Schemes*. Marcel Dekker, New York, 2001.

[11] S. STEINBERG & P. ROACHE Variational grid generation. *Numer. Meth. P.D.E.*, 2:71–9, 1986.

[12] H.-Z. TANG & T. TANG Moving mesh methods for one- and two-dimensional hyperbolic conservation laws. *to apper in SINUM*, 2003.

Appendix

Proof of Proposition 1.1.

Proof. It is obvious that there is a surjective mapping from the space $\{(x_1,\dots,x_M)\colon x_i < x_{i+1}\}$ of admissible point positions to the space $\{(e_{1/2},\dots,e_{M+1/2})\colon \sum_{i=0}^{M} e_{i+1/2} = E\}$ of admissible errors. In other words, for arbitrary set of admissible errors, there is at least one grid corresponding them. Therefore,

$$\min_{x_1,\dots,x_M} \mathcal{F}(\{e_{i+1/2}\}) = \min_{e_{1/2},\dots,e_{M+1/2}} \mathcal{F}(\{e_{i+1/2}\}), \qquad \mathcal{F}(\{e_{i+1/2}\}) = \sum_{i=0}^{M} e_i^p.$$

To minimize the latter functional, we introduce a Lagrange multiplier λ and define

$$\tilde{\mathcal{F}}(\{e_{i+1/2}\},\lambda) = \mathcal{F}(\{e_{i+1/2}\}) + \lambda\left(\sum_{i=0}^{M} e_{i+1/2} - \mathcal{E}\right).$$

Setting all partial derivatives of $\tilde{\mathcal{F}}$ to zero gives us a system of $M+2$ algebraic equations with a solution $e^*_{i+1/2} = \mathcal{E}/(M+1)$, $i = 0,\dots,M$, and $\lambda^* = p\mathcal{E}^{p-1}/(M+1)^{p-1}$. Since the Hessian $\mathcal{H}$ of functional $\mathcal{F}$ is a positive definite diagonal matrix, $\mathcal{H} = p(p-1)\,\mathrm{diag}\{e_{1/2}^{p-2},\dots,e_{M+1/2}^{p-2}\}$, the above solution minimizes the functional. Since $e^*_{i+1/2}$ is a constant, calculation of the $\mathcal{F}(\{x^*_{i+1/2}\})$ is trivial. □

Proof of Proposition 2.1.

Proof. Let us rewrite (2.2) in a matrix form $A\tilde{\omega} = \omega$. Since A is the M-matrix, all entries of A^{-1} are positive numbers. Therefore,

$$\frac{\tilde{\omega}_{i+1/2}}{\tilde{\omega}_{i-1/2}} = \frac{\sum_{j=0}^{M} \omega_{j+1/2}(A^{-1})_{ij}}{\sum_{j=0}^{M} \omega_{j+1/2}(A^{-1})_{i-1,j}} \le \max_j \frac{(A^{-1})_{ij}}{(A^{-1})_{i-1,j}}$$

Replacing the maximum by the minimum, we can estimate $\tilde{\omega}_{i+1/2}/\tilde{\omega}_{i-1/2}$ from below. Both estimates are easily proved by the induction principle.
It is proved in [5] that the entries of A^{-1} are strictly decreased along each row or column and can be computed by a recursive algorithm. In particular, $(A^{-1})_{ij} = u_i v_j$ for $i \le j$ and $(A^{-1})_{ij} = u_j v_i$ for $i \ge j$ where

$$u_0 = 1, \quad u_1 = \frac{1+\beta}{\beta}, \quad u_i = \frac{(1+2\beta)u_{i-1} - \beta u_{i-2}}{\beta}, \quad i = 2,\dots,M,$$
$$v_M = \frac{1}{(1+\beta)u_M - \beta u_{M-1}}, \quad v_i = \frac{1+\beta u_i v_{i+1}}{\beta u_{i+1}}, \quad i = M-1,\dots,1,$$

$v_0 = (1+\beta v_1)/(1+\beta)$, and $\beta = \alpha(\alpha+1)$. Moreover, $u_i > u_{i-1}$ and $v_i < v_{i-1}$ for $i = 1, \ldots, M$. The assertion of the proposition is proved by induction. If $i \leq j$, we get

$$\frac{(A^{-1})_{ij}}{(A^{-1})_{i-1,j}} = \frac{u_i}{u_{i-1}} \leq \frac{1+2\beta}{\beta} - \frac{u_{i-2}}{u_{i-1}}, \quad i > 1.$$

It is easy to check that $1 \leq u_1/u_0 \leq (1+\alpha)/\alpha$. Let us assume that

$$\frac{u_{i-1}}{u_{i-2}} \leq \frac{\alpha+1}{\alpha}.$$

for some $i \leq j$. Then, straightforward computations show that the same estimate holds for u_i/u_{i-1}. If $i > j$, we get

$$\frac{(A^{-1})_{ij}}{(A^{-1})_{i-1,j}} = \frac{v_i}{v_{i-1}} = \frac{u_{i+1}}{u_i}\,\frac{1+\beta u_i v_{i+1}}{1+\beta u_{i-1} v_i} \geq \frac{u_{i+1}}{u_i} \min\left\{1,\ \frac{u_i v_{i+1}}{u_{i-1} v_i}\right\} \geq \frac{v_{i+1}}{v_i}$$

since $u_{i+1} > u_i > u_{i-1}$ and $v_i/v_{i-1} < 1$. The proof of the proposition is completed by observing that $v_M/v_{M-1} \geq \beta/(\beta+1) \geq \alpha/(\alpha+1)$. □

ON SIMPLEX SHAPE MEASURES
WITH EXTENSION FOR ANISOTROPIC MESHES

JULIEN DOMPIERRE[*], MARIE-GABRIELLE VALLET[†],
PAUL LABBÉ[§] & FRANÇOIS GUIBAULT[‡]

[*†§‡]Centre de recherche en calcul appliqué (CERCA)
5160, boul. Décarie, bureau 400,
Montréal (Québec), H3X 2H9, Canada.

[‡] Département de génie informatique, École Polytechnique de Montréal,
Case postale 6079, succ. Centre-ville,
Montréal (Québec), H3C 3A7, Canada.

ABSTRACT: This paper surveys and analyses several simplex shape measures documented in the literature and used for mesh adaptation and mesh optimization. The paper first summarizes important properties of simplices and their degeneracies in Euclidean space. Each shape measure is then defined and validated with respect to a proposed shape measure validity criterion. Extensions to Riemannian spaces are proposed in order to deal with anisotropic meshes. A visualization scheme is also presented, which helps compare shape measures to one another.

KEY WORDS: triangle, tetrahedron, simplex shape measures, unstructured meshes, mesh optimization, anisotropy, metric.

Introduction

SHAPE measures provide an effective quantitative mean of evaluating the quality of the elements in a mesh which is of great relevance in mesh adaptation and mesh optimization. Still, while most serious work in the field of mesh adaptation directly make use of shape measures [18, 26, 27], very little work has been devoted to the actual comparison of shape measures, with the notable exceptions of Liu and Joe [22, 23, 24, 25] who have thoroughly analyzed selected shape measures, early work by Parthasarathy *et al.* [29], and recent work by Georges and Frey [18, 15].

While these published works present some of the standard shape measures in current use, new shape measures steadily appear in recent literature for which no analysis is available. Furthermore, no classification scheme has been proposed, and fitness of new measures is often not assessed.

This paper aims to survey a wider range of shape measures in general use, to define validity criteria for those measures and to classify them in broad categories, beginning with valid shape measures. The paper also addresses issues regarding the

[*]julien@cerca.umontreal.ca
[†]vallet@cerca.umontreal.ca
[§]paul@cerca.umontreal.ca
[‡]francois.guibault@polymtl.ca

use of shape measures in non-Euclidean spaces in order to be applicable to anisotropic meshes. Extensions are thus provided for the use of shape measures in Riemannian spaces.

The first section of the paper summarizes important properties of simplices and introduces a classification of simplex degeneracies in two and three dimensions. The next section introduces shape measures validity criteria, presents a wide range of shape measures, and explains how to use them in a Riemannian space which enables their use for anisotropic meshes. The third section presents a visualization scheme that helps analyze and compare shape measures to one another. Implications of the choice of a shape measure into a mesh optimization process is both theoretically and practically explored. Conclusions are drawn on the pertinence of developing new shape measures or choosing one among the currently existing ones.

1 Simplices

This section summarizes some well known properties of simplices, and introduces some often neglected notions about simplex degeneracies that will be used as foundations for the classification of shape measures and their comparison.

1.1 Definition of a Simplex

A *simplex*, in a space of dimension d, is the convex hull of $d+1$ vertices, and represents the simplest element that can be used to discretize space in that dimension. In practice, a simplex is a triangle in two dimensions and a tetrahedron in three dimensions. A more general discussion on simplices can be found in [15].

1.2 Regular Simplices

A regular simplex is an equilateral triangle in two dimensions and an equilateral tetrahedron in three dimensions. Equilaterality implies that all edges have the same length, which relies on the notion of distance, or in a general sense, on the notion of a metric. By changing the metric tensor used in the metric, it is possible to generate anisotropic meshes. In this way, anisotropic meshes are made up of equilateral elements when measured using the appropriate metric tensor.

Appendices B and C summarize a number of useful formulæ that relate properties of triangles and tetrahedra to their edge lengths. These relations will be used in Section 2 to define the simplex shape measures in Euclidean space. Possible extensions to Riemannian spaces will also be discussed.

1.3 Degenerate Simplices

A simplex is *degenerate* if its vertices are all included in a subspace. That is, a d-simplex is degenerate if its $d+1$ vertices do not span space $\mathbb{R}^d$. This means that a triangle is degenerate when the three vertices are collinear, or *a fortiori* when they collapsed into a single point. A tetrahedron is degenerate when its four vertices are

coplanar, or *a fortiori* when they are collinear or collapsed. In practice, degeneration detection is replaced by a test on area or volume. A triangle is degenerate if its area vanishes, and a tetrahedron is degenerate if its volume vanishes.

The most systematic way of classifying simplex degeneracies is not to consider the process that leads to the degeneracy but rather the final configuration of the degenerate simplex. In order to systematically classify simplex degeneracies, the following notation, inspired from astronomical atlases, is used: the four symbols  ,  ,  and  will respectively be used to denote vertices of multiplicity one, two, three and four.

Tables 1 to 3 presents simplex degeneracies. For each type of degeneracy, a name inspired from the literature (mostly Bern *et al.* [2, 3]) is used when available, and degenerate cases not yet identified are given distinctive names.

Table 1: The three degeneracies of triangle: *NDE* is the number of degenerate edges, and r_K is the circumradius limit.

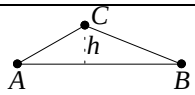
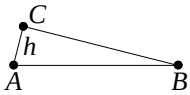
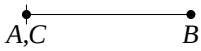
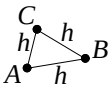
Name	$h \longrightarrow 0$	$h = 0$	*NDE*	r_K
Cap			0	∞
Needle			1	$\frac{h_{max}}{2}$
Big Crunch			3	0

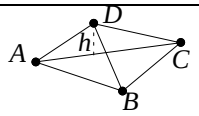
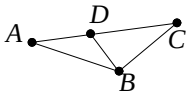
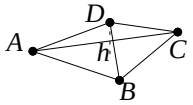
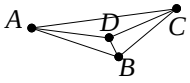
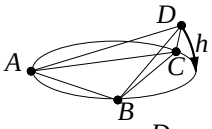
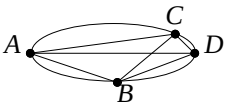
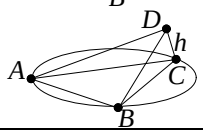
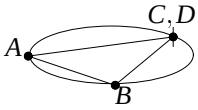
Table 1 lists the three degenerate cases for the triangle. Each case is illustrated before and at degeneracy, and for each case, the number of short edges and the final circumcircle radius are listed. These informations are central to the analysis of shape measures, as they are often used to detect degenerate cases, and are thus included in the computation of various shape measures. In the needle degeneration case, the listed circumcircle radius $h_{max}/2$ corresponds the half-length of the two non-degenerate edges of the degenerate triangle.

Table 2 presents the four types of planar tetrahedral degeneracies and Table 3 presents the five types of linear degeneracies, and the punctual tetrahedral degeneracy.

2 Shape Measures of Simplices

The concept of shape measure is widely used in the mesh adaptation literature, but while the use of this notion is quite spread, few papers actually define and compare various shape measures one to another. In this regard, papers by Liu and Joe [22, 24, 25], and especially reference [23] stand as landmark work on the topic. Other reviews

Table 2: The four planar degeneracies of tetrahedron: *NDE* is the number of degenerate edges, $\triangle$ is the type of degenerate triangular faces (c =cap, n =needle), r_K is the limit of the circumsphere radius and r_{ABC} is the circumcircle radius of triangle ABC.

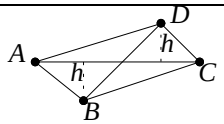
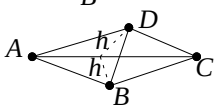
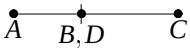
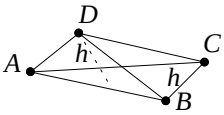
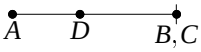
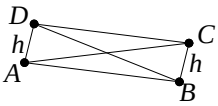
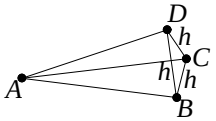
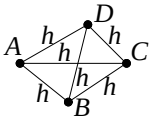
Name	$h \longrightarrow 0$	$h = 0$	*NDE*	$\triangle$	r_K
Fin			0	$1c$	∞
Cap			0	0	∞
Sliver			0	0	r_{ABC} or ∞
Wedge			1	$2n$	r_{ABC}

of tetrahedral shape measures appear as well in George and Borouchaki [18] and Parthasarathy *et al.* [29]. The underlying idea behind all simplex shape measures is to define a way to quantify the shape of a simplex. In particular, this quantity should be optimal for the regular simplex and it should be sensitive to all simplex degeneracies. There are so many simplex shape measures in the literature that we suggest the following global definition, introduced in Dompierre *et al.* [10] and derived form Liu and Joe [23]:

Definition 2.1. *A* **simplex shape measure** *is a continuous function that evaluates the shape of a simplex. It must be invariant under translation, rotation, reflection and valid uniform scaling of the simplex. It must be maximum for the regular simplex and it must be minimum for all degenerate simplices. For the ease of comparison, it should be scaled to the interval* $[0,1]$*, and be* 1 *for the regular simplex and* 0 *for a degenerate simplex.*

In this definition, the invariance by translation, rotation and reflection makes the measure independent from any coordinate system. The invariance by a valid uniform scaling of the simplex makes it independent from any measurement units. This is not considered to be a degenerate tetrahedron until the volume as actually reached zero. Also under this definition, continuity requirements are to ensure that if a simplex is close to the regular simplex, then the simplex shape measure should be close to 1. Conversely, if a simplex is close to being degenerate, then the simplex shape measure should be close to 0. Any simplex shape measure that satisfies Definition 2.1 is a *valid*

Table 3: The five linear degeneracies of tetrahedron and the punctual degeneracy: *NDE* is the number of degenerate edges, $\triangle$ is the type of degenerate triangular faces (c =cap, n =needle, BC=Big Crunch), r_K is the limit of the circumsphere radius.

Name	$h \longrightarrow 0$	$h = 0$	*NDE*	$\triangle$	r_K
Crystal			0	$4c$	∞
Spindle			1	$2c$ $2n$	∞
Splitter			1	$2c$ $2n$	∞
Slat			2	$4n$	$\frac{h_{max}}{2}$
Needle			3	$3n$ $1BC$	$\frac{h_{max}}{2}$
Big Crunch			6	$4BC$	0

simplex shape measure, and any criterion considered as a simplex shape measure but that does not satisfy Definition 2.1 is said to be an *invalid* simplex shape measure.

When the area of a triangle or the volume of the tetrahedron is negative, the simplex is more than degenerate, it is inverted. In this case, some simplex shape measures return a negative number, and some others return a positive number because, for example, they depend on the square of the area or of the volume. Due to the invariance by reflection, any valid simplex shape measure returns a positive number. Simplex shape measures should not be used to determine whether a simplex is inverted. Rather, if a mesh contains inverted simplices, the mesh optimizer should try to remove them by optimizing the volume of the tetrahedra or the area of the triangles, by minimizing the sum of the absolute value of the volumes or of the areas as proposed by Coupez [9]. In the context of inverted simplices, the shape of the inverted simplex is irrelevant, what matters is its size.

A review of simplex shape measures is presented in the following subsections. The shape measures are grouped in broad categories according to the fundamental

measures involved in their computation. Each shape measure is classified as valid or invalid according to Definition 2.1. A short discussion follows to extend these measures to the case of anisotropic meshes.

2.1 Linear Transformation Based Shape Measures

Many authors introduce shape criteria constructed from the matrix involved in the linear transformation of a simplex K to a regular reference simplex. While the choice of the defining parameters of the reference simplex (in terms of vertex numbering, coordinates, etc.) is somewhat arbitrary and impacts on the transformation matrix N, the product $N^T N$ does not depend of those parameters. Indeed, all these matrices induce the same metric tensor $\mathcal{M}_K$ (see Eq. (A.4)), which transforms simplex K into an equilateral simplex. This metric tensor is uniquely defined, and can also be determined by solving a linear system of equations that expresses the constraint whereby each edge of the simplex K is brought to unit length.

Liu and Joe [23] have defined the transformation matrix M in a simplex K of a regular simplex with the same size (same area for triangles and same volume for tetrahedra). The *mean ratio* shape measure η can be written as:

$$\eta = \frac{\left(\prod_{i=1}^{d} \lambda_i\right)^{1/d}}{\frac{1}{d}\sum_{i=1}^{d} \lambda_i} = \begin{cases} \dfrac{2\left(\lambda_1\lambda_2\right)^{1/2}}{\lambda_1+\lambda_2} = \dfrac{4\sqrt{3}S_K}{\sum\limits_{1\le i<j\le 3} L_{ij}^2} & \text{in 2 D,} \\ \dfrac{3\left(\lambda_1\lambda_2\lambda_3\right)^{1/3}}{\lambda_1+\lambda_2+\lambda_3} = \dfrac{12\sqrt[3]{9V_K^2}}{\sum\limits_{1\le i<j\le 4} L_{ij}^2} & \text{in 3 D.} \end{cases} \tag{2.1}$$

where the λ_i, $i = 1,\ldots,d$ are the eigenvalues of the matrix $M^T M$ in dimension d. Notations are detailed in Appendices A, B and C.

Formaggia and Perotto [12] introduce another linear application N and define the *stretching factor* of simplex K as the ratio of the largest to the smallest eigenvalue of N^{-1}. This stretching factor is also the square root of the *condition number* of matrix $\mathcal{M}_K$. The condition number κ of a simplex is then defined as the inverse of the condition number of matrix $\mathcal{M}_K$. κ is a shape measure which is expressed as

$$\kappa = \frac{\min\limits_i \lambda_i}{\max\limits_i \lambda_i} = \frac{\lambda_1}{\lambda_d}, \tag{2.2}$$

if the eigenvalues are sorted in increasing order.

Knupp et Freitag [21, 13] have introduced another shape measure, also called the *condition number*, based on a composition of linear applications that first transforms a simplex K into a right angled reference element (such as used in finite element analysis) and then transforms this right angled element into an equilateral simplex of unit edge length. The transformation matrix is denoted $S = AW^{-1}$ and is the inverse of the transformations described above, going from reference space to real space. The

condition number of simplex K is defined as the product of Frobenius' norm of the matrix and of its inverse. Frobenius' matrix norm is defined as

$$\|S\| = \left(\mathrm{tr}(S^T S)\right)^{1/2}.$$

The matrix $S^T S$ involved in the norm of S is the inverse of the metric tensor, ie $\mathcal{M}_K^{-1}$. It is proven in [13] that the inverse of the condition number, scaled by the space dimension,

$$\widetilde{\kappa} = \frac{d}{\|S\| \, \|S^{-1}\|} \tag{2.3}$$

is a shape measure according to Definition 2.1.

2.2 The Radius Ratio

The radius ratio ρ of a simplex K is defined to be $\rho = d\rho_K / r_K$ where ρ_K and r_K are the inradius and circumradius of K, respectively, and d is the dimension of the space. Using the Eqs. (B.1) and (C.1) of ρ_K and r_K in two and three dimensions, the radius ratio is written as

$$\rho = d\frac{\rho_K}{r_K} = \begin{cases} \dfrac{8S_K^2}{p_K L_{12} L_{13} L_{23}} & \text{in 2 D,} \\ \dfrac{216V^2}{\sqrt{(a+b+c)(a+b-c)(a+c-b)(b+c-a)}\sum_{i=1}^{4} S_i} & \text{in 3 D,} \end{cases} \tag{2.4}$$

where a, b and c are the products of opposite edge lengths of a tetrahedron K.

2.3 The Minimum of the Solid Angles

A triangulation is a Delaunay triangulation if the circumsphere of each element does not contain any other vertex of the triangulation. In two dimensions, the Delaunay property is equivalent to maximize the minimum of the angles of the triangles of the triangulation, called the max–min angle criterion. It was therefore natural to use the minimum of the angles of a triangle as a shape measure. This shape measure in two dimensions will yield a triangulation satisfying the Delaunay criterion.

In three dimensions, Delaunay triangulations do not generally satisfy the max–min angle criterion [31]. Remember that a triangulation containing badly shaped tetrahedra like the sliver in Table 2 can pass the Delaunay criterion. A simplex shape measure based on the minimum of the solid angles θ_{min} will result in triangulations of better quality than Delaunay triangulations having some sliver elements.

The solid angle θ_i at the vertex P_i of a tetrahedron K is defined to be the surface area formed by projecting each point of the face not containing P_i to the unit sphere centered at P_i.

The *minimum of the solid angles* simplex shape measure is defined as

$$\theta_{min} = \alpha^{-1} \min_{1 \le i \le d+1} \theta_i, \tag{2.5}$$

where θ_i is given by Eq. (B.3) in two dimensions; θ_i is given by Eq. (C.3) in three dimensions; $\alpha = \pi/3 \simeq 1.047$ is the value of the three solid angles of the regular triangle and $\alpha = 6\arcsin\left(\sqrt{3}/3\right) - \pi \simeq 0.551$ is the value of the four solid angles of the regular tetrahedron.

This measure is sensitive either to narrow and to large solid angles. In two dimensions, the maximum of the solid angles in a triangle is π, the sum of the solid angles of a triangle is π and a large solid angle near π implies that the triangle has also small solid angles. In three dimensions, the area of a unit sphere is 4π, the maximum of the solid angles for a positive tetrahedron is 2π in the case of a flat tetrahedron where a vertex sees half of the space. The solid angle at the corner of a right-angled tetrahedron is $\pi/2$. It is shown in Gaddum [16] that $0 \leq \sum_{i=1}^{4} \theta_i \leq 2\pi$. Therefore, a large solid angle near 2π for K implies that K has small solid angles.

The evaluation of the expression $\theta_i = 2\arcsin(\cdot)$ is always in the interval $[0,\ \pi]$. It means that the Eq. (C.3) can be only used to measure solid angles less or equal to π. So the maximum solid angle of a tetrahedron can wrongly be evaluated using Eq. (C.3). However, the minimum of the solid angles is not affected. Let $\theta_1 \leq \theta_2 \leq \theta_3 \leq \theta_4$. Since $0 \leq \sum_{i=1}^{4} \theta_i \leq 2\pi$, only θ_4 may be larger than π. The computation with Eq. (C.3) of θ_4 returns $\widetilde{\theta}_4 = 2\pi - \theta_4$. Substituting θ_4 by $2\pi - \widetilde{\theta}_4$ in $\sum_{i=1}^{4} \theta_i \leq 2\pi$ gives $\theta_1 + \theta_2 + \theta_3 \leq \widetilde{\theta}_4$. So, θ_1 which is lower or equal to θ_2 and θ_3 is also lower or equal to $\widetilde{\theta}_4$. The conclusion is that even if the Eq. (C.3) does not return the correct value for a solid angle greater than π, it does not affect the tetrahedron shape measure (2.5) based on the minimum of the solid angles. An equivalent deduction can be done in two dimensions.

Instead of computing the $\arcsin(\cdot)$ in Eq. (B.3) and (C.3), from a computational point of view, a cheaper simplex shape measure is used in [23]

$$\sigma_{min} = \beta^{-1} \min_{1 \leq i \leq d+1} \sigma_i, \tag{2.6}$$

where $\sigma_i = \sin(\theta_i)$ in two dimensions; $\sigma_i = \sin(\theta_i/2)$ in three dimensions; $\beta = \sin(\alpha) = \sqrt{3}/2 \simeq 0.866$ is the value of σ_i for the three solid angles of the regular triangle; and $\beta = \sin(\alpha/2) = \sqrt{6}/9 \simeq 0.272$ is the value of σ_i for the four solid angles of the regular tetrahedron.

2.4 The Interpolation Error Coefficient

In finite element analysis there are theorems [32] that bound the interpolation error of a function over a finite element with a coefficient multiplied by the semi-norm of the function. This coefficient is something like D_K/ρ_K where D_K is the diameter of the element K and ρ_K is the roundness of the element K. The diameter of an element is the length of the greatest straight line inside the element. For a simplex, it is the longest edge. The roundness of an element is the diameter of the greatest sphere (circle in two dimensions) included in the element. For a simplex, it is twice the inradius. The diameter and the roundness are defined for simplicial and non simplicial elements but it is not necessarily easy to compute the diameter of the greatest sphere included in an hexahedron or a prism.

When a simplex K is degenerate or close to, the diameter D_K is large and the roundness ρ_K is small, then the coefficient D_K/ρ_K is a large value that indicates a large

bound for the interpolation error. So, this coefficient can be used to measure the shape of a simplex K. (Note that a large bound for the interpolation error does not necessarily mean that the interpolation error is effectively large).

To be compatible with Definition 2.1 of the simplex shape measures and to avoid division by zero for degenerate simplices, the *interpolation error coefficient* simplex shape measure is defined using the inverse of the coefficient of the interpolation error. This simplex shape measure is named γ, according to publications from the GAMMA project (Génération Automatique de Maillages et Méthodes d'Adaptation) at the Institut National de Recherche en Informatique et en Automatique (INRIA), France [18, 6, 14, 17, 18].

$$\gamma = \begin{cases} 2\sqrt{3}\,\dfrac{\rho_K}{h_{max}} & \text{in 2 D,} \\ 2\sqrt{6}\,\dfrac{\rho_K}{h_{max}} & \text{in 3 D.} \end{cases} \tag{2.7}$$

2.5 The Minimum of the Dihedral Angles

The minimum of the dihedral angles is a tetrahedron shape measure that has no two-dimensional equivalent. The dihedral angle at an edge of a tetrahedron is the angle between the intersection of the two faces sharing this edge and a plane perpendicular to the edge. For a positive tetrahedron, the dihedral angle is bounded by zero and π. It is equal to π minus the angle between the normals of the faces.

$$\varphi_{min} = \alpha^{-1} \min_{1 \le i < j \le 4} \varphi_{ij} = \alpha^{-1} \min_{1 \le i < j \le 4} \left(\pi - \arccos\left(n_{ij1} \cdot n_{ij2}\right)\right), \tag{2.8}$$

where n_{ij1} and n_{ij2} are the two normals to the triangular faces adjacent to the edge ij and $\alpha = \pi - \arccos(-1/3) = 1.231$ is the value of the six dihedral angles of the regular tetrahedron. The minimum of the dihedral angles is sometimes used as a tetrahedron shape measure.

According to Definition 2.1 of a tetrahedron shape measure, the minimum of the dihedral angles φ_{min} is an *invalid* tetrahedron shape measure. The underlying problem with tetrahedron shape measures based on the dihedral angles is that they fail to detect some degenerate tetrahedra. Referring to Table 3, the smallest dihedral angle of the needle can be as large as $\pi/3$ and the largest dihedral angle as small as $\pi/2$. The spindle and the crystal can have a minimum dihedral angle close to $\pi/3$, with a maximum dihedral angle close to π.

2.6 The Edge Ratio

The edge ratio r of a simplex K is defined to be the ratio between the length of the shortest edge over the longest, i.e.,

$$r = h_{min}/h_{max}, \tag{2.9}$$

where h_{min} and h_{max} are defined by Eqs. (B.2) and (C.2).

According to Definition 2.1 of a simplex shape measure, the edge ratio r is an *invalid* simplex shape measure because it does not vanish for all degenerate simplices. In two dimensions, it can be as large as $1/2$ for the cap. In three dimensions, it can be as large as $\sqrt{2}/2$ for the sliver, $1/2$ for the fin, $\sqrt{3}/3$ for the cap and $1/3$ for the crystal.

2.7 The Delaunay Criterion as a Shape Measure

The Delaunay criterion can be used to connect vertices to built a simplicial triangulation. As seen above in Sec. 2.3, using this criterion in two dimensions is strictly equivalent to reconnecting vertices regarding to the minimum of the angles shape measure.

Remarks must be made about three-dimensional triangulations satisfying the Delaunay criterion. It is important to note that valid simplex shape measures will detect slivers. Since triangulations that satisfy the Delaunay criterion can nevertheless contain slivers, satisfying the Delaunay criterion does not constitute a guarantee that a triangulation does not contain degenerate tetrahedra, and in that respect, the Delaunay criterion by itself does not act as a valid shape measure.

2.8 Extension to Riemannian Spaces

All simplex shape measures have been presented here in the frame of the usual Euclidean space. In order to get anisotropic meshes, these measures are now extended to the context of a Riemannian space.

A metric tensor is a tensor representing the deformation of space equivalent to the size specification map. Appendix A summarizes notions about metrics used in this paper, and exhaustive discussions on the topic of metrics can be found in [11, 20, 19, 5, 7, 8, 4] among many others; the first references being Vallet [33, 34], and the most complete being George and Borouchaki [18, 15].

A Riemannian metric is defined by a metric tensor field (see Appendix A) whose variations must be continuous. For meshing purpose, we assume smooth variations of the metric tensor. With this assumption metric variations can be locally neglected resulting in a local uniform, eventually stretched, metric tensor.

Practically, the metric tensor $\mathcal{M}(P)$ is averaged on a simplex into $\overline{\mathcal{M}}$ using a quadrature formula for each of its coefficients. This yields an Euclidean space that is not necessarily Cartesian (the tensor is not necessarily the identity matrix). Edge lengths are then computed using Eq. (A.2) with a uniform metric tensor. This simplifies in

$$L_{ij}^{\mathcal{M}} = \sqrt{(P_j - P_i)^T \, \overline{\mathcal{M}} \, (P_j - P_i)}. \tag{2.10}$$

Simplex measures are computed the same way using Eq. (A.5) where $\mathcal{M}$ is approximated by $\overline{\mathcal{M}}$. Hence, expressions for area of a triangle and volume of a tetrahedron in the local metric simplify into

$$S_K^{\mathcal{M}} = S_K \sqrt{\det(\overline{\mathcal{M}})}, \tag{2.11}$$

and

$$V_K^{\mathcal{M}} = V_K \sqrt{\det(\overline{\mathcal{M}})}. \tag{2.12}$$

The other simplex characteristics ρ_K, r_K, h_{min}, h_{max} and θ_i can now be approximated in the local metric using $L_{ij}^{\mathcal{M}}$, $S_K^{\mathcal{M}}$ and $V_K^{\mathcal{M}}$ in the approximate formulæ given in Appendices B and C. Finally, simplex shape measures can be approximated in the Riemannian metric.

3 Comparison of Triangle Shape Measures

A method is presented that helps to visualize the behavior of the various shape measures. This method is used to compare and choose a shape measure for its simplicity and its regularity. The effect of the metric tensor stemming from the anisotropic mesh is also illustrated.

3.1 Visualization of Triangle Shape Measures

Vallet [34] introduced a way to visualize the shape measure of a triangle. Consider three vertices *A*, *B* and *C* lying on the *xy*-plane (Fig. 1 left). The vertex *A* is at coordinate $(0,\ 1/2)$, the vertex *B* is at coordinate $(0,\ -1/2)$ and the vertex *C*, at coordinate $(x,\ y)$, is free to move in the half-plane $x \geq 0$.

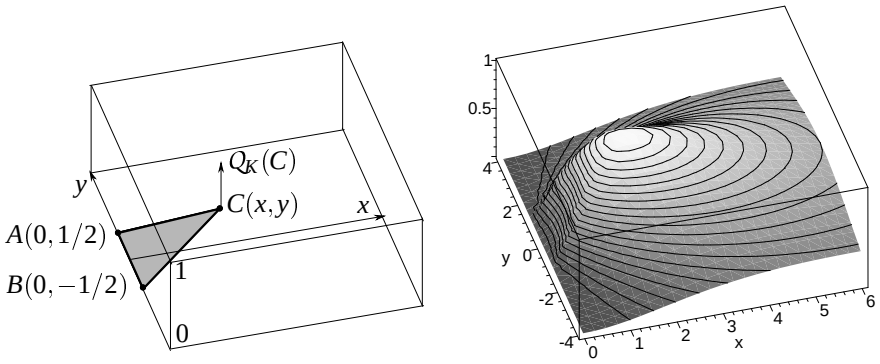


Figure 1: Definition of the location of the three vertices *A*, *B* and *C* of a triangle (left) to build the three-dimensional plot of the triangle shape measure (right).

At each location of vertex *C*, a given shape measure of the triangle *ABC* is evaluated which gives the three-dimensional plot of the shape measure shown in Fig. 1 on right. Using this method, Figs. 2 and 3 show the contour plot of some triangle shape measure introduced in Section 2. By examining these contour plots, it is easy to assess whether a triangle shape measure is valid according to the Definition 2.1: valid shape measures reach a maximum for the equilateral triangle at coordinate $(\sqrt{3}/2,\ 0)$; a minimum has to be zero and must be reached on the *y*-axis where the three vertices of

the triangle are collinear. Between these two extrema, the triangle shape measure must be a continuous function with values between 0 and 1 exclusively. These plots show that the edge ratio is not a valid triangle shape measure (see Fig. 3 on right) because it does not vanish on the *y*-axis where the three vertices are collinear and the triangle is degenerate.

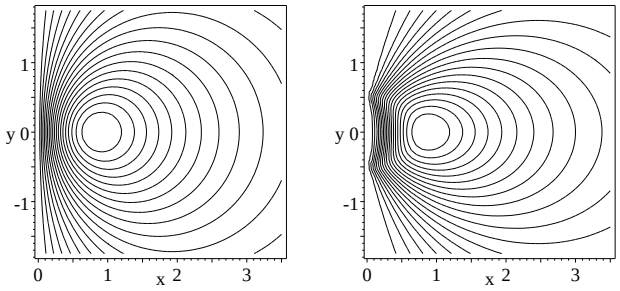


Figure 2: Contour plot of the mean ratio shape measure η (Eq. (2.1)) left, and of the radius ratio shape measure ρ (Eq. (2.4)) right.

3.2 Choice of a Measure for Mesh Optimization

Analysis of Fig. 2 and 3 helps to determine which triangle shape measure is the most convenient to use for mesh optimization. Clearly, an invalid simplex shape measure is a bad choice because it will not detect degenerate simplices and may drive the mesh optimizer in a wrong direction. Note however that in the neighborhood of the regular triangle, the contour lines of all shape measures have a correct behavior. Even the triangle edge ratio measure *r*, plotted in Fig. 3 on right will yield correct meshes as long as the mesh optimizer stays far away from degenerate triangles.

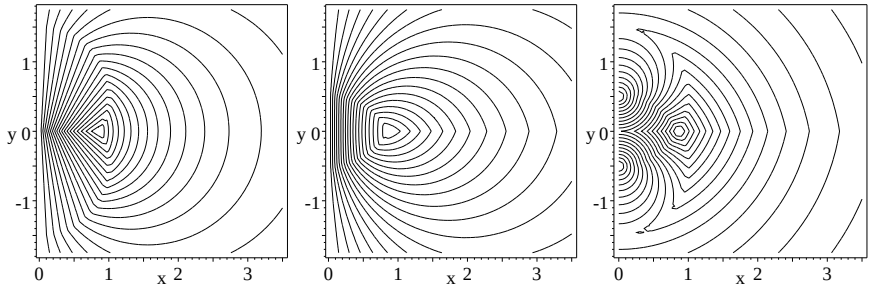


Figure 3: Contour plot of the minimum of solid angles shape measure θ_{min} (Eq. (2.5)) on left; of the interpolation error coefficient shape measure γ (Eq. (2.7)) in center and of the edge ratio measure *r* (Eq. (2.9)) on right.

All valid triangle shape measures plotted in Figs. 2 and 3 are continuous. Among them, ρ and η are differentiable, but θ_{min} and γ are not differentiable everywhere because these measures use the minimum or maximum operator, which are not differentiable. Non differentiable simplex shape measures can be used for mesh optimization but convergence may be less efficient.

Another well known feature of optimization algorithms is that the convergence is easier and faster if the function to maximize is circular around the optimal point. Conversely, the convergence will be slower if the function to maximize has sharp ridges (or narrow valleys for minimization problems). Finally, consider that the simplex shape measure ρ evaluated with Eq. (2.4) is an undefined fraction for some degenerate simplices. Moreover, as proved in [1], contour plots drawing the mean ratio measure η are circles. The conclusion from these remarks is that the mean ratio η is the most convenient simplex shape measure for two dimensional mesh optimization. It is also the less expensive to evaluate numerically.

So, from a practical point of view, the mean ratio η is the most convenient triangle and tetrahedron shape measure to use in a mesh optimizer.

3.3 Visualization of Anisotropic Shape Measure

The effect of the metric tensor on the contour plot of the shape measure can also be illustrated. Indeed, as presented in Sections 1.2 and 2.8, the metric tensor takes into account the anisotropy required for the mesh. Thus, the metric tensor stores the information necessary to specify the stretching and the orientation of the simplex to build. If the metric tensor is constant, it induces an Euclidean space that modifies the optimal position of the third vertex.

Figure 4 presents the contour plot of the two dimensional shape measure η for three different metric tensors. The η shape measure is the ratio of the area of the triangle over the sum of square of the length of the edges of the triangle (Eq. (2.1)). In order to take into account the different metric tensors, the area of the triangle in the metric space is computed using Eq. (A.5) and the length of the edges in the metric space are computed using Eq. (A.2). For a constant metric tensor, Eq. (A.2) can be simplified to Eq. (2.10) and the area of the triangle in the metric space can be written as a function of the length of its edges by using Heron's formula (B.4).

In Fig. 4, the metric tensor $\mathcal{M}$, is a positive definite symmetric tensor that is equal to $m_{11} = 0.2$, $m_{12} = 0$ and $m_{22} = 1$ for the figure on the left, $m_{11} = 20$, $m_{12} = 0$ and $m_{22} = 1$ for the figure in the middle and $m_{11} = 0.9$, $m_{12} = 0.4$ and $m_{22} = 1$ for the figure on the left. In comparison, Fig.2 on the right is nothing more than a representation of the η shape measure for which the metric tensor is the identity matrix. In these four graphs, degeneracies appear when the vertices are collinear and is independent of the metric tensor. However, the optimal position of the third vertex of the triangle depends on the value of the metric tensor used. Moreover, the contour lines of the η shape measure, that are circles for the identity matrix, now become ellipses in the general case.

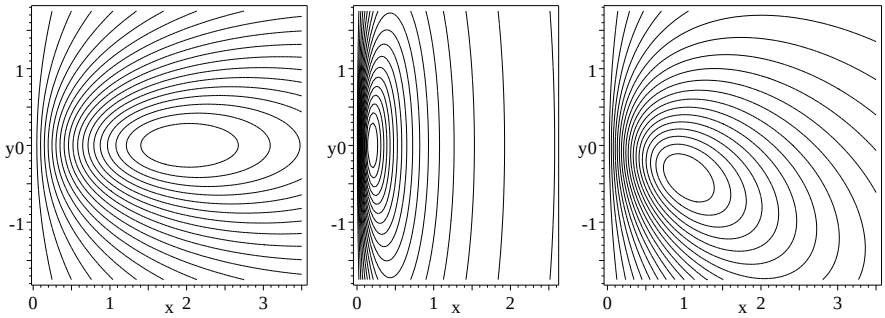


Figure 4: Contour lines of the η shape measure (Eq. (2.1)) for three constant metric tensors.

4 Implications for Mesh Optimization

For mesh optimization, all shape measures are more or less interchangeable, despite their differences. To illustrate the similarity of various shape measures, Fig. 5 shows a superposition of contour plots drawn previously.

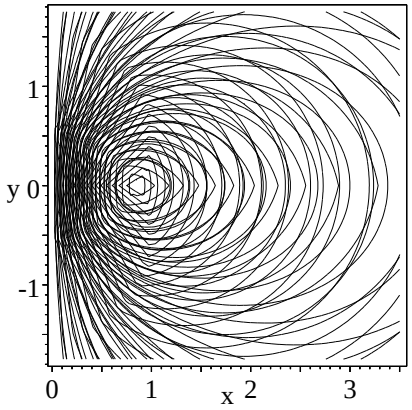


Figure 5: Superposition of the contour plot of the triangle shape measure η (Eq. (2.1)), ρ (Eq. (2.4)), θ_{min} (Eq. (2.5)) and γ (Eq. (2.7)).

This similarity can be formalized by the notion of equivalence between shape measures.

4.1 Equivalence Relation

One of the deepest analysis of simplex shape measures available is from Liu and Joe [23]. They define the notion of simplex shape measure equivalence in the following way:

Definition 4.1. Let μ and ν be two different simplex shape measures scaled in the interval $[0,\ 1]$. μ and ν are *equivalent* if there exist positive constants c_0, c_1, e_0 and e_1 such that

$$c_0\nu^{e_0} \leq \mu \leq c_1\nu^{e_1}. \tag{4.1}$$

If e_0 (e_1) is the minimum (maximum) possible exponent, then the lower (upper) bound is said *optimal*. If c_0 (c_1) is the maximum (minimum) possible constant, then the lower (upper) bound is said *tight*.

Note that this relation is an equivalence relation for shape measures since it is reflexive, symmetric, and transitive. The symmetry appears as follows: if μ is equivalent to ν with $c_0\nu^{e_0} \leq \mu \leq c_1\nu^{e_1}$, then ν is equivalent to μ with $c_2\mu^{e_2} \leq \nu \leq c_3\mu^{e_3}$, where $c_2 = c_1^{-1/e_1}$, $e_2 = 1/e_1$, $c_3 = c_0^{-1/e_0}$ and $e_3 = 1/e_0$. The transitivity appears as follows: if μ is equivalent to ν with $c_0\nu^{e_0} \leq \mu \leq c_1\nu^{e_1}$ and if ν is equivalent to ν with $c_2\nu^{e_2} \leq \nu \leq c_3\nu^{e_3}$ then μ is equivalent to ν with $c_4\nu^{e_4} \leq \mu \leq c_5\nu^{e_5}$, where $c_4 = c_0c_2^{e_0}$, $e_4 = e_0e_2$, $c_5 = c_1c_3^{e_1}$ and $e_5 = e_1e_3$.

4.2 Equivalence of Simplex Shape Measures

Liu and Joe [23] proved the equivalence of the tetrahedron shape measures η, ρ and σ_{min} (Eqs. (2.1), (2.4) and (2.6) respectively). More precisely, they demonstrated the following inequalities with strict upper bounds:

$$\begin{aligned}
\eta^3 \leq \rho \leq \eta^{3/4}, &\qquad \rho^{4/3} \leq \eta \leq \rho^{1/3},\\
0.2296\eta^{3/2} \leq \sigma_{min} \leq 1.1398\eta^{3/4}, &\qquad 0.8399\sigma_{min}^{4/3} \leq \eta \leq 2.6667\sigma_{min}^{2/3},\\
0.2651\rho^2 \leq \sigma_{min} \leq \rho^{1/2}, &\qquad \sigma_{min}^2 \leq \rho \leq 1.9420\sigma_{min}^{1/2}.
\end{aligned}$$

Furthermore, it can be shown that the η, κ, $\widetilde{\kappa}$ and γ shape measures (Eqs. (2.1), (2.2), (2.3) et (2.7)) belong to the same equivalence class, at least in two dimensions for γ. Indeed, the following relationship can be verified (the bounds are not necessarily optimal or tight).

$$\begin{aligned}
\frac{2}{3}\gamma^2 \leq \rho \leq \frac{2}{\sqrt{3}}\gamma &\qquad \text{in 2 D},\\
\kappa^{1/2} \leq \widetilde{\kappa} \leq d\kappa^{1/2} &\qquad \text{in } d \text{ D},\\
\kappa \leq \eta \leq d\kappa^{1/d} &\qquad \text{in } d \text{ D},\\
\widetilde{\kappa} \equiv \eta &\qquad \text{in 2 D},\\
\sqrt{2/3}\eta^{3/2} \leq \widetilde{\kappa} \leq 3\eta^{1/2} &\qquad \text{in 3 D}.
\end{aligned}$$

These equivalence imply that if one of these shape measures approaches zero, which indicates a poorly-shaped simplex, then so do the others. But the rate at which they approach zero or unity may differ as do μ and $\nu = \mu^2$ for example.

4.3 Equivalence Classes for Shape Measures

The equivalence relation can serve to partition the possible shape measures into equivalence classes. A shape measure would then be equivalent to all shape measures of it's class. In practice, it seems that all the shape measure presented in Sec. 2 belong to the same equivalence class. Indeed, it is the case for η, κ, $\widetilde{\kappa}$, γ, ρ et σ_{min}. It is tempting to conclude that *the equivalence class of Definition 4.1 includes all the shape measures that satisfy Definition 2.1*. But such is not the case. A counter example given by A. Liu serves to prove this point.

Let μ be a shape measure that satisfy Definition 2.1, then $\nu = 2^{(\mu-1)/\mu}$ is also a shape measure according to Definition 2.1. It can not be proven that μ et ν are equivalent according to Definition 4.1 since there does not exist any constants c_0 and e_0 such that $c_0\mu^{e_o} \leq \nu$ when μ tends toward zero. This is due to the fact that the exponential asymptotic behavior of ν tends towards zero faster than any possible polynomial asymptotic behavior.

Although the equivalence relation is weak, it is still possible to compare meshes obtained by optimizing different shape measures. Indeed, any shape measures that satisfy Definition 2.1 can be used by a mesh optimizer. Remember that, by definition, they will all detect all simplex degeneracies. They will all be sensitive to badly shaped simplices. Moreover, the more a mesh is optimized with a given simplex shape measure, the closer to the optimal mesh it is for any other simplex shape measure. In the limit, if it were possible to mesh a domain with only equilateral simplices, as it is in two dimensions, all mesh optimizers should converge to that mesh, whatever shape measure used in the mesh optimizer. These assertions are justified in the next section.

4.4 Practical Comparison of Shape Measures

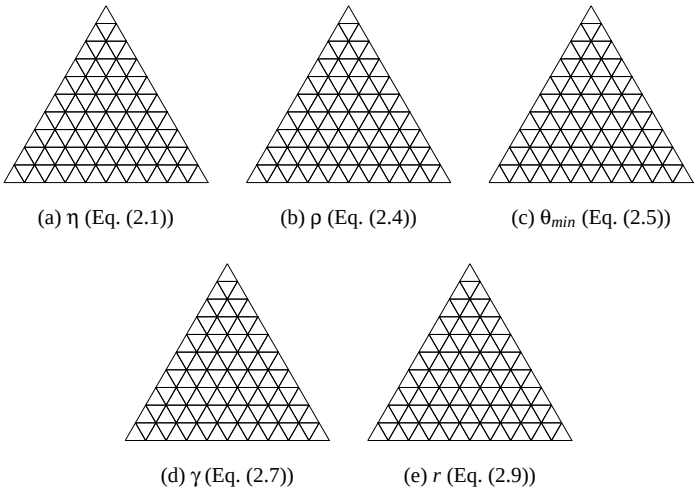
Three two-dimensional test-cases are presented to practically compare various shape measures. Due to the problem of visualizing tetrahedral meshes, no three-dimensional example is given. Results can be extended to the third dimension except that no perfect mesh exists in three dimensions. It is impossible to fill space with regular tetrahedra. So, the optimal mesh is always unknown and slightly depends on the tetrahedron shape measure used.

The test-cases are optimized using the mesh optimization package **OO**$\mathcal{RT}$ (**O**bject-**O**riented $\mathcal{R}$emeshing $\mathcal{T}$oolkit)[1]. It can be described as a mesh optimizer that acts on every characteristics of a mesh, namely the number of vertices, vertex coordinates and vertex connectivity. It is based on an algorithm performing successive local modifications such as mesh refinement, edge collapsing, vertex relocation or vertex reconnection. Stopping criteria are tight enough to perform a real mesh optimization, not only mesh enhancement.

In two dimensions, for certain domains, a perfect mesh can be obtained in which all the triangles are optimal. For instance, the domain consisting of an equilateral triangle meshed using an uniform target size of $1/10$ of the domain edge length can be meshed using only equilateral triangles. In that case, the initial mesh has no influence on the result since it is unique. Using the same algorithm with various shape measures,

[1] See `http://www.cerca.umontreal.ca/oort`.

the perfect mesh must be obtained from any initial mesh, and is obtained with the mesh optimizer.



(a) η (Eq. (2.1)) (b) ρ (Eq. (2.4)) (c) θ_{min} (Eq. (2.5))

(d) γ (Eq. (2.7)) (e) r (Eq. (2.9))

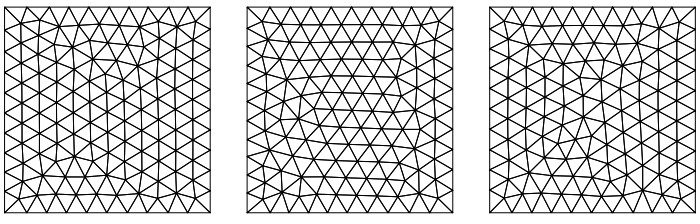
Figure 6: Triangular meshes of an equilateral triangular domain with a mesh target edge length of $1/10$ of the domain size and with five different shape measures in the edge swapping process.

This example shows that the triangle shape measure has no effect on the final mesh, if a perfect mesh can be obtained. If the heuristic used by the mesh optimizer is able to get it, it will obtain it, whatever the shape measure used. In this example, three shape measures have led to exactly the same choices during the edge swapping process: the same 124 edges have been swapped, on about 2000 checked edges. The values of the shape measures differed but comparing the same configurations drove to the same decisions.

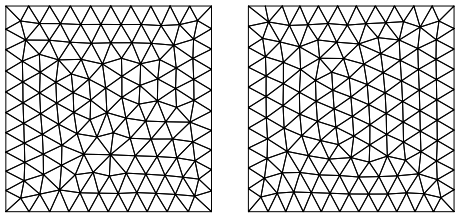
The previous example is really academic since perfect meshes are impossible on almost every domain. The next example is a square. Optimal meshes are not perfect, so they vary with the shape measure. But differences are small (see Fig. 7). The choice of a shape measure has more effect when the optimal mesh is far from a perfect mesh.

The last test-case is the same as the first one, but the optimization algorithm differs. Vertex relocation has been excluded from the possible mesh modifications. The resulting mesh optimizer is less powerful, so it is not surprising that the final meshes are not perfect anymore (Fig. 8).

Although the five meshes obtained with various shape measures differ significantly, it is not right to conclude from these differences that one shape measure is better than the other. The only valid conclusion is that the mesh optimizer is not powerful enough to reach the optimal mesh. The problem is in the optimization algorithm, not in the simplex shape measure used for optimization.

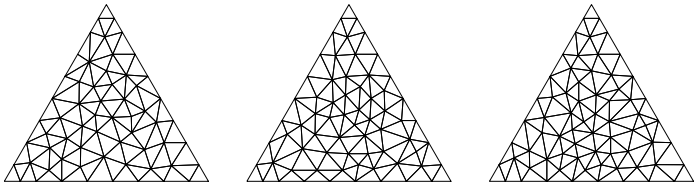


(a) η (Eq. (2.1)) (b) ρ (Eq. (2.4)) (c) θ_{min} (Eq. (2.5))

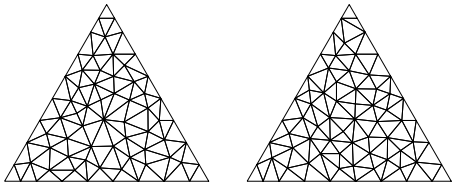


(d) γ (Eq. (2.7)) (e) r (Eq. (2.9))

Figure 7: Triangular meshes of a square domain with a mesh target edge length of $1/10$ of the domain size, using five different shape measures in the edge swapping process.



(a) η (Eq. (2.1)) (b) ρ (Eq. (2.4)) (c) θ_{min} (Eq. (2.5))



(d) γ (Eq. (2.7)) (e) r (Eq. (2.9))

Figure 8: Triangular meshes of an equilateral triangular domain with the same target and the same triangle shape measures as on Fig. 6. Only the optimization algorithm differs.

Conclusions

A taxonomy of simplex degeneracies has been proposed and a systematic way of classifying them based on the final configuration of the degenerate simplex was presented.

A definition of simplex shape measure has been put forth that takes into account all possible degeneracies. Using this definition, commonly used shape measures were cast into a standard notation and analyzed to determine which were valid shape measures. A method was given to use them for anisotropic meshing by extending the shape measures to Riemannian space by expressing basic concepts such as length, area and volume in a metric.

Several shape measures were visually compared which has clearly expressed the regularity of the mean ratio η (Eq. (2.1)) and the radius ratio ρ (Eq. (2.4)), and the shortcomings of invalid shape measures were particularly evident. All the commonly used shape measures were then compared using the equivalence relation of Liu and Joe [23], and all were shown to be in the same equivalence class. Furthermore, it was shown on academic test-cases that the choice of a shape measure to drive mesh optimization is much less crucial that the heuristic of the mesh optimizer.

Although all of the tested shape measures yielded comparable results, the mean ratio η (Eq. (2.1)) is particularly appealing due to its simplicity and its regularity.

From the present work, it can be concluded that more attention should be given to optimization and generation algorithms rather than defining new shape measures. Indeed, *our intuition* about mesh generation and optimization is that it is a *non linear*, and even *discontinuous* problem. It is therefore impossible to generate directly the optimal mesh. The optimal mesh can only be reached with an iterative process that can remove vertices, change the coordinates of the vertices and change the connectivity between vertices. These are the operations that allow to handle all degrees of freedom of a mesh. All these operations must be combined together in an heuristic such that they work together in the same direction and that they converge towards the optimal mesh. Ideally, they should optimize the same objective function measuring how much a mesh fits with the target. In that context, the choice of a specific shape measure has far less impact on the final mesh than the careful combination of operations within the optimization strategy.

References

[1] R. E. Bank and R. K. Smith. Mesh smoothing using a posteriori error estimates. *SIAM J. Numer. Anal.*, 34(3):979–997, June 1997.

[2] M. Bern, P. Chew, D. Eppstein, and J. Ruppert. Dihedral bounds for mesh generation in high dimensions. In *6th ACM-SIAM Symp. Discrete Algorithms*, pages 189–196, San Francisco, CA, 1995.

[3] M. Bern and D. Eppstein. *Computing in Euclidean Geometry*, chapter Mesh Generation and Optimal Triangulation, pages 47–123. World Scientific, D.-Z. Du and F. K. Hwang Eds., 2nd edition, 1995.

[4] H. Borouchaki and P. J. Frey. Adaptive triangular-quadrilateral mesh generation. *Int. J. Numer. Meth. Engng*, 41:915–934, 1998.

[5] H. Borouchaki, P.-L. George, F. Hecht, P. Laug, and É. Saltel. Delaunay mesh generation governed by metric specification. Part I. Algorithms. *Finite Elem. in Anal. and Design*, 25:61–83, 1997.

[6] H. Borouchaki, F. Hecht, É. Saltel, and P.-L. George. Reasonably efficient Delaunay based mesh generator in 3 dimensions. In *Fourth International Meshing Roundtable*, pages 3–14, Albuquerque, New Mexico, October 1995. Sandia National Laboratories.

[7] G. C. Buscaglia and E. A. Dari. Anisotropic mesh optimization and its application in adaptivity. *Int. J. Numer. Meth. Engng*, 40:4119–4136, 1997.

[8] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulation. *Int. J. Numer. Meth. Fluids*, 25:475–491, 1997.

[9] T. Coupez. *Grandes transformations et remaillage automatique*. PhD thesis, École Nationale Supérieure des Mines de Paris, November 1991.

[10] J. Dompierre, P. Labbé, F. Guibault, and R. Camarero. Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization. In *Seventh International Meshing Roundtable*, pages 459–478, Dearborn, MI, October 1998. Sandia National Laboratories.

[11] J. Dompierre, M.-G. Vallet, M. Fortin, W. G. Habashi, D. Aït-Ali-Yahia, S. Boivin, Y. Bourgault, and A. Tam. Edge-based mesh adaptation for CFD. In *Conference on Numerical Methods for the Euler and Navier-Stokes Equations*, pages 265–299, Montréal, September 1995. CRM-CERCA.

[12] L. Formaggia and S. Perotto. Anisotropic error estimator for finite element methods. In *31st Computational Fluid Dynamics*, Lecture Series 2000–05. von Karman Institute for Fluid Dynamics, March 2000.

[13] L. Freitag and P. M. Knupp. Tetrahedral element shape optimization via the Jacobian determinant and condition number. In *Eight International Meshing Roundtable*, pages 247–258, South Lake Tahoe, CA, October 1999. Sandia National Laboratories.

[14] P. J. Frey, H. Borouchaki, and P.-L. George. Delaunay tetrahedralization using an advancing-front approach. In *Fifth International Meshing Roundtable*, Pittsburgh, PA, October 1996. Sandia National Laboratories.

[15] P. J. Frey and P.-L. George. *Mesh Generation. Application to Finite Elements*. Hermès, Paris, 2000.

[16] J. W. Gaddum. The sums of the dihedral and trihedral angles in a tetrahedron. *Amer. Math. Monthly*, 59:370–371, 1952.

[17] P.-L. George. Improvements on Delaunay-based three-dimensional automatic mesh generator. *Finite Elem. in Anal. and Design*, 25(3):297–317, 1997.

[18] P.-L. George and H. Borouchaki. *Delaunay Triangulation and Meshing. Applications to Finite Elements*. Hermès, Paris, 1998.

[19] W. G. Habashi, M. Fortin, J. Dompierre, M.-G. Vallet, D. Aït-Ali-Yahia, Y. Bourgault, M. P. Robichaud, A. Tam, and S. Boivin. Anisotropic mesh optimization for structured and unstructured meshes. In *28th Computational Fluid Dynamics Lecture Series*, Lecture Series 1997–02. von Karman Institute, von Karman Institute for Fluid Dynamics, March 1997.

[20] F. Hecht and B. Mohammadi. Mesh adaption by metric control for multi-scale phenomena and turbulence. In *AIAA 35th Aerospace Sciences Meeting & Exhibit*, number AIAA–97–0859, Reno, NV, January 1997.

[21] P. M. Knupp. Matrix norms and the condition number, A general framework to improve mesh quality via node-movement. In *Eight International Meshing Roundtable*, pages 13–22, South Lake Tahoe, CA, October 1999. Sandia National Laboratories.

[22] A. Liu and B. Joe. On the shape of tetrahedra from bisection. *Math. of Comp.*, 63(207):141–154, 1994.

[23] A. Liu and B. Joe. Relationship between tetrahedron shape measures. *Bit*, 34:268–287, 1994.

[24] A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on bisection. *SIAM J. Sci. Comp.*, 16:1269–1291, 1995.

[25] A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision. *Math. of Comp.*, 65(215):1183–1200, July 1996.

[26] S. H. Lo. Optimization of tetrahedral meshes based on element shape measures. *Computers & Structures*, 63(5):951–961, 1997.

[27] S. H. Lo. 3D mesh refinement in compliance with a specified node spacing function. *Computational Mechanics*, pages 11–19, 1998.

[28] D. J. Mavriplis. Adaptive mesh generation for viscous flows using Delaunay triangulation. *J. Comp. Phys.*, 90(2):271–291, 1990.

[29] V. N. Parthasarathy, C. M. Graichen, and A. F. Hathaway. A comparison of tetrahedron quality measures. *Finite Elem. in Anal. and Design*, 15:255–261, 1993.

[30] G. Polya. *Mathematics and Plausible Reasoning. Vol. 2: Patterns of Plausible Inference*. Princeton University Press, 1954.

[31] V. T. Rajan. Optimality of the Delaunay triangulation in R^d. *Disc. & Comp. Geometry*, 12:189–202, 1994.

[32] P.-A. Raviart and J.-M. Thomas. *Introduction à l'analyse numérique des équations aux dérivées partielles*. Masson, 2^{nd} edition, 1988.

[33] M.-G. Vallet. Génération de maillages anisotropes adaptés. Application à la capture de couches limites. Technical Report 1360, Institut National de Recherche en Informatique et en Automatique, France, December 1990.

[34] M.-G. Vallet. *Génération de maillages éléments finis anisotropes et adaptatifs*. PhD thesis, Université Pierre et Marie Curie, Paris VI, France, 1992.

A Notion of Riemannian Metric

The standard way of measuring distances in Euclidean space is a special case of a more general approach to measure that shall be briefly summarized here. The objective behind the introduction of this length measurement approach is to recast in a more general framework the various numerical constraints imposed by applications on numerical mesh generation and adaptation. Each type of application imposes domain specific constraints on mesh generation in terms of optimal element size distribution, element size distortion and stretching. Through the use of a metric, constraints can be expressed in domain independent terms, and in doing so, mesh quality can be assessed in absolute terms, regardless of specific application domains.

In order to abstract shape and size specifications, which can take numerous forms and names, a general tensorial approach is taken, whereby all desired mesh characteristics are expressed in terms of a continuous second order tensor. The Riemannian metric framework was chosen for mesh constraints representation since this formalism can account for both isotropic and anisotropic mesh control specifications, and includes a wide range of size and shape specification schemes in current use in the literature.

Let P be a point in space $\mathbf{R}^d$, and $\mathcal{M}(P)$ be a continuous second order symmetric tensor field define over $\mathbf{R}^d$. If $\mathcal{M}(P)$ is symmetric positive definite, it can be expressed as the product of three tensors:

$$\mathcal{M}(P) = R^T(P)D(P)R(P), \tag{A.1}$$

with $R(P)$ a rotation tensor, $R^T(P) = R^{-1}(P)$ the inverse rotation tensor, and $D(P)$ a diagonal tensor with all terms strictly positive. These diagonal terms are the eigenvalues of $\mathcal{M}(P)$. The metric tensor field $\mathcal{M}(P)$ associated with a method to measure distance then defines a *Riemannian metric* or simply a *metric* over space $\mathbf{R}^d$.

A.1 Length

With this definition, many simplex quantities can be rewritten in terms of the metric. First, the length of a parametric curve $\gamma(t) = \{(x(t),y(t),z(t)), t \in [0,1]\}$ can be expressed in metric space $\mathcal{M}$ by equation

$$L_{\mathcal{M}}(\gamma) = \int_0^1 \sqrt{(\gamma'(t))^T \mathcal{M}(\gamma(t))\,\gamma'(t)}\,dt. \tag{A.2}$$

In this expression, $\gamma(t)$ represents a point on the curve and $\gamma'(t)$ the tangent vector at that point.

Mavriplis [28] introduced metric change in terms of local space deformation. To illustrate this, lets denote

$$D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix},$$

the diagonal tensor made up of the eigenvalues of the metric tensor in three dimensions. The local linear transformation N

$$N = \begin{pmatrix} \sqrt{\lambda_1} & 0 & 0 \\ 0 & \sqrt{\lambda_2} & 0 \\ 0 & 0 & \sqrt{\lambda_3} \end{pmatrix} R, \tag{A.3}$$

can now be constructed from $R(P)$, the rotation tensor, and $D^{1/2}$. This local transformation relates to the metric $\mathcal{M}$ in the way that it is equivalent to measure the length of a vector X in the original space through the metric $\mathcal{M}$, or to measure the length of the transformed vector NX. Indeed,

$$X^T \mathcal{M} X = X^T N^T N X = (NX)^T NX = \|NX\|^2, \tag{A.4}$$

Lengths are thus easily measured with a metric. Shape measures, however, are often expressed in terms not only of lengths, but also of area, volume and angles, which must also be computed using a metric.

A.2 Measure

The measure of simplex K — area in two dimensions and volume in three dimensions — in the metric $\mathcal{M}$, is the integral of the square root of the metric's determinant over K. This measure will be denoted $V_K^{\mathcal{M}}$. Through a change of variable, using Eq. (A.4), we can write

$$V_K^{\mathcal{M}} = \int_{N(K)} 1\, dV = \int_K \det(N)\, dV$$

since $\det(N)$ is the Jacobian of the linear transformation. Moreover, from Eq. (A.3), we have $\det(\mathcal{M}) = (\det(N))^2$, so that

$$V_K^{\mathcal{M}} = \int_K \sqrt{\det(\mathcal{M})}\, dV. \tag{A.5}$$

This integral can be computed numerically using a quadrature rule. Approximate formulæ can also be derived, that are based on edge lengths, as are presented in the following appendices.

A.3 Angle

Another quantity often involved in shape measure is the angle between two vectors in the metric space. Let two vectors V and W that meet at point P, we have

$$\cos(\theta) = \frac{V^T \mathcal{M}(P) W}{\|NV\|\, \|NW\|} \tag{A.6}$$

where θ is the angle between V and W.

B Approximate Formulæ for the Triangle

When dealing with shape measures in metric space, computation of all quantities must be recast in this new framework. While some basic quantities can be precisely expressed and numerically computed using exact formalution, derived quantities tend to become very involved in terms of computational complexity and cost. For that reason, approximate formulæ are often very helpful to help analyze and compare shape measures in metric space. The following formulæ are valid in Euclidean and for uniform metric spaces. A uniform metric space is a deformed space where the deformation tensor is constant from point to point.

Since edge length is the simplest measure to compute over a simplex, care has been taken to formulate all these approximate formulæ in terms of edge lengths measured in Euclidean or metric space.

Let K stands for a non degenerate triangle with vertices P_1, P_2 and P_3; S_K denotes the area of K and $L_{ij} = \|P_j - P_i\|, 1 \leq i < j \leq 3$, denotes the length of the three edges $\overline{P_iP_j}$ of K.

The half-perimeter p_K, the inradius ρ_K and the circumradius r_K are given by

$$p_K = \frac{(L_{12} + L_{13} + L_{23})}{2}, \qquad \rho_K = \frac{S_K}{p_K}, \qquad r_K = \frac{L_{12}\,L_{13}\,L_{23}}{4S_K}. \tag{B.1}$$

In finite elements terms, the diameter h_{max} of an element is the maximum Euclidean distance between two points of an element, which is the longest edge for a simplex. The smallest edge of an element is denoted h_{min}.

$$h_{max} = \max(L_{12}, L_{13}, L_{23}), \qquad h_{min} = \min(L_{12}, L_{13}, L_{23}). \tag{B.2}$$

The angle θ_i at vertex P_i of the triangle K is expressed as a function of the edge lengths as

$$\theta_i = \arcsin\left(2S_K\left(\prod_{\substack{j,k\neq i \\ 1\leq j<k\leq 3}} L_{ij}L_{ik}\right)^{-1}\right). \tag{B.3}$$

Finally, the area S_K of the triangle can also be expressed as a function of its edge lengths by Heron's formula:

$$S_K^2 = p_K(p_K - L_{12})(p_K - L_{13})(p_K - L_{23}). \tag{B.4}$$

Note that if the three lengths used, L_{12}, L_{13} and L_{23}, do not form a triangle, i.e. they do not satisfy the triangular inequality, as it can happen in a non-Euclidean metric, then the square of the area has a negative value. Heron's formula has no sense in that case. One must then resort to Eq. (A.5) in order to compute the precise triangular area in metric space.

C Approximate Formulæ for the Tetrahedron

As for the triangle, the knowledge of the edge lengths of a tetrahedron is sufficient to uniquely define a tetrahedron in Euclidean or constant metric space. Significant

formulæ for the tetrahedron can thus be expressed in terms of its edge lengths, which will be valid, again, if sharp deformations in space are not encompassed in single mesh elements.

Let K stands for a non degenerate tetrahedron with vertices P_1, P_2, P_3 and P_4; $L_{ij} = \|P_j - P_i\|, 1 \leq i < j \leq 4$, denotes the length of the six edges $\overline{P_iP_j}$ of K; S_1 denotes the area of the triangular face $\triangle P_2P_3P_4$, S_2 the area of face $\triangle P_1P_3P_4$, S_3 the area of face $\triangle P_1P_2P_4$ and S_4 the area of face $\triangle P_1P_2P_3$; V_K denotes the volume of tetrahedron K.

The inradius ρ_K and the circumradius r_K are given by

$$\begin{aligned} \rho_K &= \frac{3V_K}{S_1+S_2+S_3+S_4}, \\ r_K &= \frac{\sqrt{(a+b+c)(a+b-c)(a+c-b)(b+c-a)}}{24V_K}, \end{aligned} \tag{C.1}$$

where $a = L_{12}L_{34}$, $b = L_{13}L_{24}$ and $c = L_{14}L_{23}$ are the products of opposite edge lengths of K. In a tetrahedron, two edges are opposed if they share no vertex.

The minimum and maximum edge lengths h_{min} and h_{max} are trivially given by

$$\begin{aligned} h_{min} &= \min(L_{12}, L_{13}, L_{14}, L_{23}, L_{24}, L_{34}), \\ h_{max} &= \max(L_{12}, L_{13}, L_{14}, L_{23}, L_{24}, L_{34}). \end{aligned} \tag{C.2}$$

The solid angle θ_i at the vertex P_i of the tetrahedron is defined to be the surface area formed by projecting each point of the face not containing P_i to the unit sphere centered at P_i. Liu and Joe [23] give a formula to compute solid angles of a tetrahedron in terms of edge lengths

$$\theta_i = 2\arcsin\left(12V_K\left(\prod_{\substack{j,k\neq i \\ 1\leq j<k\leq 4}}\left((L_{ij}+L_{ik})^2 - L_{jk}^2\right)\right)^{-1/2}\right). \tag{C.3}$$

Finally, the volume V_K of a tetrahedron can also be expressed as a function of its edge lengths. Let a, b, c, e, f and g, the length of the six edges of the tetrahedron. Edges a, b and c connect to the same vertex of the tetrahedron, e is the edge opposite of a, f to b and g to c. The edges e, f and g are the three sides of a face of the tetrahedron, opposite to the vertex to which a, b and c connect. For example, a, b, c, e, f and g can be L_{12}, L_{13}, L_{14}, L_{34}, L_{24} and L_{23} respectively. Then, the volume of the tetrahedron is given by [30]

$$\begin{aligned} 144V_K^2 &= 4a^2b^2c^2 + \left(b^2+c^2-e^2\right)\left(c^2+a^2-f^2\right)\left(a^2+b^2-g^2\right) \\ &- a^2\left(b^2+c^2-e^2\right) - b^2\left(c^2+a^2-f^2\right) - c^2\left(a^2+b^2-g^2\right). \end{aligned} \tag{C.4}$$

Contrary to Heron's formula (B.4) in two dimensions, this three-dimensional version of Heron's formula is not symmetric. The three edges a, b and c which start from the same vertex do not play the same role as the three edges e, f and g which share a face. Eq. (C.4) can be rewritten in a formula that exhibits more symmetry [30]. However, it is costlier to compute and more subject to roundoff error.

Note that if the six lengths do not form a tetrahedron, as can happen in a non-Euclidean metric, then the square of the volume has a negative value. Heron's three-dimensional formula has no sense in that case. One must then resort to Eq. (A.5) in order to compute the precise tetrahedral volume in metric space.

TRANSIENT ADAPTIVE DISCONTINUOUS GALERKIN METHOD WITH ANISOTROPIC MESHES

JEAN-FRANÇOIS REMACLE[¶]

Université Catholique de Louvain
Département de Génie Civil et Environnemental
Bâtiment Vinci, Place du Levant 1
1348 Louvain-la-Neuve, Belgium

ABSTRACT: In this paper, we present a method for performing anisotropic adaptive computations. The discontinuous Galerkin method that is used for solving transient flow problem is briefly introduced. We develop then an error indication technique that provides directional error information in order to build a non uniform metric field. We present two sample problems involving hundred of mesh refinements, both in 2D and 3D.

KEY WORDS: Anisotropic meshes, Discontinuous Galerkin, Transient Adaptation

1 The Discontinuous Galerkin Method

Transient flow problems involving wave propagation are of great interest in computational fluid dynamics. An accurate tracking of features like moving shocks or fluid interfaces in an Eulerian fashion implies multiple mesh adaptations in order to follow complex features of the flow. The discontinuous Galerkin method (DGM) is a good candidate for solving our problems of interest. The DGM can be regarded as an extension of finite volume methods to arbitrary orders of accuracy without the need to construct complex stencils for high-order reconstruction. We seek to determine $\mathbf{u}(\Omega,t) : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathsf{L}^2(\Omega)^m = V(\Omega)$ as the solution of a *system of conservation laws*

$$\partial_t \mathbf{u} + \mathbf{div}\vec{\mathbf{F}}(\mathbf{u}) = \mathbf{r}. \tag{1.1}$$

With the aim of constructing a Galerkin form of (1.1), let $(\cdot,\cdot)_\Omega$ and $\langle\cdot,\cdot\rangle_{\partial\Omega}$ denote the standard $\mathsf{L}^2(\Omega)$ and $\mathsf{L}^2(\partial\Omega)$ scalar products respectively. Multiply equation (1.1) by a test function $\mathbf{w} \in V(\Omega)$, integrate over Ω and use the divergence theorem to obtain the following variational formulation

$$(\partial_t \mathbf{u}, \mathbf{w})_\Omega - (\vec{\mathbf{F}}(\mathbf{u}), \mathbf{grad}\,\mathbf{w})_\Omega + \langle \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}, \mathbf{w}\rangle_{\partial\Omega} = (\mathbf{r}, \mathbf{w})_\Omega \ , \quad \forall \mathbf{w} \in V(\Omega). \tag{1.2}$$

The physical domain Ω is first discretized into a collection of $\mathcal{N}_e$ elements $\mathcal{T}_e = \bigcup_{e=1}^{\mathcal{N}_e} e$ called a mesh. The continuous function space $V(\Omega)$ containing the solution of (1.2) is then approximated on each element e of the mesh to define a finite-dimensional space $V_e(\mathcal{T}_e)$. With discontinuous finite elements, V_e is a "broken" function space that

[¶]remacle@gce.ucl.ac.be

consists in the direct sum of elementary approximations $\mathbf{u}_e$ (we use here a polynomial basis $\mathbb{P}^q(e)$ of order q):

$$V_e(\mathcal{T}_e) = \{\mathbf{u} \mid \mathbf{u} \in \mathsf{L}^2(\Omega)^m, \mathbf{u}_e \in \mathbb{P}^q(e)^m = V_e(e)\}. \tag{1.3}$$

Because all approximation are disconnected, we can solve the conservation laws on each element. Now, a discontinuous basis implies that the normal trace $\mathbf{F}_n = \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}$ is not defined on ∂e. If $\mathbf{u}_e$ and $\mathbf{u}_{e_k}$ are the restrictions of solution $\mathbf{u}$, respectively, to element e and element e_k, a *numerical flux* $\mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k})$ is usually used on each portion ∂_{e_k} of ∂e shared by element e and neighboring element e_k. We have then a DG formulation: $\forall \mathbf{w} \in V_e(e)$,

$$(\partial_t \mathbf{u}_e, \mathbf{w})_e - (\vec{\mathbf{F}}(\mathbf{u}_e), \mathbf{grad}\,\mathbf{w})_e + \sum_{k=1}^{n_e} \langle \mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k}), \mathbf{w} \rangle_{\partial e_k} = (\mathbf{r}, \mathbf{w})_e \tag{1.4}$$

where n_e is the number of faces of element e. Several operators are possible [6, 7]. Herein, a quasi-exact Riemann solver is used to compute the numerical fluxes and a slope limiter [1] is used to produce monotonic solutions when polynomial degrees $q > 0$ are used.

2 An Error Indicator

For $q = 1$, the following quantity is used as an error indicator:

$$|e|(\nabla\varepsilon)_e = \sum_{k=1}^{n_e} \int_{\partial e_k} \frac{(u_e - u_{e_k}^+)}{2} \vec{n}_k \, ds \tag{2.1}$$

where u is a scalar quantity depending on $\mathbf{u}$, $|e|$ is the volume of element e and $(\nabla\varepsilon)_e$ is a gradient dued to downwind jumps $(\mathbf{u}_e - \mathbf{u}_{e_k}^+)$ of the solution that show super-convergence [2]. We use the 3 principal directions $\vec{h}_k$, $k = 1,2,3$ of the Hessian $H_{i,j}(u) = \frac{\partial^2 u}{\partial x_i \partial x_j}$ to compute directional errors $(\nabla\varepsilon)_e \cdot \vec{h}_k$. In each principal direction $\vec{h}_k$, we compute a directional size field that takes into account directional errors.
The mesh adaptation based on the directional size field is then performed by means of some local mesh modifications [4]. In case of transient computations, the mesh-to-mesh interpolations are performed using some local projection schemes. Those are are differentiated with respect to the nature of each atomistic mesh modification operator: operators including only topological modifications, operators including only geometrical modifications, operator including both topological and geometrical modifications.

3 Examples

We present the results of two compressible inviscid flow problems involving the solution of the Euler equations [3] by a DGM. The three-dimensional Euler equations have the form (1.1) with $\mathbf{u} = \{\rho, \rho v_x, \rho v_y, \rho v_z, E\}^t$, $\vec{\mathbf{F}}(\mathbf{u}) = \{\rho\vec{v}, \rho v_x\vec{v} + P\vec{e}_x, \rho v_y\vec{v} + P\vec{e}_y, \rho v_z\vec{v} + P\vec{e}_z, (\rho E + P)\vec{v}\}^t$ and $\mathbf{r} = \mathbf{0}$. Here, ρ is the fluid density, $\vec{v}$ the velocity,

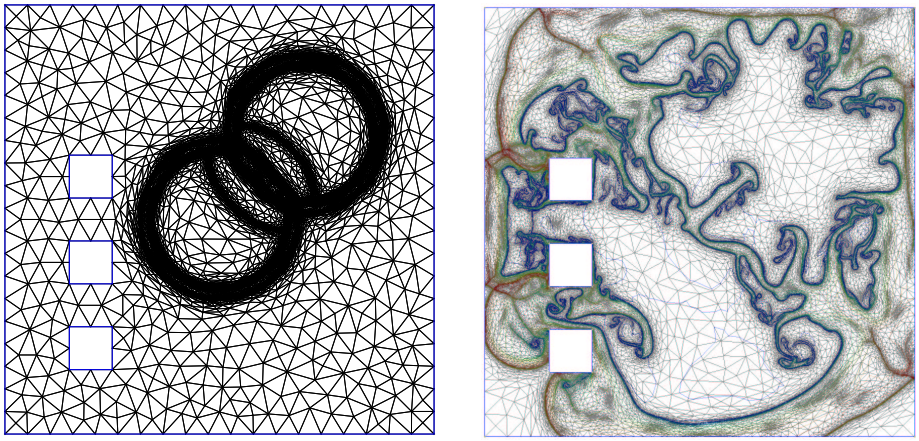


Figure 1: Mesh after 25 adaptations (left) and mesh+density contours after 227 adaptations (right) for the 2D problem.

E the internal energy, P the pressure and $\vec{e}_x$, $\vec{e}_y$ and $\vec{e}_z$ are the unit vectors in the x, y and z directions, respectively. An equation of state of the form $P = P(\rho, E)$ is also necessary to close the system. Here, we have chosen the perfect gas equation of state $P = (\gamma - 1)\,\rho \left[E - \frac{\|\vec{v}\|^2}{2}\right]$ with the gas constant $\gamma = 1.4$.
One of the most widely known models of blast waves is the famous Taylor-Sedov point source solution. The Sedov problem [5] involves the self-similar evolution of a cylindrical or spherical blast wave from a delta-function initial pressure perturbation in an otherwise homogeneous medium. In our example, we have made 2 Sedov explosions collide into a square domain that contains 3 rectangular obstacles (Figure 1). In practice, we initialize the code by depositing a quantity of energy $E = 1$ into a small region of radius dr. The pressure inside this volume, P_0, is given by $P_0 = \frac{3(\gamma-1)E}{3\pi dr^2}$. Enywhere else, the density is set to $\rho = 1$ and the velocity is null initially.
We see on Figure 1 two meshes after different numbers of adaptations. The collision of the two blasts creates a very high density region in the center of the system that rapidely becomes unstable. We observe on Figure 1 the power of anisotropic mesh refinement to capture accurately complex features resulting of the various instabilities of the flow. We have finally run a one blast Sedov problem in 3D. One picture of the results is shown on Figure 2.

References

[1] R. Biswas, K. D. Devine, and J. E. Flaherty, *Parallel adaptive finite element method for conservation laws*, Applied Numerical Mathematics **14** (1984), 255–283.

[2] Lilia Krivodonova and Joseph E. Flaherty, *Error estimation for discontinuous galerkin solutions of two-dimensional hyperbolic problems*, Advances in Compu-

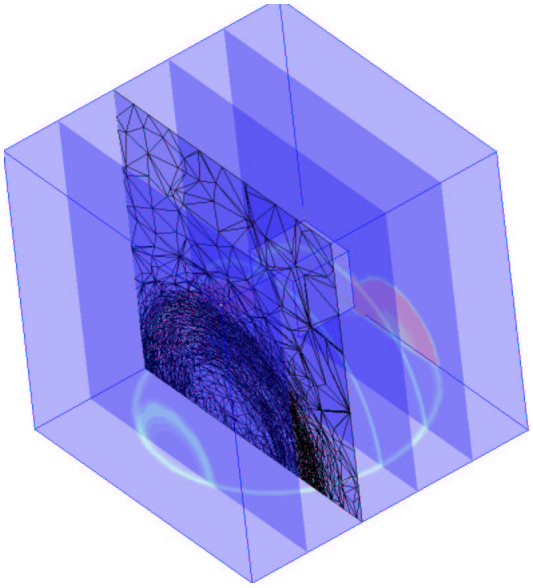


Figure 2: Cut of the mesh and some cuts of the density for the 3D Sedov problem.

tational Mathematics (2002), submitted.

[3] R. Le Veque, *Numerical methods for conservation laws*, Birkhäuser-Verlag, 1992.

[4] X. Li, M. S. Shephard, and M. W. Beall, *Accounting for curved domains in mesh adaptation*, International Journal for Numerical Methods in Engineering (2002), accepted.

[5] L. Sedov, *Similarity and dimensional methods in mechanics*, New York Academic, 1959.

[6] B. Van Leer, *Flux vector splitting for the Euler equations*, Tech. report, ICASE Report, NASA Langley Research Center, 1995.

[7] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, Journal of Computational Physics **54** (1984), 115–173.

MORPHING AND MODIFICATION OF UNSTRUCTURED MESHES

TIMOTHY J. BAKER[¶]

[¶]Dept. of MAE
Princeton University
Princeton, NJ 05844, U.S.A.

ABSTRACT: Unstructured meshes of triangles in 2D, or tetrahedra in 3D, are particularly well suited to mesh modification since local refinement and/or coarsening can be achieved without the introduction of hanging nodes or other artifices that often plague adaptation schemes for structured meshes. A dynamic mesh modification technique will be briefly described followed by a discussion of the element quality and element deformation measures that are needed to determine an appropriate region of the mesh for modification. A number of examples will be presented to illustrate the application of this dynamic adaptation method to the simulation of time dependent problems involving domains that undergo substantial deformation.

KEY WORDS: Mesh movement, coarsening, refinement, element quality measures.

Introduction

Mesh modification for time evolving domains can be carried out by a three stage combination of mesh movement, mesh coarsening and mesh enrichment. One application of this three stage procedure forms one cycle of dynamic adaptation. The extent of domain deformation that can be accommodated during one cycle depends on how far the mesh movement (*i.e.* r-refinement) stage can stretch the elements without creating an invalid mesh with negative element volumes. Mesh coarsening is then carried out to remove points associated with elements that have become badly shaped during the r-refinement stage. Finally, mesh enrichment serves to re-create a mesh whose element quality is comparable to that of the original mesh. This modification cycle can be repeated any number of times to obtain a good quality mesh for any homotopic deformation. At each stage, one is operating on a valid (*i.e.* conforming, space filling and non-overlapping) mesh which thus avoids the difficulties that are often associated with opening up pockets and remeshing empty regions of space.

1 Mesh movement

When mesh coarsening and enrichment is introduced as part of a dynamic adaptation cycle, it is not necessary to move the mesh as far as possible during each r-refinement stage. In fact, considerations of computational efficiency favor an r-refinement scheme that permits a smaller degree of mesh deformation if this can be accomplished at a much smaller computational cost than more robust mesh movement methods. In practice, a

[¶]baker@tornado.princeton.edu

robust mesh movement scheme is computationally twenty times more costly than the combined mesh coarsening and enrichment stages. The computational cost of the mesh coarsening and enrichment stages is therefore insignificant compared with the mesh movement stage. It is therefore advantageous to employ the simplest mesh movement scheme which typically incurs only five times the computational cost of the remaining mesh modification stages.

A particularly simple and cost effective r-refinement scheme is obtained if one approximates Laplace's equation by summing differences of the dependent variable along each edge incident to a mesh point. Let ϕ be the dependent variable and let $\nabla \cdot (\tau \nabla \phi) = 0$ be the equation to be solved. The variable diffusivity τ [1] is introduced to provide a degree of rigidity in regions that are occupied by elements whose volumes are extremely small. Mesh breakdown can be delayed by keeping the mesh displacement nearly constant in such regions while the introduction of a variable diffusivity involves negligible computational overhead. Let V denote the volume of an element, with V_{max} and V_{min} respectively the maximum and minimum element volumes. Define a variable diffusivity by

$$\tau = 1 + \frac{V_{max} - V_{min}}{V} \tag{1.1}$$

Clearly, $\tau = 1$ for a mesh whose elements all have the same size. For a mesh with varying element size, $\tau_{max} = \frac{V_{max}}{V_{min}}$ occurs for elements whose volume is equal to V_{min}.

If ϕ_0, respectively ϕ_k, represents the discrete approximation to ϕ at the mesh point P_0, respectively P_k, then the residual at P_0 is given by

$$\nabla \cdot (\tau \nabla \phi)|_0 \approx \frac{1}{S_0} \sum_{k=1}^{m} \sigma_{0k} (\phi_k - \phi_0) \tag{1.2}$$

where the k summation is over the m edges $\{\overline{P_0 P_k} | k = 1, \ldots, m\}$ incident to the point P_0, the coefficient $\sigma_{0k} = \sum_j \tau_j$ where the j summation is over all the elements incident to the edge $\overline{P_0 P_k}$ and the term $S_0 = \sum_{k=1}^{m} \sigma_{0k}$.

Using the superscript n to denote the nth iteration, one can write a point Jacobi scheme as

$$\phi_0^{n+1} = \phi_0^n + \frac{\varepsilon}{S_0} \sum_{k=1}^{m} \sigma_{0k} (\phi_k^n - \phi_0^n) \tag{1.3}$$

where $0 < \varepsilon \leq 1$ is a relaxation factor.

2 Mesh coarsening

The method of coarsening that is currently implemented uses edge collapse. Given a triangle that is a candidate for removal, take its shortest edge and allow the two endpoints of this edge to collapse to a single point. As a result, this edge together with the two incident triangles is removed from the triangulation and the number of points in the mesh decreases by one. The effect of this operation, when repeatedly applied to a region of the triangulation, will result in a much coarser distribution of points. Unfortunately, the quality of the triangulation is also severely degraded. The mesh enrichment algorithm described in the next section dramatically improves the mesh

quality, but even better results are obtained if the coarsened mesh is optimized prior to mesh enrichment. In the planar case, the coarsened mesh is optimized by swapping diagonals in order to maximize the minimum of the six angles of any pair of triangles with a common edge which form a convex quadrilateral. This operation will lead to the Delaunay triangulation of the coarsened point set for any planar triangulation [2].
A similar collapse procedure can be applied in three dimensions although the implementation is not so straightforward as in the planar case. It is necessary to identify the ring of tetrahedra incident to the candidate edge. The two end points of the edge are collapsed into a single point, the edge is removed and the ring of tetrahedra disappears from the mesh. It is necessary to check whether the modified tetrahedra are still valid (*i.e.* have positive volumes). In three dimensions, as in the planar case, it is advantageous to optimize the coarsened mesh prior to the enrichment stage. This is particularly important since slivers, which may have been created in the coarsened tetrahedral mesh, are not necessarily removed during mesh enrichment. Slivers are zero volume, or near zero volume, tetrahedra for which the ratio of circum-radius R to average edge length L is $O(1)$ [3]. These singular tetrahedra can be removed by edge/face swapping procedures [4].

3 Mesh enrichment

The decision, whether or not to enrich a particular region of the mesh, is based on a comparison between the actual local length scale h (*e.g.* element width, circum-radius) and the desired length scale specified by a scalar variable ρ called the length density function. Suppose, for example, that the goal is to convert an existing volume mesh into one with a smooth gradation in mesh density throughout the domain and such that the density of the volume mesh near the boundary surface matches the mesh size of the boundary triangulation. The value of the mesh density function at each point on the boundary is computed as the average length of the incident boundary edges. Solving Laplace's equation on the current volume mesh, using the values of the length density function at the boundary as Dirichlet data, will yield appropriate values of ρ at each mesh point. If the value of ρ at any position in the mesh is less than the actual local length scale h then the mesh is refined by the insertion of an extra point followed by a local mesh reconstruction using an incremental Delaunay algorithm. Several possibilities have been considered for selecting the position of point placement (*e.g.* at element barycenters [5], along edges [6], at element circum-centers [7] or along Voronoï segments [8]). The Voronoï segment method [8, 10] works extremely well in 2D but does not extend readily to 3D. The circum-center point insertion (CPI) method generates provably good quality meshes in 2D [9, 10]. In 3D the CPI method can generate meshes with good element quality provided care is taken to remove all slivers [4].
A element is thus marked for refinement if the circum-radius is too large when compared with the length density function ρ. The local mesh reconstruction exploits a constrained Delaunay algorithm that can be applied to any valid tetrahedral or triangular mesh. Elements to be replaced are those whose circumspheres contain the new point subject to the constraint that no boundary face should be removed and that the

new point is always visible from the exposed faces of the remaining elements. This particular procedure for inserting a new point is an integral component of the constrained Delaunay method that has been successfully exploited in a number of tetrahedral mesh generators [4, 5, 6].

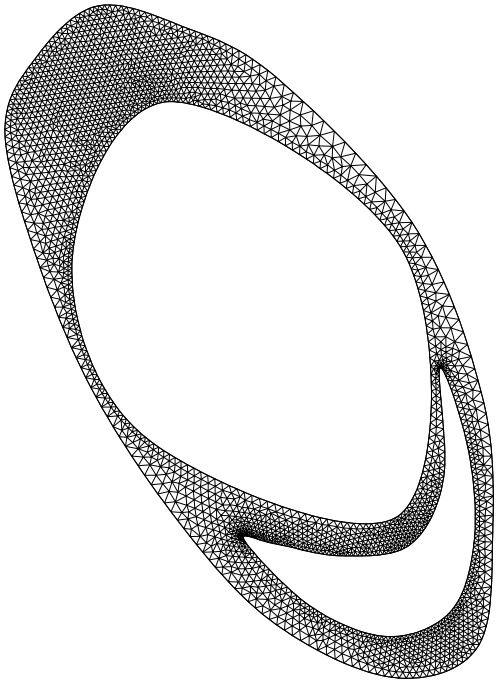


Figure 1: Heart slice at initial position.

4 Mesh deformation and quality measures

A key feature in the successful implementation of the dynamic adaptation procedure is the use of mesh deformation and quality measures to monitor the time evolution of the mesh. The extent of boundary deformation that can be tolerated during the r-refinement stage is not known *a priori*. It is therefore necessary to allow the option of automatically repeating the r-refinement stage with a smaller boundary deformation if any of the deformed elements has a negative volume.

More critical, however, is the need to identify those elements that should be removed during the coarsening stage. On the one hand, it is possible to classify elements according to a deformation measure that assesses the relative change in element shape as a result of mesh movement. On the other hand, one can exploit a quality measure to assess the instantaneous element shape or aspect ratio. It turns out that the most effective strategy is to coarsen the mesh based on a combined assessment of both element

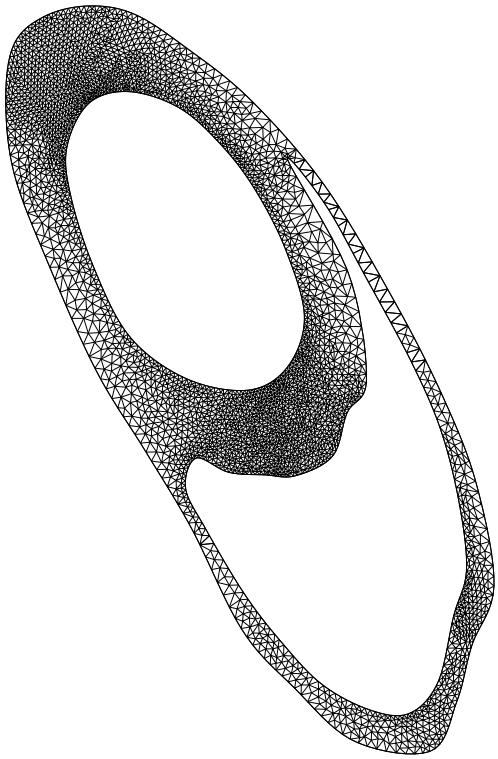


Figure 2: Change in heart slice after one quarter period.

deformation and element quality.

4.1 Deformation measure

Consider a tetrahedron whose vertices are given by the points with position vectors $\mathbf{x}_n$ where $n = 0,1,2,3$. The position vector $\mathbf{x}_n$ is a column vector of the three coordinates ($\mathbf{x}_n^T = (x_n, y_n, z_n)$). The edge matrix T_n is defined as

$$T_n = (-1)^n(\mathbf{x}_{n+1} - \mathbf{x}_n,\, \mathbf{x}_{n+2} - \mathbf{x}_n,\, \mathbf{x}_{n+3} - \mathbf{x}_n) \tag{4.1}$$

where addition in the indices is to be interpreted modulo four (*i.e.* $\mathbf{x}_{n+4} \equiv \mathbf{x}_n$) [11]. Thus, T_n is formed by the three edge vectors joining the vertex at $\mathbf{x}_n$ to each of the three remaining vertices. The determinant $|T_n|$ of the matrix is equal to six times the volume of the tetrahedron (twice the area of the triangle in the 2D case). If a right handed rule is assumed for the edge ordering then $|T_n| > 0$ for elements with positive volume.
In the definition of the edge matrix T_n the vertex n plays a special role. Call this the corner vertex for the edge matrix T_n. Now let F be the elementary column exchange

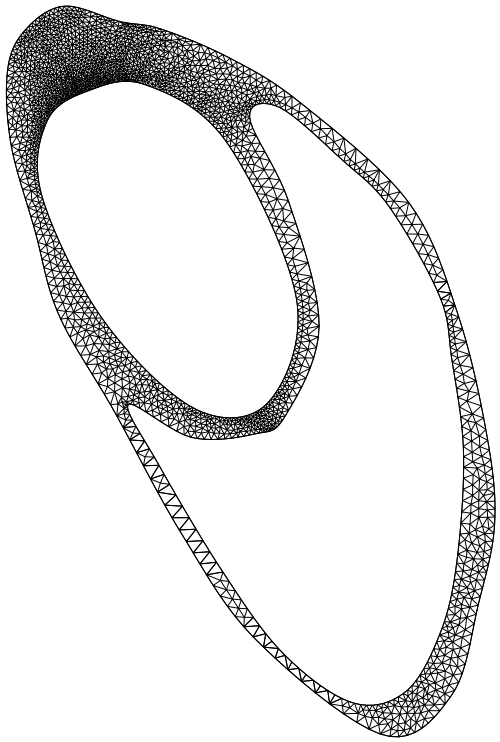


Figure 3: Change in heart slice after one half period.

matrix

$$F = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \tag{4.2}$$

It follows that

$$T_{n+1} = T_n F = T_0 F^n \tag{4.3}$$

For any given element there is a sequence of $d+1$ edge matrices $T_n, n = 0, \ldots, d$ where $d = 2$ for a triangle and $d = 3$ for a tetrahedron. It is clear that any intrinsic property of the element, such as a deformation measure or quality measure, should not depend on the choice of the corner vertex.

Suppose that the element defined by the edge matrix T_n is mapped, under the action of an r-refinement procedure, to an element whose corresponding edge matrix is $\tilde{T}_n$. Define the deformation matrix B_n associated with this mapping by

$$\tilde{T}_n = B_n T_n \quad \text{so that} \quad B_n = \tilde{T}_n T_n^{-1} \tag{4.4}$$

Then,

$$B_{n+1} = \tilde{T}_{n+1} T_{n+1}^{-1} = \tilde{T}_n F F^{-1} T_n^{-1} = B_n \tag{4.5}$$

The matrix B_n is therefore independent of the choice of the corner vertex n and one may, without any ambiguity, write B for the deformation matrix.
Now use the polar decomposition theorem to write B as $B = PU$ where U is a unitary matrix representing pure rotation and P is a positive definite matrix whose eigenvalues correspond to the modes of element distortion. An eigenvalue larger than 1 represents a stretching along the associated eigenvector while a compression is indicated by an eigenvalue less than 1. The eigenvalues of the dilatation matrix P, also known as the singular values of B, are found by taking the square root of the eigenvalues of BB^T. Note that B^TB and BB^T have the same eigenvalues even though their eigenvectors will generally be different.
Let σ_{max} and σ_{min} be the maximum and minimum singular values of B (*i.e.* the maximum and minimum eigenvalues of the dilatation matrix P). If B corresponds to a pure rotation then $\sigma_{max} = \sigma_{min} = 1$ and a uniform scaling by a factor μ would result in $\sigma_{max} = \sigma_{min} = \mu$.
When distortion is so severe that an eigenvalue of the dilatation matrix P becomes zero, the determinant of B and hence also the determinant of the new edge matrix $\tilde{T}_n$ is zero. It follows that the volume of the element has shrunk to zero and any further deformation will create elements with negative signed volumes. In practice, it seems prudent to repeat the mesh movement with a smaller boundary deformation if any of singular values falls below 0.1 in size.

4.2 Quality measure

Several options are available for constructing an element quality measure to assess the instantaneous element shape or aspect ratio. In references [12, 13] the ratio of average edge length to in-radius was used. Alternatively, one can construct a quality measure based on a unitarily invariant matrix norm [11, 15] by considering the deformation matrix associated with the transformation of a reference equilateral element (*i.e.* regular tetrahedron in 3D, equilateral triangle in 2D). A detailed discussion of quality measures and their properties for the 2D case can be found in [14].
Let T_n be an edge matrix for the element T and let W be the reference equilateral element. The associated deformation matrix is

$$A_n = T_n W^{-1} \tag{4.6}$$

In this case, since W is a fixed edge matrix, A_n will depend on the corner vertex n. However,

$$A_{n+1} = T_{n+1}W^{-1} = T_n F W^{-1} = A_n(WFW^{-1}) \tag{4.7}$$

It can be shown that $U = WFW^{-1}$ is an orthogonal matrix, from which it follows that $|||A_{n+1}||| = |||A_n|||$ for any matrix norm that is unitarily invariant. This result appears in reference [11] as theorem 3. Although reference [11] considers only the Frobenius norm,

$$|||A||| \equiv \|A\|_2 = (trace A^T A)^{\frac{1}{2}} \tag{4.8}$$

their proof generalizes to any unitarily invariant norm. A unitarily invariant norm $|||A|||$ is a matrix norm such that $|||UA||| = |||AV||| = |||A|||$ for any unitary (orthogonal) matrices U and V.

To verify that $U = WFW^{-1}$ is orthogonal we may choose any convenient representation for the reference equilateral element since an edge matrix is obviously invariant under translation, and U is clearly invariant under a uniform scaling of W. Consider a cube of side length 1 and centered at the origin. The inscribed tetrahedron defined by the vertex positions, $\mathbf{x}_0^T = (\frac{1}{2}, \frac{1}{2}, -\frac{1}{2})$, $\mathbf{x}_1^T = (-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2})$, $\mathbf{x}_2^T = (\frac{1}{2}, -\frac{1}{2}, \frac{1}{2})$, and $\mathbf{x}_3^T = (-\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ is evidently a regular tetrahedron with edge length $\sqrt{2}$. With the corner vertex at $n = 0$ we obtain the edge matrix,

$$W = \begin{pmatrix} -1 & 0 & -1 \\ -1 & -1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \tag{4.9}$$

whence we obtain

$$W^{-1} = \frac{1}{2} \begin{pmatrix} -1 & -1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix} \tag{4.10}$$

and

$$U = WFW^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{4.11}$$

It follows that $UU^T = I$ and $|U| = 1$.

We now define a quality measure for the element T to be a scalar variable $\tau(T)$ with the following properties:

i. $0 < \tau(T) \leq 1$,

ii. $\tau(T) = 1 \iff T$ is an equilateral element,

iii. $\tau(T)$ is invariant under translation, uniform scaling and rotation of T.

A unitarily invariant norm of a deformation matrix A is invariant under translation and rotation but not under a uniform scaling. This can be remedied [11] by constructing the condition number

$$\kappa(A) = \||A\|| \, \||A^{-1}\||. \tag{4.12}$$

It is known that a scalar function $N(A)$ of the matrix A is a unitarily invariant norm if and only if $N(A)$ is a symmetric gauge function of the singular values of A [16]. One example of a family of symmetric gauge functions is the family of ℓ^p norms. When applied to the singular values of a matrix they generate unitarily invariant norms known as the Schatten p norms [16],

$$N_p(A) = \left(\sum_{i=1}^{d} \sigma_i^p \right)^{\frac{1}{p}}, \qquad 1 \leq p < \infty \tag{4.13}$$

The case $p = 2$ is the Frobenius norm, the limiting case $p \to \infty$ is the spectral norm

$$\||A\||_2 = max\{\sigma; \sigma^2 \quad \text{is an eigenvalue of} \quad A^T A\} \tag{4.14}$$

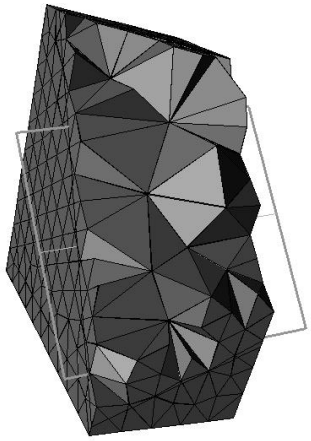


Figure 4: Box surface and cut through volume mesh.

and the case $p = 1$ is the trace norm,

$$||A||_{tr} = \sum_{i=1}^{d} \sigma_i(A) \tag{4.15}$$

The condition number $\kappa_p(A)$ based on the p norm is given by

$$\kappa_p(A)^p = \left(\sum_{i=1}^{d} \sigma_i^p \right) \left(\sum_{j=1}^{d} \sigma_j^{-p} \right) \tag{4.16}$$

If we choose the spectral norm, we see that $||A||_2 = \sigma_{max}$ and $||A^{-1}||_2 = \sigma_{min}^{-1}$ where σ_{max} and σ_{min} are the maximum and minimum singular values. The condition number $\kappa_\infty(A)$ based on the spectral norm is therefore

$$\kappa_\infty(A) = \frac{\sigma_{max}}{\sigma_{min}} \tag{4.17}$$

Evidently $\kappa_\infty(A) \geq 1$ with equality when $\sigma_{min} = \sigma_{max}$ which can only occur if the matrix A represents a uniform scaling and/or a rotation of the reference equilateral element. Using the spectral norm we can thus define the following quality measure for the element T,

$$\tau_\infty(T) = \frac{1}{\kappa_\infty(A)} = \frac{\sigma_{min}}{\sigma_{max}} \tag{4.18}$$

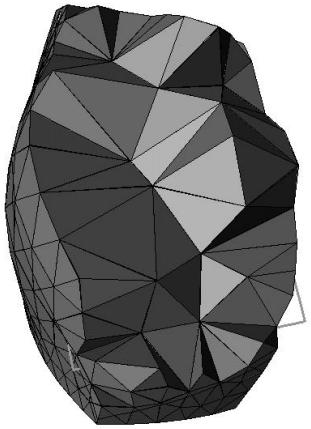


Figure 5: Rounded box surface and cut through volume mesh.

It can be shown [15] that the condition numbers and quality measures derived from the Schatten p norms are equivalent in the sense that

$$\kappa_\infty(A) \leq \kappa_p(A) \leq d^{\frac{2}{p}}\,\kappa_\infty(A) \tag{4.19}$$

and

$$\tau_\infty(T) \leq \tau_p(T) \leq d^{\frac{2}{p}}\,\tau_\infty(T) \tag{4.20}$$

In order to identify the elements that should be removed after the mesh movement stage, it is necessary to consider both the deformation measure (*i.e.* the shape measure derived from the matrix defined by (4.4)) as well as the absolute element quality given by the shape measure based on the matrix defined in (4.6). The deformation measure assesses the relative change in element aspect ratio. The majority of elements that need to be removed will have small values for their deformation measure. It is, however, possible for some elements adjacent to a solid surface to undergo a small deformation during each mesh movement stage and yet experience a large cumulative deformation over several time steps. Since the deformation measure only monitors the relative change between successive cycles, this pathological behavior will not be detected by the deformation measure alone. We therefore need the element shape measure which assesses the absolute element aspect ratio to detect these cases.
On the other hand, the element aspect ratio or quality measure alone tends to detect several elements that exist in the initial mesh and which, being a long way from the

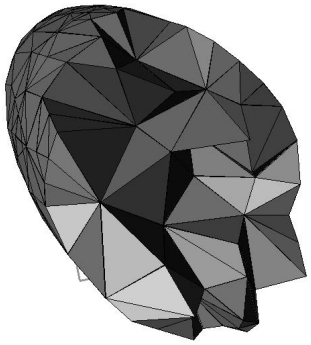


Figure 6: Ellipsoidal surface and cut through volume mesh.

region of deformation, do not change their shape during mesh movement. A successful strategy to identify the appropriate region for coarsening uses both measures. In other words, a element is marked for removal if either its deformation measure is moderately small or if its quality measure is extremely small.

5 Examples

The first example, presented in figures 1, 2 and 3, illustrates an application of the dynamic adaptation procedure in the planar case. Using MRI imaging, it is possible to measure the shape of a heart at three or four instants of time during a cardiac period. One can obtain a reasonably accurate representation of the change in the shape of the heart by interpolating between these snapshots. Figure 1 shows a 2D lateral slice through the middle of a heart. The meshed domain consists of heart muscle in the region bounded by the pericardium on the outside and the two ventricles on the inside. During one cardiac period the heart muscle changes shape dramatically and the ventricles alternately contract and expand by considerable amounts. The sequence of figures shows the change in shape during one half of a cardiac period. The change from figure 1 to figure 2, which represents one quarter of a cardiac period, was accomplished in 16 mesh modification cycles. Another 16 mesh modification cycles transformed the mesh in figure 2 to the mesh, shown in figure 3, that conforms with the shape of the heart slice after one half of a cardiac period. During the second half of the cardiac period, the heart slice retraces its shape change and the mesh continues to adapt dynamically to conform with the evolving domain.
The next five figures illustrate a example of dynamic adaptation applied to a 3D domain that starts as a box, morphs into a sphere, then into an ellipsoidal shape and finally

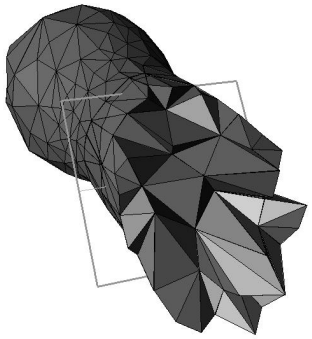


Figure 7: Elongated surface and cut through volume mesh.

into a dumbbell. Each figure shows a cut through the volume mesh and the exposed faces of those tetrahedra that intersect the cutting plane. Part of the triangulation on the boundary surface is also evident. The vizualization software used to investigate the 3D meshes and produce figures 4 through 8 was provided by Pascal Frey [17]. The shape change for this toy example was defined analytically. Both coarsening and refinement procedures were applied to the surface triangulation in order to maintain an acceptable triangle quality on the boundary surface. The adaptation sequence during each modification cycle thus consists of (i) movement of the boundary surface points according to a prescribed analytic expression, (ii) movement of the tetrahedral mesh to conform with the new boundary position, (iii) coarsening and enrichment of the boundary surface triangulation to recover an acceptable surface mesh quality, and (iv) coarsening and enrichment of the volume triangulation. The coarsening and refinement procedures that are applied to the boundary triangulation directly modify the tetrahedral mesh so that the volume mesh conforms at all times to the triangulation of the boundary surface.

6 Acknowledgments

Support for this work has been provided by NASA Ames under grant NAG-2-1327.

References

[1] A. MASUD & T.J.R. HUGHES A Space-time Finite Element Method for Fluid-structure Interaction, *SUDAM Report No. 93-3*, Stanford University, Stanford, CA, 1993.

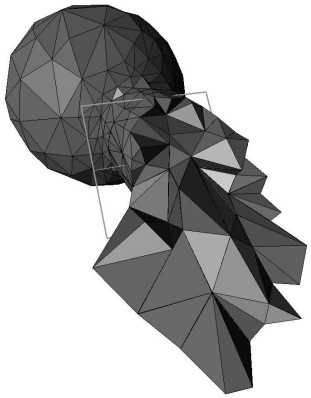


Figure 8: Dumbbell surface and cut through volume mesh.

[2] C.L. LAWSON Software for C^1 surface interpolation, *Mathematical Software III*, pub. Academic Press, 161-194, 1977.

[3] T.J. BAKER Element Quality in Tetrahedral Meshes, *Proc. 7th Int. Conf. on Finite Element Methods in Flow Problems*, 1018-1024, 1989.

[4] T.J. BAKER & J.C. VASSBERG Tetrahedral Mesh Generation and Optimization, *Proc. 6th International Conference on Numerical Grid Generation*, pub. ISGG, 337-349, 1998.

[5] O. HASSAN, L.B. BAYNE, K. MORGAN, & N.P. WEATHERILL An adaptive unstructured mesh method for transient flows involving moving boundaries, *EC-COMAS98*, 668-674, Sept. 1998.

[6] P.L. GEORGE & H. BOROUCHAKI *Delaunay Triangulation and Meshing*, pub. Hermes, 1998.

[7] D.G. HOLMES & D.D. SNYDER The generation of unstructured triangular meshes using Delaunay triangulation, *Proc. 2nd International Conference on Numerical Grid Generation*, pub. Pineridge Press, 643-652, 1988.

[8] S. REBAY Efficient Unstructured Mesh Generation by means of Delaunay Triangulation and Bowyer-Watson Algorithm, *J. Comp. Physics*, 125-138, 1993.

[9] P. CHEW Guaranteed Quality Mesh Generation for Curved Surfaces, *Proc. 9th Symp. Comp. Geom.*, pub. ACM Press, 274-280, 1993.

[10] T.J. BAKER Triangulations, Mesh Generation and Point Placement Strategies, *Frontiers of Computational Fluid Dynamics 1994*, (ed. D.A. Caughey, M.M. Hafez), pub. J.Wiley and Sons, 1994.

[11] L.A. FREITAG & P.M. KNUPP Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number, *8th Int. Meshing Roundtable*, 1999.

[12] T.J. BAKER & P.A. CAVALLO Dynamic adaptation for deforming tetrahedral meshes, *AIAA 14th CFD Conf.*, AIAA Paper 99-3253, 1999.

[13] J.H. PREVOST, T.J. BAKER, J. LIANG & Z. SUO A finite element method for stress-assisted surface reaction and delayed fracture, *Int. J. Solids and Structures* **38**, 5185-5203, 2001.

[14] P.P. PÉBAY, & T.J. BAKER Analysis of Triangle Quality Measures, to be published in *Math. Comp.*, 2003.

[15] T.J. BAKER Deformation and quality measures for tetrahedral meshes, *Eccomas00*, Barcelona, Spain, Sept. 2000.

[16] R.A. HORN & C.R. JOHNSON *Matrix Analysis*, pub. CUP, 1985.

[17] P.J. FREY Medit: Interactive Mesh Vizualization Software, *INRIA Tech. Report RT-0247*, January 2001.

THE MESQUITE TOOLKIT

P.M. KNUPP[¶]

Sandia National Laboratories
Albuquerque, NM, U.S.A.

ABSTRACT: Poor mesh quality is known to adversely affect both solution efficiency and accuracy. There has been considerable research on a wide variety of mesh improvement algorithms. The impact of these algorithms has been somewhat limited because they are often embedded in particular meshing or application software packages. A stand-alone, freely available software package would make the most effective algorithms available to many more applications. To this end, we are developing the MESQUITE software toolkit for mesh quality improvement. The toolkit will also serve as a platform for further research into mesh quality improvement methods. We describe the motivation, goals, and design of MESQUITE as well as give some computational results using the underlying algorithms that show the benefit of the package.

KEY WORDS:

[¶]pknupp@sandia.gov

LOW MACH NUMBER REACTING FLOW MODELING WITH STRUCTURED ADAPTIVE MESH REFINEMENT

H. NAJM[*¶], J. RAY[¶], C. KENNEDY[¶], R. MCCOY[¶],
S. LEFANTZI[¶], P. PÉBAY[¶] & W. KOLLMANN[†],

[¶]Sandia National Laboratories
7011 East Ave., MS 9051
Livermore, CA 94550, USA

[†]University of California, Davis
Mechanical & Aeronautical Engineering
Davis, CA 95616, USA

ABSTRACT: We discuss the numerical challenges associated with the modeling of reacting flow with detailed chemical kinetics, with particular emphasis on the low Mach number formulation. We highlight the need for adaptive mesh refinement, and discuss structured adaptive mesh refinement strategies. Finally, we present some results of reacting flow modeling with structured adaptive mesh refinement.

KEY WORDS: Structured, Adaptive, Mesh, Refinement, Reacting, Flow.

Introduction

There are significant challenges with the computations of reacting flow with detailed kinetics. These stem primarily from chemical complexity, where accurate description of the chemistry of even the simplest hydrocarbon fuel (e.g. methane: CH_4) requires on the order of 30 chemical species and 200 elementary reaction steps. Thus, one is generally faced with many governing equations per mesh cell, and a complex/expensive chemical source term evaluation. One also typically encounters a wide range of time scales in detailed reacting flow modeling. These range from $\sim$1 ms flow time scales to $\sim$1 ns fastest chemical time scales. This leads to significant stiffness in the governing equations, necessitating implicit or partially implicit constructions to ensure time integration stability at acceptable time step sizes. Using an explicit time integration construction, one is forced to use time steps on the order of 1 ns. Moreover, flame reaction zones exhibit very sharp spatial gradients, with hydrocarbon radical profile widths of about 100 μm [1]. This necessitates the use of very small mesh cell sizes ($h \sim 15\,\mu$m) in order to adequately resolve the reaction zone structure. The utilization of highly refined meshes, coupled with the large diffusivity of the H-radical, leads to a significant explicit time step limitation ($\sim$20 ns) in the integration of the diffusive terms. This again requires the utilization of either implicit or operator-split stabilized-explicit diffusion time integration approaches.

[*]hnnajm@ca.sandia.gov

The large computational cost associated with the evaluation of the reaction source terms necessitates highly optimized source code implementation to accelerate these evaluations. On modern computational architectures, with hierarchical memory structures, it is memory access rather than floating point operation count that is the computational bottleneck. Experience with detailed chemistry computations are consistent with this observation. The source code optimizations necessary to reduce the cost of the chemical source term evaluations have to do more with memory utilization than with reduction of floating point operation counts. In particular, using meta-codes to process the manually-coded source term and jacobian routines and produce completely unrolled and 'hard-wired' source code (valid for a particular chemical mechanism because of the implemenation of reaction rate parameters as constants) has been found to have a significant impact on computational speed. We have experienced a 30-50% overall code speedup with these hard-wired source term evaluations on SGI hardware, depending on the L2 cache size. The smaller the cache, and/or the slower the memory bus speed, the larger is the payoff.

We have also used additive implicit-explicit (IMEX) schemes to integrate the chemical source term implicitly, and the convective-diffusive transport terms explicitly [2], in order to accelerate the code performance. Using, DVODE [3] for integration of stiff chemical source terms has been effective in ameliorating the difficulties associated with time-integration stiffness of the chemistry. With IMEX schemes, one is left with the explicit diffusion time step restriction as the primary bottleneck in time integration. We have addressed this using operator-splitting, where the chemistry is integrated implicitly with DVODE, while the transport terms are integrated in symmetrically split 1/2-time steps [4]. By specifically isolating the integration of the diffusion terms within the operator-split diffusional half-steps, we have been able to utilize either fractional time stepping or stabilized explicit Runge-Kutta-Chebyshev schemes to integrate the diffusional source terms efficiently.

It is important to note that the utilization of operator splitting needs to be done with care. Specifically, when both stiff and non-stiff operators are involved, as does the reaction-diffusion splitting, then it is advantageous to let the last operator in the time step be the stiff operator [5]. This is because the fast time scales in the stiff term define an algebraic manifold that the solution tends to stay on [6]. These fast time scales bring the solution quickly to the manifold. On the other hand, the non-stiff (diffusion) term allows the solution to drift from the manifold by a small amount in a 1/2-time step. While experience has shown that this drift is not noticeable with the actual species mole fractions, the computed source terms for the species associated with the fast modes are indeed significantly affected. Thus, in the context of reaction-diffusion splitting based on a second-order symmetric Strang [7] splitting, it is preferrable to use an $R_{\Delta t/2}D_{\Delta t}R_{\Delta_t/2}$ operator sequence rather than the $D_{\Delta t/2}R_{\Delta t}D_{\Delta_t/2}$ sequence. We have used both approaches in different contexts. We find that the results computed with the $D_{\Delta t/2}R_{\Delta t}D_{\Delta_t/2}$ sequence with sufficiently small time steps ($\sim \leq 100$ ns for a methane flame) are acceptable. On the other hand, at larger time steps, splitting errors can lead to unphysical spatial structure of the chemical source terms. Yet, when this happens, the actual time integration results are acceptable, because the stiff integration

procedure in DVODE quickly brings the solution to the manifold in the beginning of each chemistry step, using very small time steps. We have found that a relaxation of the computed large-time-step solution with very small time steps ($\sim$2 ns) for a short time interval ($\sim$40 ns) is sufficient to bring the solution back to the manifold such that physical chemical source terms are observed.

We note that the loss of local-truncation-error accuracy for the splitting schemes ending with the slow operator is associated only with the fast modes. Moreover, we note that stiffness has a stabilizing effect, and the local errors for fast modes do *not* propagate. The only error which propagates is that of the slow species, which is second order for symmetrically-split integrators [5].

It is important to point out that a fully implicit construction is not generally an adequate approach. The highest convective flow speed and smallest flow length scale impose a necessary upper limit on the time step for accuracy purposes, even if larger time steps were stable using an implicit scheme. Further, if the smallest mesh cell size is of the order of the smallest length scale, then this accuracy-relevant time step limitation is effectively the same as the explicit convective CFL stability limitation. Thus it is more efficient to integrate the convection terms explicitly since the time step has to be that small anyway, hence the need for some IMEX or operator-split implementation.

This leaves the issue of the mesh. Clearly, it is not advisable to use uniform meshes, because the high degree of spatial refinement necessary in the reaction zone leads to an unnecessarily fine mesh density everywhere, thereby limiting the computations to relatively small domains. Using a uniform 15 μm mesh within a (small) 10 cm computational domain would require in excess of 6500 mesh cells in each dimension. Thus, it is advantageous to use non-uniform meshes. Moreover, the fact that flame reaction zones propagate by burning and/or convective transport indicates that the solver has to track the reaction zone(s) and adjust the mesh accordingly. In order to avoid global mesh changes resulting from local flame movement, it is generally better to avoid using moving-mesh solutions to this problem, and to employ instead local adaptive mesh refinement (AMR) and coarsening.

We have used structured adaptive mesh refinement (SAMR) constructions to compute two-dimensional premixed flame ignition and non-premixed jet flames. The following sections outline our experiences with some of these computations.

1 Low Mach Number Formulation

We are concerned with generally low speed laboratory-scale flames. These flames typically have flow speeds on the order of 10 m/s, such that $M \approx 0.03$ and $M^2 \approx 0.001$, thereby justifying the low Mach number approximation [8] used below.

The physical model is based on extending the formulation developed in [2, 4]. The

model relies on the zero-Mach-number limit of the compressible conservation equations [8]. In this limit, acoustic waves are ignored and the pressure field is decomposed into a spatially-uniform component $P_0(t)$ and a hydrodynamic component $p(x,t)$ which varies in space and time. The model assumes a gas mixture with zero bulk viscosity [9], and ignores Soret and Dufour effects [10], as well as body forces and radiant heat transfer.

Under the above assumptions, the non-dimensional governing equations are expressed as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0 \tag{1.1}$$

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u^2)}{\partial x} + \frac{\partial (\rho uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\Phi_x \tag{1.2}$$

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho vu)}{\partial x} + \frac{\partial (\rho v^2)}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\Phi_y \tag{1.3}$$

$$\frac{\partial T}{\partial t} + v \cdot \nabla T = \frac{1}{\mathrm{RePr}} \frac{\nabla \cdot (\lambda \nabla T)}{\rho c_p} + \frac{1}{\mathrm{ReSc}} \frac{1}{c_p} \sum_{i=1}^{N} c_{p,i} \mathbf{V}_i \cdot \nabla T + \mathrm{Da} \frac{w_T}{\rho c_p} \tag{1.4}$$

$$\frac{\partial (\rho Y_i)}{\partial T} = -\nabla \cdot (\rho v Y_i) - \frac{1}{\mathrm{ReSc}} \nabla \cdot (\rho Y_i V_i) + \mathrm{Da}\, w_i \tag{1.5}$$

Here, ρ is the density, T is the temperature, $v = (u,v)$ is the velocity vector, Y_i is the mass fraction of species i, μ is the dynamic viscosity, λ is the thermal conductivity, c_p is the mixture specific heat, w_i is the chemical production rate of species i, w_T is rate of chemical heat release, V_i is the diffusion velocity of species i, $c_{p,i}$ is the heat capacity of species i, Re, Pr, Sc, and Da are the Reynolds, Prandtl, Schmidt, and Damköhler numbers respectively, while Φ_x,Φ_y are the viscous stress terms.

The mixture is assumed to obey the perfect gas law, with individual species molecular weights, specific heats, and enthalpies of formation. The equation of state is expressed as:

$$P_0 = \rho T / \overline{W} \tag{1.6}$$

where $\overline{W} \equiv 1/\left(\sum_{i=1}^{N} Y_i/W_i\right)$ is the local molar mass of the mixture, N is the total number of species, and W_i is the molecular weight of species i. Note that for an open domain P_0 is constant. The specific heat of the mixture is given by:

$$c_p = \sum_{i=1}^{N} Y_i c_{p,i} \tag{1.7}$$

where $c_{p,i}$ is the specific heat of the i-th species at constant pressure.

We use the GRImech1.2 [11] chemical mechanism for modeling methane-air chemistry. This mechanism has 32 species and 177 reactions, and corresponds to the reaction network diagram shown in Fig. 1.

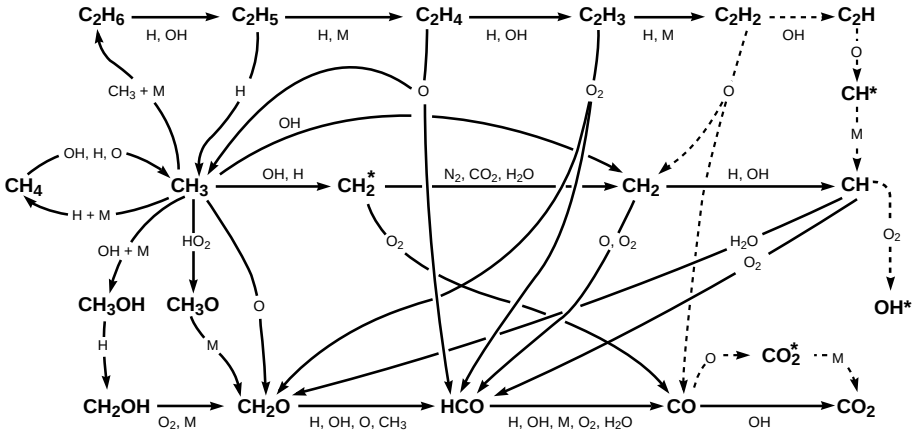


Figure 1: Reaction network diagram for premixed methane-air flame chemistry with GRImech1.2 [11].

2 SAMR Implementations

The past two decades have seen significant development of techniques for SAMR of both compressible and low Mach number flow. Berger & Oliger [12] presented some of the earliest work in this area, using mesh adaptation based on superposition of locally refined multiple levels of rectangular mesh patches. This was followed by further developments by Berger & Colella [13]. A number of investigations by Colella and coworkers focused on the implementation of these techniques using rectangular mesh patches for both compressible and low Mach number reacting and non-reacting flows [14–21]. A range of other investigations also focused on rectangular patch SAMR [22–24], with specific applications in astrophysics [25–27]. There has also been work on the implementation of rectangular patch-SAMR in the context of discrete ordinate computations of radiation heat transfer [28], and for coupling eulerian flow computations with direct simulation monte-carlo particle computations [29].

Other SAMR developments relaxed the rectangular uniform mesh patch constraint [30–33], allowing for stair-case mesh boundaries. Smooke and co-workers used similar mesh constructions in the context of both steady and unsteady bunsen flame computations [34–36]. There has also been work coupling these mesh constructions with Lagrangian vortex method [37] computations of jet flames [38–40].

In general these techniques deal with a hierarchy of meshes $M_i,\ i = 0,1,\ldots,N$, where mesh M_0 is the coarsest mesh level, with grid cell size h_0, covering the whole computational domain. Each higher level mesh M_i spans a subset of mesh M_{i-1} and is characterized by a grid cell size $h_i = h_{i-1}/q$ where typically $q = 2$.

Spatial derivative discretizations lead to the coupling of mesh levels at internal mesh

boundaries. Discretizations on patches of mesh M_i use information from M_i for internal patch mesh points, and from M_{i-1} for neighbor cells at mesh boundaries. There are generally two options for dealing with discretizations at mesh/patch boundaries. One approach uses specialized stencils combining points from M_i and M_{i-1}. This approach involves significant complexity, increasing with the number of spatial dimensions. It requires a "library" of stencils for each distinct derivative computation. Another approach, which we have used, uses a buffer of "green" cells at the level of M_i around the mesh/patch perimeter, whose field values are found by interpolation from M_{i-1} data. This construction requires additional data structures and interpolation operations, however, it leads to much reduced stencil complexity. It has the particular advantage of requiring the utilization of only uniform-Δx derivative discretizations on all mesh levels. Finally, one-sided derivatives are typically implemented at domain boundaries.

SAMR implementations also require the utilization of spatial interpolation for various purposes. Coarse-to-fine interpolation is required when refining mesh regions, in order to define data on newly created children of refined coarser-level mesh cells. Further, coarse-to-fine interpolation is required for evaluating "green" cell data around mesh/patch boundaries if required. On the other hand, fine-to-coarse interpolation procedures are required for updating coarse-mesh cells with data from their refined children upon evaluation of the solution at a new time level. One-sided interpolants are generally necessary near domain boundaries.

SAMR time integration constructions typically use finer time steps on finer mesh levels in accordance with numerical stability constraints. When the time step is controlled by the convective CFL condition, then $\Delta t_i = \frac{1}{2}\Delta t_{i-1}$. On the other hand, when it is viscous CFL-controlled, then $\Delta t_i = \frac{1}{4}\Delta t_{i-1}$. In either case, recursive time stepping algorithms are utilized for time stepping on the various mesh levels. An example time integration procedure for forward euler, using convective-CFL time step control, is as follows

$$\begin{aligned} U_0^{n+1} &= U_0^n + F_0^n \Delta t_0 \\ U_1^{n+1/2} &= U_1^n + F_1^n \Delta t_1 \\ U_1^{n+1} &= U_1^{n+1/2} + F_1^{n+1/2} \Delta t_1 \\ &\cdots \end{aligned}$$

where, U_i^n is the solution vector on mesh level i at time level n, and $F_i^n = F(U_i^n)$. Note that $F_1^{n+1/2}$ evaluated on a fine mesh at internal mesh/patch boundaries requires $U_0^{n+1/2}$ at the coarse-mesh neighbors. This necessitates integration of U_0 before U_1 and time interpolation between U_0^n and U_0^{n+1} to provide $U_0^{n+1/2}$.

We note that the solution of the low Mach number equations using projection methods [41] involves the solution of an elliptic equation for the pressure field. This makes the low Mach number solution on SAMR meshes more challenging than that of compressible flow equations where no such elliptic constraints are encountered. The complexity arises primarily from the continuity constraint at internal mesh/patch boundaries. Colella and coworkers have outlined detailed 'synchronization' approaches for

dealing with this issue [21]. These involve the solution of a second elliptic equation for the velocity-error correction necessary to satisfy continuity at mesh boundaries. These globally-coupled elliptic equations are effectively solved using a multigrid procedure on the mesh hierarchy. Note that a projection on the velocity field u leads to a Helmholz equation for the pressure $\nabla(\frac{1}{\rho}\nabla p) = \sigma$, while a projection on (ρu) results in a pressure Poisson equation $\nabla^2 p = \tilde{\sigma}$.

Finally, we note that a significant challenge for parallel AMR computations is the need for a good *dynamic* load balancing strategy. This is necessary for maintaining good parallel scalability with increasing numbers of processors, and for enabling massively parallel computations that take advantage of advanced parallel computational hardware. Typically, large ranges of spatial scales lead to a high degree of local refinement, which leads to highly localized computational effort where local refinement has lead to multiple mesh levels. Reacting flow computations typically lead to such refinements in the flame reaction zones. Moreover, the fast rates of change of species concentrations in the flame zone lead to the need for many small time steps within the stiff time integration procedure (e.g. with DVODE). In fact, the highest chemistry integration costs per mesh cell are in the flame zone irrespective of the mesh topology. The compounded effect of high local mesh density and higher computational cost per mesh cell leads to significant localization of computational work in the mesh regions where the flame reaction zone exists. This work needs to be distributed efficiently among the rest of the processors if scalable parallel performance is desired. Of course, at sufficiently high flow Reynolds numbers, flow turbulence levels increase the need for high mesh refinement at locations outside the flame reaction zone. Moreover, if high turbulence is found everywhere in the domain, and it requires mesh cells of the same size as those necessary in the flame, then AMR is simply not efficient, and a single uniform mesh computation is most optimal.

3 Jet Flame Ignition SAMR Computations

In the following we show some results from a jet flame study using a coupled Lagrangian-Eulerian vortex-method/SAMR construction The details of the numerical construction are discussed elsewhere [38–40]. We consider a planar jet flowing vertically into a rectangular 2D 10×5cm domain. The jet fuel is N_2-diluted CH_4, and is buffered on either side by a coflow stream of air. The domain side walls are far from the jet, and are assumed to be slip impermeable walls. The lower edge of the domain is a specified inlet, while the upper edge employs an outflow boundary condition that enforces global mass conservation.

The flow is initialized at room temperature and left to reach a steady state. At a designated time a transient heating pulse is applied to one side of the left jet shear layer. The gas ignites and a circular ignition front propagates away from the ignition point. Two such ignition kernels are shown in Fig. 2, ignited on either side of the stoichiometric mixture fraction line in the jet shear layer. Only a small segment of the domain is

shown here to examine the ignition kernel growth in detail. In either case, ignition in the air (top frame sequence) or fuel (bottom frame sequence) streams, the ignition kernel grows until it arrives at, and passes through, the mixing layer. As the front approaches the stoichiometric mixture fraction line, it encounters ideal burning conditions, sharpens, and accelerates significantly. It then deccelerates as it proceeds into leaner/richer regions, with progressively lesser fuel/oxidizer concentration. The front segment propagating away from the stoichiometric mixture fraction line simply degrades in burning intensity and propagation speed as it moves into richer/leaner regions with very sparse oxidizer/fuel available.

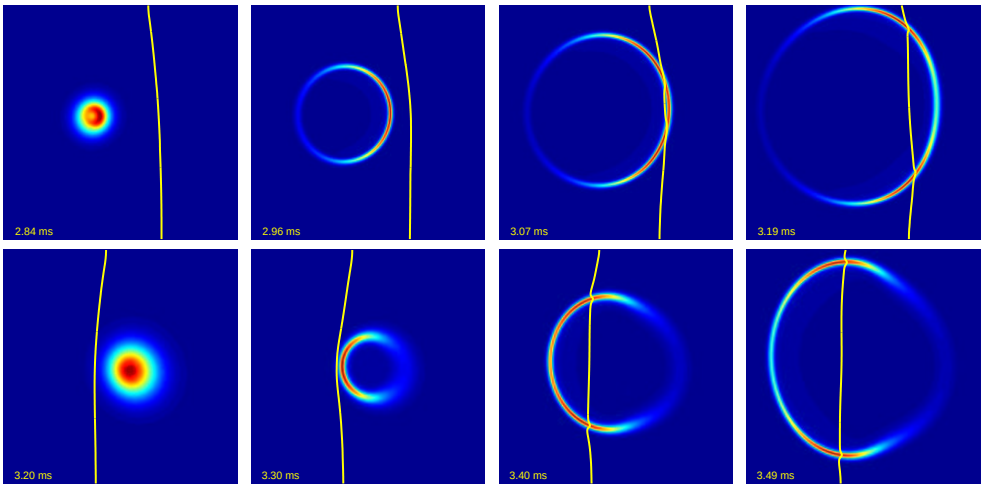


Figure 2: Ignition fronts in a jet shear layer. Jet fuel (CH_4) is on the right and coflow air is on the left of the stoichiometric mixture-fraction line shown.

The detailed structure of the ignition kernel and the associated adaptive mesh structure are shown in Fig. 3. The flame heat release rate is plotted in the left frame, with superposed contours of the mixture fraction, while the adaptive mesh is shown on the right. Note the sharpness of the heat release profile, and the corresponding high resolution mesh architecture. This figure outlines clearly the necessity of an adaptive mesh architecture in order to efficiently capture this transient flame kernel. It also illustrates the severeness of the dynamic load-balancing challenge, as much of the computational work is concentrated in a very small and non-stationary subset of the computational domain.

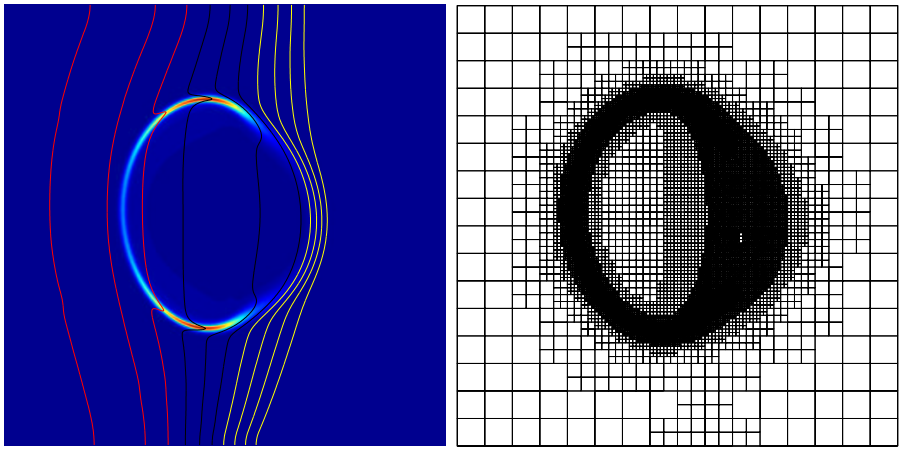


Figure 3: Ignition front and associated adaptive mesh structure. The left frame shows the flame heat release rate, with the superposed contours of the mixture fraction showing the progression of the mixture fraction from lean to stoichiometric to rich in going from left to right. The right frame shows the adaptive mesh at this instant of flow evolution.

The laterally propagating ignition flame kernel segments soon extinguish as they reach regions of pure air or pure fuel on either side of the stoichiometric mixture fraction line. Two edge flames are left over, characteristic of burning into a partially premixed flow, propagating upstream and downstream along the stoichiometric mixture fraction line. The edge flames propagate away from the ignition region, as shown in Fig. 4. The top flame proceeds upwards and leaves through the domain exit, while the lower flame proceeds downwards towards the jet inlet and (if a stable lifted jet flame exists under these flow/flame conditions) stabilizes against the incoming stream at a location where its burning speed exactly balances the flow velocity in the shear layer. In this manner, a diffusion flame is established, stretching from the lifted edge flame at the jet inlet, up towards the domain exit.

The final topology and mesh-structure of the lower (nearly-stabilized) edge flame are shown in Fig. 5. The figure shows the edge flame in the right frame, using the CO_2 field, with superposed heat release rate contours. The adaptive mesh structure in the

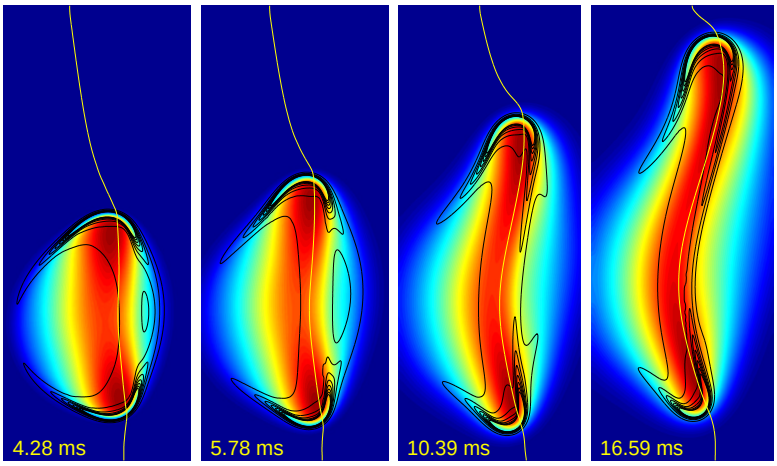


Figure 4: Evolution of the two edge flames along the stoichiometric mixture fraction line away from the ignition region. Heat release rate contours are shown superposed on the mole fraction of CO_2.

edge flame region is shown in the left frame, again indicating the degree of localized mesh refinement required to capture the flame structure accurately.

4 GrACE SAMR Implementation

We also illustrate some results from reaction-diffusion computations using the GrACE SAMR toolkit [42]. We have implemented a SAMR component, based on GrACE, within the Common Component Architecture (CCA) [43].

GrACE uses a rectangular patch-based SAMR architecture. Mesh patches are stored in trees, in a Directed Acyclic Graph (DAG). The DAG structure is hidden from users. The Berger & Rigoutsos algorithm [44] is used for clustering points into patches. Inverse space filling curves [45–47] are implemented for load partitioning. Time refinement is based on the convective CFL condition. GrACE has been used for a wide range of applications ranging from geomechanics to astrophysics [42].

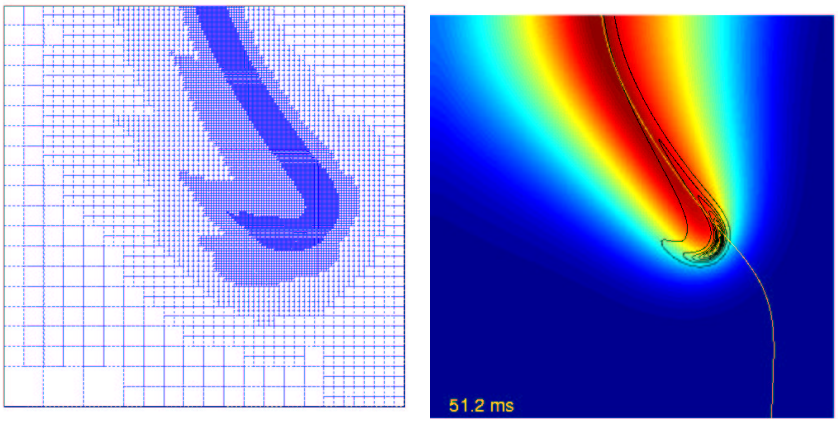


Figure 5: Edge flame structure at the base of the jet flame. Left frame shows the mesh topology, while the right hand frame shows the heat release rate contours superposed on the CO_2 mole fraction.

GrACE uses dynamic load balancing, with an adaptive choice of partitioners [45]. The choice is based on balancing tradeoffs between different measures of mesh goodness, including computational time, scatteredness of adaptation, and the volume of data transfer during rebalance operations. Geometric multilevel inverse space-filling curves (ISP) [46] and sequence partitioners (SP) [48] are used as the basis for the load balancing strategy. Generally, space filling curves provide a higher emphasis on speed rather than the quality of the partitioned structure [45].

The CCA framework [43] provides a flexible, massively parallel, and efficient means of assembling a code. The visual design of a GrACE-based reaction diffusion code is shown in Fig. 6.

We present results from a reaction-diffusion study of ignition in hydrogen-oxygen mixtures [49,50]. The flow is initialized with a uniformly premixed H_2-O_2 mixture at room temperature. Ignition is initiated at three spots using localized transient heating pulses. Only the reaction-diffusion equations are accounted for, with the heat release rate artificially zeroed, and no momentum equations solution. This is primarily for demonstration and evaluation of the GrACE SAMR functionality on a parallel distributed linux platform. The evolution of the ignition fronts is shown in Fig. 7, where they expand initially towards each other until they meet and annihilate each other, leading to a final state of completely burnt products mixture. Fig. 8 shows the adaptive mesh topology in the full computational domain (left) and in an amplified region where strong flame curvature is evident.

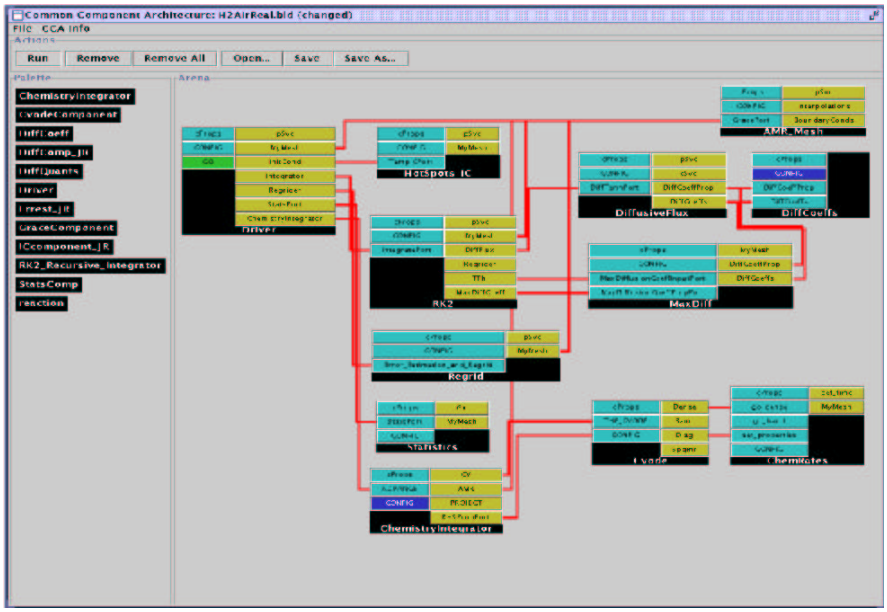


Figure 6: CCA/GrACE AMR Component Assembly for H_2-O_2 Ignition.

Conclusions

We have presented a brief discussion of the utility of structured adaptive mesh refinement algorithms for detailed computations of reacting flow. SAMR is necessary for enabling large-scale reacting flow computations with detailed chemical kinetics and resolved flame structure. SAMR preserves uniform mesh discretization operations, thereby simplifying spatial discretization strategies. Further, rectangular patch-based SAMR allows the utilization of efficient matrix-vector array data structure operations.

We have described the low Mach number governing equations formulation. We have illustrated low Mach number methane-air reacting flow results with detailed kinetics using a Lagrangian-Eulerian SAMR construction, and a reaction-diffusion implementation of homogeneous hydrogen-air ignition using the fully Eulerian SAMR GrACE toolkit within a CCA framework. Results highlighted the degree of mesh adaptation necessary for capturing detailed reacting flow structure and dynamics. They also highlighted the need for very good dynamic load balancing strategies to maintain parallel code scalability.

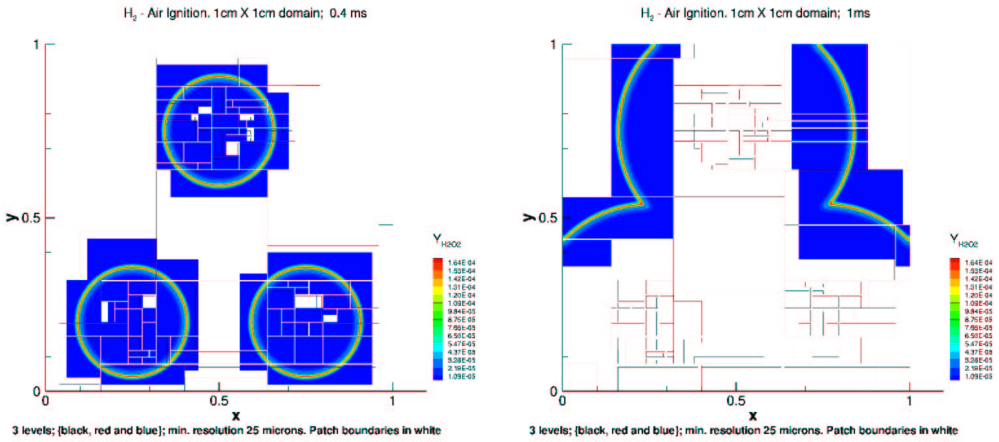


Figure 7: CCA/GrACE AMR Computations of H_2-O_2 Ignition. The left frame shows the evolution of the reaction-diffusion fronts at early time, followed by a late time snapshot in the right frame.

Acknowledgements

Support was provided by the US Dept. of Energy (DOE), Office of Science, Office of Basic Energy Sciences (BES), Division of Chemical Sciences, Geosciences, and Biosciences, and by the DOE BES SciDAC Computational Chemistry Program.

References

[1] Najm, H.N., Knio, O.M., Paul, P.H., and Wyckoff, P.S., *Comb. Sci. Tech.*, 140(1-6):369–403 (1998).

[2] Najm, H.N., Wyckoff, P.S., and Knio, O.M., *J. Comp. Phys.*, 143(2):381–402 (1998).

[3] Brown, P.N., Byrne, G.D., and Hindmarsh, A.C., *SIAM J. Sci. Stat. Comput.*, 10:1038–1051 (1989).

[4] Knio, O.M., Najm, H.N., and Wyckoff, P.S., *J. Comp. Phys.*, 154:428–467 (1999).

[5] Sportisse, B., *J. Comp. Phys.*, 161:140–168 (2000).

[6] Valorani, M., Najm, H.N., and Goussis, D., *Combustion and Flame* (2003) in press.

[7] Strang, G., *SIAM J. Numer. Anal.*, 5(3):506–517 (1968).

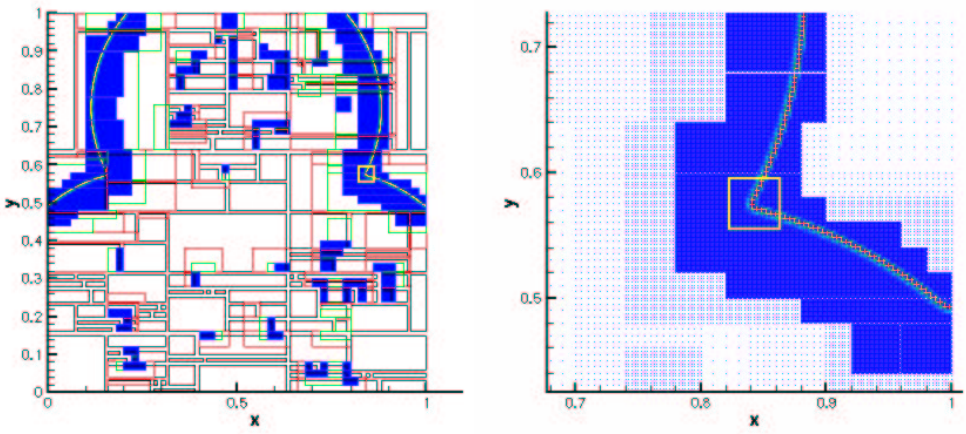


Figure 8: GrACE Meshes over the full computational domain (left) and a magnified flame-front region (right).

[8] Majda, A., and Sethian, J., *Comb. Sci. and Technology*, 42:185–205 (1985).

[9] Schlichting, H., *Boundary-Layer Theory*, McGraw-Hill, New York, 7^{th} edition, (1979).

[10] Williams, F.A., *Combustion Theory*, Addison-Wesley, New York, 2nd edition, (1985).

[11] Frenklach, M., Wang, H., Goldenberg, M., Smith, G.P., Golden, D.M., Bowman, C.T., Hanson, R.K., Gardiner, W.C., and Lissianski, V., Top. Rep. GRI-95/0058, GRI, (1995).

[12] Berger, M., and Oliger, J., *J. Computational Phys.*, 53:484–512 (1984).

[13] Berger, M.J., and Colella, P., *Journal of Computational Physics*, 82:64–84 (1989).

[14] Bell, J.B., and Colella, P., *J. Comp. Phys.*, 85:257–283 (1989).

[15] Bell, J.B., and Marcus, D.L., *J. Comput. Phys.*, 101:334–348 (1992).

[16] Almgren, A.S., Bell, J.B., P.Colella and Howell, L.H., *11th AIAA Computational Fluid Dynamics Conference Orlando, Florida*, Orlando, Florida, Lawrence Livermore National Laboratory, pp. 1–10, (1993).

[17] Almgren, A.S., Bell, J.B., and Szymczak, W.G., Technical Report UCRL-JC-112842, Lawrence Livermore National Laboratory, Livermore, California, (1993).

[18] Pember, R.B., Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., and M.Lai Preprint UCRL-JC-118634, Lawrence Livermore National Laboratory, Livermore, CA, (1994).

[19] Pember, R.B., Howell, L.H., Bell, J.B., Colella, P., Crutchfield, W.Y., Fiveland, W.A., and Jesse, J.P., Technical Report LBL-38551, Lawrence Berkeley National Laboratory, Berkeley, CA, (1997).

[20] Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., and Welcome, M., *J. Comput. Phys.*, 142:1–46 (1998).

[21] Martin, D.F., and Colella, P., *J. Comp. Phys.*, 163:271–312 (2000).

[22] Lemke, M., and Quinlan, D., *Communications in Applied Numerical Methods*, 8:609–619 (1992).

[23] Quirk, J.J., and Hanebutte, U.R., NASA Contracter Report and ICASE Report NASA Contracter Report 191530 ICASE Report No. 93-63, NASA, Hampton, VA, (1993).

[24] Quirk, J.J., NASA Contracter Report-ICASE Report NASA Contracter Repory 194938 and ICASE Report No. 94-51, NASA, Hampton, VA, (1994) Contract NAS1-19480.

[25] Bryan, G.L., and Norman, M.L., in *12^{th} 'Kingston Meeting': Computational Astrophysics* (D. Clarke and M. West, Eds., volume 123 of *ASP Conference Series*, (1997), pp. 363–368, .

[26] Bryan, G.L., and Norman, M.L., in *Structured Adaptive Mesh Refinement (SAMR) Grid Methods* (S. Baden N. Chrisochoides D. Gannon and M. Norman, Eds., volume 117 of *IMA Volumes in Mathematics and its Applications*, Springer, (1999), pp. 165–170, .

[27] S. Baden N. Chrisochoides D. Gannon and M. Norman, Eds. *Structured Adaptive Mesh Refinement (SAMR) Grid Methods*, volume 117 of *IMA Volumes in Mathematics and its Applications*, Springer, (1999).

[28] Jessee, J.P., Fiveland, W.A., Howell, L.H., Colella, P., and Pember, R.B., Report LBNL-38800 U-405, LBNL, (1996).

[29] Garcia, A.L., Bell, J.B., Crutchfield, W.Y., and Alder, B.J., Technical Report LBNL-42907, Ernest Orlando Lawrence Berkeley National Laboratory, (1999).

[30] Biswas, R., Devine, K.D., and Flaherty, J.E., Technical Report SAND92-7341, Sandia National Labs., Albuquerque, NM, (1992).

[31] Wheat, S.R., Devine, K.D., and Maccabe, A.B., Technical Report SAND93-2172A, Sandia National Labs., Albuquerque, NM, (1993).

[32] Devine, K., Ph.d. dissertation, Rensselaer Polytechnic Institute, Troy, NY, (1994).

[33] Devine, K., and Flaherty, J., *Appl. Numer. Math.*, 20:367–386 (1996).

[34] Bennett, B.-A.V., and Smooke, M.D., *Combust. Theory Model.*, 2:221–258 (1998).

[35] Bennett, B.-A.V., and Smooke, M.D., *J. Comput. Phys.*, 151:684–727 (1999).

[36] Bennett, B.-A.V., McEnally, C.S., Pfefferle, L.D., and Smooke, M.D., *Combustion and Flame*, 123:522–546 (2000).

[37] Chorin, A.J., and Bernard, P.S., *J. Comp. Phys.*, 13:423–429 (1973).

[38] Najm, H.N., Schefer, R.W., Milne, R.B., Mueller, C.J., Devine, K.D., and Kempka, S.N., Technical Report SAND98-8232, Sandia National Labs., Alb., New Mexico, (1998).

[39] Najm, H.N., Milne, R.B., Devine, K.D., and Kempka, S.N., *ESAIM Proceedings*, volume 7, pp. 304–313, (1999).

[40] Ray, J., Najm, H.N., Milne, R.B., Devine, K.D., and Kempka, S.N., *Proc. Comb. Inst.*, 28:219–226 (2000).

[41] Chorin, A.J., *J. Comput. Phys.*, 2:12–26 (1967).

[42] GrACE Homepage, http://www.caip.rutgers.edu/ parashar/TASSL/.

[43] Allan, B. A., Armstrong, R. C., Wolfe, A. P., Ray, J., Bernholdt, D. E., and Kohl, J. A., *Concurrency: Practice and Experience*, 14:323–345 (2002) Also at http://www.cca-forum.org/ccafe03a/index.html.

[44] Berger, M., and Rigoutsos, I., *IEEE Transactions on Systems, Man and Cybernetics*, 21 (1991).

[45] Steensland, J., Ph. D. Thesis., University of Uppsala, Uppsala University Library, Box 510, SE-751, 20 Uppsala, Sweden, (2002).

[46] Ou, Chao-Wei, and Ranka, Sanjay, *Journal of Parallel and Distributed Computing*, 42:109–121 (1997).

[47] Pilkington, J., and Baden, S., *IEEE Transactions on Parallel and Distributed Systems*, 7:288–300 (1996).

[48] Olstad, Bjorn, and Manne, Fredrik, *IEEE Transactions on Computers*, 44:1322–1326 (1995).

[49] Lefantzi, Sophia, and Ray, Jaideep, *Proceedings of the Second MIT Conference on Computational Fluid and Solid Mechanics*, Boston, Mass., Elsevier Science, (2003).

[50] Lefantzi, S., Ray, J., and Najm, H. N., *Proceedings of the International Parallel and Distributed Processing Symposium*, Nice, France, (2003).

NEW RESULTS ON QUALITY MEASURES FOR PLANAR QUADRILATERAL MESHES

PHILIPPE P. PÉBAY*

Sandia National Laboratories
MS 9051 P.O. Box 969
Livermore, CA 94550, U.S.A.

ABSTRACT: This article examines the quality assessment of planar quadrilateral mesh elements in a comprehensive way. First, an analytic characterization of quadrangular shape is provided, and existing concepts of *stretching* and *skewness*, earlier proposed by for specific geometries, are generalized. Then, two triangle quality measures are extended to quadrilaterals and their respective extremal and asymptotic behaviors examined, showing in particular that they cannot detect, if needed, triangular degeneracy of a quadrilateral. An existing quality measure is then discussed, which is able to handle this case. In particular, an unbalanced asymptotic behavior is demonstrated, justifying the need for a new approach. Toward this goal, the triangle quality measure based on FROBENIUS norm is modified in order to replace equilateral reference element by right isosceles triangles, with control on the specific right angle. Finally, two new quadrilateral quality measures are designed and examined using these results. Numerical results illustrate the matter.

KEY WORDS: quadrilateral, quadrilateral meshing, mesh quality.

Introduction

It is now widely known that the geometric properties of the mesh supporting a finite element or finite volume computation directly impact the accuracy of the numerical result. For example, in the particular case of finite element analysis of elliptic problems, [3] shows that accuracy of the approximate solution is directly related to a geometric estimate of mesh elements, generally referred to as *aspect ratio*, in the case where these elements are simplicial.

Although triangular and tetrahedral quality measures have been extensively discussed, in particular in [1, 2, 5, 6, 9, 11, 12], little work has been done concerning planar quadrilaterals. For such elements, [13] proposes estimating the quality by means of the *aspect ratio*, the *skewness* and the *stretching factor*. Although natural for some particular geometries, such measures are not well defined in a general context. More general quadrilateral quality estimates have been proposed, but neither closely examined nor compared, in [7]. An effort towards providing an in-depth analysis of several planar quadrilateral measures can be found in [10], which discusses in particular an extension to quadrilaterals of triangle quality measures based on the FROBENIUS matrix norm. More recently, [8] has proposed a global framework for such extensions.

*pppebay@ca.sandia.gov

This work was supported by the United States Department of Energy, Office of Defense Programs & Office of Basic Sciences, SciDAC Computational Chemistry Program.

After recalling a few useful geometrical results, this article firstly discusses the analytical characterization of planar quadrilaterals. In particular, asymptotic degeneracy cases are examined. The modification of triangle quality measures based on matrix norms, proposed in [10] in order to adapt these measures to the case where reference elements are right isosceles triangles, is provided *in extenso*. This adaptation is motivated by the idea of extending them to quadrilateral quality measure. Then, the first quadrangle shape estimates to be considered are skewness and stretching factor. They are examined in detail and it is explained why they cannot provide quadrangle quality measures in a general context. Because of these limitations, extensions of some triangle quality measures to quadrilaterals are then provided. These quality measures are fully analyzed and it is shown that they satisfy the desired extremal and, depending on the context, asymptotic properties. When these asymptotic properties do not comply with the requirements of the application, an alternate estimate introduced in [7] is examined in detail. Finally, the quadrilateral quality measures based on the FROBENIUS matrix norm, previously tailored to right isosceles reference elements, are defined and studied. Several numerical examples illustrate the theoretical results.

1 Preliminaries

1.1 Triangles

In all that follows, t denotes a triangle with vertices E, F and G, area $\mathcal{A}_t$, semiperimeter p_t, inradius r, circumradius R, edge lengths $e = FG$, $f = EG$ and $g = EF$, and we denote the angle at vertex E (resp. F, G) as η (resp. ϕ, ψ) and the radius of the inscribed (resp. circumscribed) circle of t as r (resp. R). In addition, the vertices E, F and G have respective coordinates (x_E, y_E), (x_F, y_F) and (x_G, y_G) in an arbitrary orthonormal affine reference frame parallel to the plane of the triangle t. The following standard norm-like notations will also be used:

$$\begin{array}{rcl} |t|_0 & = & \min(e,f,g) \\ |t|_2 & = & \sqrt{e^2+f^2+g^2} \\ |t|_\infty & = & \max(e,f,g). \end{array} \tag{1.1}$$

Some results from elementary geometry are assumed without proof (see for example [4] for proofs and details). In particular, the following well-known relations will be used:

$$2R = \frac{efg}{2\mathcal{A}_t} = \frac{e}{\sin\eta} = \frac{f}{\sin\phi} = \frac{g}{\sin\psi}, \tag{1.2}$$

where $\mathcal{A}_t$ is given by

$$\mathcal{A}_t = r p_t, \tag{1.3}$$

as well as by HERON's formula:

$$\mathcal{A}_t = \sqrt{p_t(p_t-e)(p_t-f)(p_t-g)}. \tag{1.4}$$

Finally, it is recalled that the edge ratio is defined as:

$$\tau = \frac{|t|_\infty}{|t|_0}, \tag{1.5}$$

see [11] for a study of the behavior of τ, with respect to the extremal angles of t.

1.2 Quadrilaterals

For the sake of conciseness, "quadrilateral" will hereafter mean a "planar, non-degenerate and convex quadrilateral". Such a quadrilateral shall be denoted generically as $q = ABCD$, with area $\mathcal{A}$, semiperimeter p, edges of lengths $a = AB$, $b = BC$, $c = CD$ and $d = DA$ and denote the angle at vertex A (resp. B, C, D) as α (resp. β, γ, δ) and θ the arithmetic mean of either α and γ, or β and δ, called the *torsion*. Norm-like notations, similar to those for triangles, will also be used:

$$\begin{array}{rcl} |q|_0 & = & \min(a,b,c,d) \\ |q|_2 & = & \sqrt{a^2+b^2+c^2+d^2} \\ |q|_\infty & = & \max(a,b,c,d). \end{array} \tag{1.6}$$

Remark 1.1. The computation of $\mathcal{A}$ can be performed in several different ways. In particular, it seems natural to decompose q in two triangles, whose respective areas are obtained in a straightforward manner obtained. Although twice more costly, it is not a bad idea to compute the four possible triangular areas, since this allows the detection, on the fly, whether or not q is convex, non-convex, skew, degenerate (*cf.* [7] for details). Practically speaking, this is of the greatest interest, since both topological consistency checking and geometrical quality measurement can be done at the same time.

Most of the useful metric equalities of triangles do not extend to quadrilaterals, and this is the first obstacle to the generalization of results such as those presented in [11] in the case of triangles. Nevertheless, HERON's formula can be generalized for quadrilaterals:

$$\mathcal{A} = \sqrt{(p-a)(p-b)(p-c)(p-d) - abcd\cos^2\theta}. \tag{1.7}$$

Remark 1.2. This provides an opportunity to discuss the choice of α and γ in the definition of θ. It is well known that the sum of the four angles of a convex quadrilateral is equal to 2π; hence, $\theta_1 = \frac{\alpha+\gamma}{2}$ and $\theta_2 = \frac{\beta+\delta}{2}$ are supplementary. It follows in particular that θ_1 and θ_2 have opposite cosines, thus equal squared cosines. Therefore, whatever pair of opposite angles is picked in order to define θ, (1.7) returns the same result, *i.e.*, it is symmetrical.

1.3 Cocyclicity

It might be useful to recall some results about planar cocyclicity, and in particular the fact that, unlike triangles, quadrilaterals cannot necessarily be inscribed in a circle. More precisely,

Definition 1.1. A quadrilateral is *cyclic* if it can be inscribed in a circle.

Example 1.1. The quadrilateral @, illustrated Figure 1, which is homothetic to the quadrilateral with successive vertices coordinates $(\sqrt{3},1)$, $(\sqrt{2},\sqrt{2})$, $(1,\sqrt{3})$ and $(-\sqrt{3},1)$, is cyclic.

A demonstration of the beautiful *Inscribed Angle Theorem* can be found in any good elementary geometry handbook; using the classical notation for vector angles, it comes as follows:

Theorem 1.1. *P, Q, R and S are cocyclic if and only if:*

$$(\overrightarrow{PQ}, \overrightarrow{PS}) \equiv (\overrightarrow{RQ}, \overrightarrow{RS}) \quad [\pi] \tag{1.8}$$

and allows to readily deduce the

Corollary 1.2. *q is cyclic if and only if* α *and* γ *are supplementary.*

Remark 1.3. According to Remark 1.2, Corollary 1.2 can be equivalently expressed using β and δ instead of α and γ.

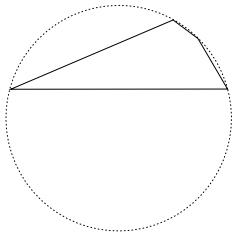


Figure 1: Cyclic quadrilateral @.

1.4 Deriving quadrilateral quality from triangle quality

A natural approach to measure the quality of any given non-degenerate convex quadrilateral consists in seeing it as a pair of two non-degenerate triangles sharing one common edge, which is also a diagonal of the quadrilateral. Hence, an apparently good idea would be to examine the qualities of these two triangles, but which quality ? Generally speaking (see [11] for a notable exception), the quality of a triangle is considered to be optimal[1] only for equilateral triangles. Unfortunately, the following example shows that using such triangle quality measures for quadrilaterals is not straightforward.

Example 1.2. Figure 2 illustrates the case where *q* is a rhombus, such that its shortest diagonal has the same length as its edges, hence denoted as ◊. By definition, ◊ can be decomposed either into two equilateral triangles or into two obtuse isosceles triangles. In the general sense of triangle quality, the former case is considered as optimal, while the latter is far from this. In other words, the choice of the particular partition of *q* in two triangles has an effect over the resulting quadrilateral quality measurement; which one shall be chosen ?

The only certainty at this point is that either both triangular decompositions of *q* must be taken into account, or another approach of quadrilateral quality, independent from the underlying triangles, must be used.

[1] more precisely, reaches its strict and unique *minimum*, 1.

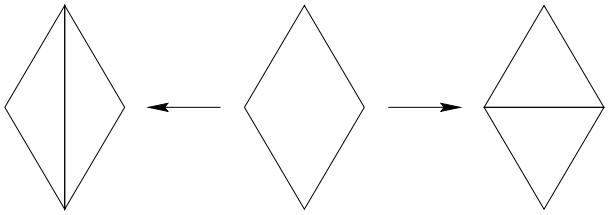


Figure 2: The two possible triangulations of the same rhombus ◊.

2 The quadrilateral space

The aim of this section is to define a proper set on which the analysis of quadrilateral quality measures will be relevant. In particular, the concept of equivalence class will be used.

2.1 Homothecy equivalence classes

A quadrilateral can be seen as either a geometric or an analytic object. Although the former is certainly more intuitive, the latter allows the use of calculus to perform an analysis of quadrilateral quality measures. A consistent analytic representation of quadrilaterals is one that is bijective with the set of geometric quadrilaterals. In addition, quality measures in the general sense (*cf.* [7, 11]) do not depend on size, but only on shape; in other words, they are invariant through homothecy.
Hence, it is more suitable to use an analytical representation of quadrilaterals, up to homothecy.
Equality up to homothecy is an equivalence class, in the strict mathematical sense: reflexive, symmetrical and transitive. Therefore, it allows the definition of equivalence classes of quadrilaterals; in particular, any quadrilateral q belongs to one and only one equivalence class, and this class is the set of all quadrilaterals which are homothetic to q. In addition, any equivalence class can be represented by one of its elements, *e.g.*, the only quadrilateral with unitary semiperimeter, as shown in Figure 3. In other words, considering only the set of such quadrilaterals is sufficient for quality measure analysis, since these measures must be invariant to scaling. This set, which is in fact the set of all equivalence classes[2], is simply denoted as $\mathcal{Q}_1$.

2.2 Analytic characterization

Since any edge length of a non-degenerate quadrilateral q is strictly smaller than the sum of the three other ones, it follows that

$$\frac{p}{a} = \frac{a+b+c+d}{2a} = \frac{1}{2} + \frac{b+c+d}{2a} > \frac{1}{2} + \frac{a}{2a} = 1 \tag{2.1}$$

[2] or, equivalently, the quotient space of the set of quadrilaterals by the equality up to homothecy.

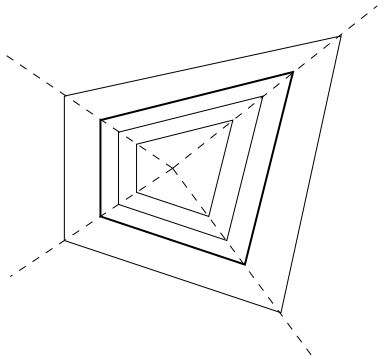


Figure 3: Equality up to homothecy: the unique quadrilateral with unitary semiperimeter represents the entire homothecy class.

and, for the same reasons, $\frac{p}{b} > 1$, $\frac{p}{c} > 1$ and $\frac{p}{d} > 1$. Hence, denoting as x, y and z the ratios between three edge lengths to the semiperimeter of q, *e.g.*, $x = \frac{a}{p}$, $y = \frac{b}{p}$ and $z = \frac{c}{p}$, it is clear that $x < 1$, $y < 1$ and $z < 1$. In addition,

$$x + y + z = \frac{a+b+c}{p} = \frac{2p-d}{p} = 2 - \frac{d}{p}, \tag{2.2}$$

whence

$$1 < x + y + z < 2 \tag{2.3}$$

and

$$0 < 2 - x - y - z = \frac{d}{p} < 1. \tag{2.4}$$

Hence, on the one hand, the quadrilateral q_1 with consecutive edge lengths x, y, z and $2-x-y-z$ is homothetic to q (with ratio p); on the other hand, its semiperimeter is obviously unitary thus $q_1 \in \mathbb{Q}_1$. In other words, q_1 is the class representative of q, on which quality measure analysis shall be performed.
Now, the knowledge of the four edge lengths of a quadrilateral is not sufficient to determine its shape[3]. For example, knowing that a quadrilateral has four equal edge lengths only permits to conclude that it is a rhombus; nothing is known about the angles of this rhombus which might be, in particular, a square.

Remark 2.1. This makes a noticeable difference with the triangular case, for which there is a bijection between, on the one hand, edge lengths ratios and, on the other hand, the angles of this triangle.

In fact, the knowledge of a, b, c, d and, *e.g.*, the angle α, completely determines q and, in particular, the other angles β, γ and δ, for an obvious reason: knowing α permits to determine one of the diagonals, therefore the triangle opposite to α is fully determined

[3] or, equivalently, the knowledge of x, y and z is not sufficient to determine the homothecy equivalence class.

by its three edges, according to Remark 2.1. Simply stated, this means that, in addition to edge lengths, quadrilaterals have one and only one other degree of freedom, provided by any of their four angles. Thus, the set of quadrilaterals can be seen as a subset of $\mathbb{C}\ \mathbb{R}^5$ and, since angles are invariant through homothecy, $\mathbb{Q}_1$ can be interpreted as a subset of $\mathbb{R}^4$. In other words, the quadrilateral shape space is four-dimensional[4].

2.3 Asymptotic configurations

As $\mathbb{Q}_1$ is four-dimensional, it is not as natural to examine the asymptotic behavior, as in the case of triangles, in particular because it is impossible to visualize[5] a hypersurface of R^5. Nevertheless, examining some usual configurations can provide useful informations concerning some lower-level shape estimates, such as the torsion or the stretching factor. For example, the set of parallelograms up to homothecy has the nice property of being a 2-dimensional subset of $\mathbb{Q}_1$, since $a = c$ and $b = d$. Moreover, these two degrees of freedom can be straightforwardly seen as the stretching factor and the torsion, and this allows a direct relationship between the behavior of a quality measure to these parameters.
Another interesting asymptotic case occurs when two consecutive edges of q tend towards alignment, in other words when q gets close to being a triangle t, as shown Figure 4. Although this is a somewhat intuitive notion, this situation can be formalized as follows: precisely,

Definition 2.1. A *triangular degeneracy* occurs when one, and only one, angle of a quadrilateral is close to π and none is close to 0.

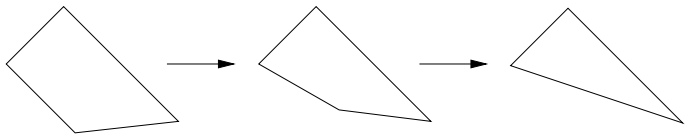


Figure 4: Triangular degeneracy.

Triangular degeneracy might be considered as either acceptable or not, *i.e.*, the quality measure should either remain bounded or diverge to $+\infty$, respectively. For example, in the context of hybrid meshes, it could be useful to have a triangle-quadrilateral quality measure, which would not diverge in the case of triangular degeneracy of q, but rather continuously match along the transition between the two kinds of polygons. Contrarily, most quadrilateral finite elements solvers would poorly handle triangularily degenerate elements.
Provided these results concerning the analytical characterization of quadrilateral shape, it is now possible to examine precisely the extremal and asymptotic properties of several possible quadrilateral quality measures.

[4] intuitively, since no vector field structure has been properly defined.
[5] at least, for human beings.

3 Edge ratio

It seems natural to extend τ, the edge ratio which has been defined for triangles, to quadrilaterals. In this case, we have:

$$\tau = \frac{|q|_\infty}{|q|_0}. \tag{3.1}$$

Remark 3.1. τ can be seen as a generalization of the stretching factor, defined in [13] for specific geometries. Therefore, it seems rather intuitive to extend the use of this expression to all quadrilaterals, and τ will be also referred to as the *stretching factor* of q.

3.1 *Extremum*

By definition, $\tau \geq 1$, with equality if and only if $|q|_\infty = |q|_0$, *i.e.*, $\min(a,b,c,d) = \max(a,b,c,d)$, thus $a = b = c = d$. Therefore, τ has a unique absolute *minimum*, 1, attained only by rhombii. Squares are only a particular minimal case.

3.2 Asymptotic behavior

3.2.1 Parallelogram

Obviously, if q is a parallelogram, then either $\tau = \frac{b}{a}$ or $\tau = \frac{a}{b}$. In particular, τ is completely independent of the angles of q, which is not the case when considering the edge ratio of a triangle. Intuitively, the behavior of τ can be seen as one-dimensional, with respect to the four dimensions of Q_1. This is why it has been proposed for specific geometries, in particular for rectangles: in terms of homothecy equivalence classes, they are completely characterized by their length to width ratio, *i.e.*, by τ. Minimal for and only for squares, it increases and tends to $+\infty$ as the rectangle stretches, hence its name. To the contrary, τ is unable to provide any valuable information when q is a rhombus: even very flattened, with θ as close to 0 as desired, it is considered as optimal as a square by τ.

3.2.2 Triangle degeneracy

If q degenerates towards a triangle t, the two following different cases may occur:

i. if two vertices tend to merge, then $|q|_0$ tends to 0. Thereore, τ diverges to $+\infty$;

ii. otherwise, if the triangle degeneracy occurs with no edge lengths tending to 0, then $|q|_0$ tends to a non-zero value while $|q|_\infty$ remains bounded, preventing τ from degenerating.

In addition, a continuation of τ to the edge ratio of the triangle occurs only if $|q|_0$ and $|q|_\infty$ remain unchanged, when jumping from q to t.
Clearly, neither the asymptotic nor extremal properties of q correspond to what is expected from a quadrilateral quality measure. However, it is a very intuitive dimension of the quadrilateral space, thus useful for examining the behavior of such measures.

Remark 3.2. The edge ratio also provides a triangle quality measure, optimal only for equilateral triangles. It has the original property of accepting flattened elements, but not needles for which it diverges. See [11] for details.

4 Skewness

Skewness does not seem to have been properly defined in a general context. We propose the following definition:

$$\sigma = \frac{1}{1 - \left|1 - \frac{2\theta}{\pi}\right|}. \tag{4.1}$$

As mentioned in Remark 1.2, the two possible values of θ, θ_1 and θ_2, are supplementary thus

$$\left|1 - \frac{2\theta_1}{\pi}\right| = \left|1 - \frac{2(\pi - \theta_2)}{\pi}\right| = \left|-1 + \frac{2\theta_2}{\pi}\right| = \left|1 - \frac{2\theta_2}{\pi}\right|, \tag{4.2}$$

from which it follows that σ is independent from the particular choice of either θ_1 or θ_2 for θ. More precisely, σ is symmetrical with respect to $\frac{\pi}{2}$.

4.1 *Extremum*

By definition, one has

$$\begin{aligned}
0 < \theta < \pi \quad &\Longleftrightarrow \quad 0 < \frac{2\theta}{\pi} < 2 && (4.3)\\
&\Longleftrightarrow \quad -1 < 1 - \frac{2\theta}{\pi} < 1 && (4.4)\\
&\Longleftrightarrow \quad 0 \leq \left|1 - \frac{2\theta}{\pi}\right| < 1 && (4.5)\\
&\Longleftrightarrow \quad 0 < 1 - \left|1 - \frac{2\theta}{\pi}\right| \leq 1 && (4.6)\\
&\Longleftrightarrow \quad \sigma \geq 1. && (4.7)
\end{aligned}$$

In addition,

$$\begin{aligned}
\sigma = 1 \quad &\Longleftrightarrow \quad 1 - \left|1 - \frac{2\theta}{\pi}\right| = 1 && (4.8)\\
&\Longleftrightarrow \quad \left|1 - \frac{2\theta}{\pi}\right| = 0 && (4.9)\\
&\Longleftrightarrow \quad \theta = \frac{\pi}{2}, && (4.10)
\end{aligned}$$

thus σ reaches its unique and strict *minimum* when $\theta = \frac{\pi}{2}$; in other words, using Corollary 1.2, for cyclic quadrilaterals.
Although squares and, more generally, rectangles, are cyclic, the converse is obviously untrue. In particular, the quadrilateral @, introduced in Example 1.1 and illustrated Figure 1, is considered as optimal by σ, and could even be flattened as much as desired, provided its four vertices remain cocyclic.

4.2 Asymptotic behavior

It is clear that σ strictly decreases (resp. increases) on $]0,\frac{\pi}{2}]$ (resp. $[\frac{\pi}{2},\pi[$); moreover,

$$\lim_{\theta\to 0^+}\frac{1}{1-\left|1-\frac{2\theta}{\pi}\right|}=\lim_{\theta\to\pi^-}\frac{1}{1-\left|1-\frac{2\theta}{\pi}\right|}=+\infty, \tag{4.11}$$

since $\sigma(\theta)=\frac{\pi}{2\theta}$ $\left(\text{resp. } \frac{\pi}{2(\pi-\theta)}\right)$ when $\theta\to 0^+$ (resp. $\theta\to\ \pi^-$).
Figure 5 represents the graph of skewness *vs.* torsion. The previously mentioned symmetry of σ results in a symmetric graph, and the fact that the absolute value is not $\mathcal{C}^1$ in 0 results in a salient point[6] when $\theta=\frac{\pi}{2}$.

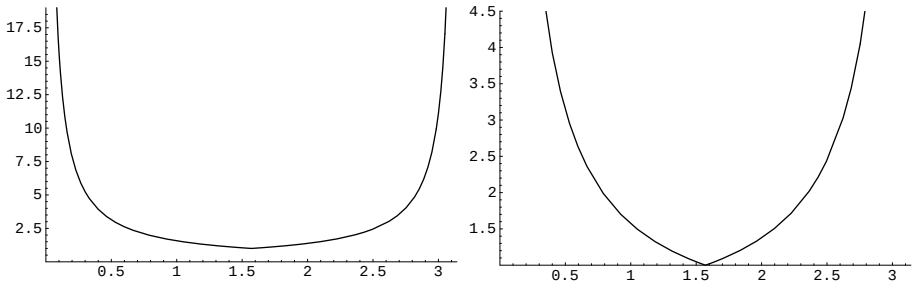


Figure 5: Graphs at two different scales of σ as a function of θ.

Rectangles are only a particular case of inscribed quadrilaterals; therefore, any rectangle is optimal, however stretched it is. Conversely, any non-rectangular parallelogram, *e.g.*, any non-square rhombus, is not optimal for σ. In fact, σ can be also seen as one-dimensional, but along a different direction of Q_1 than τ. In particular, triangular degeneracy is not caught by σ: if, say, $\alpha\to\pi$ then, according to Definition 2.1, γ and, consequently $\theta_1=\frac{\alpha+\gamma}{2}$, tend neither to 0 nor π; hence, σ does not diverge.

5 Edge to inradius

Among triangle qualities that extend naturally to quadraliterals is the comparison of edge lengths with inradius. Of course, any convex quadrilateral does not have, in general, an inscribed circle[7] and, hence, it might seem paradoxical to attempt to extend such edge to inradius comparisons to quadrilaterals. However, in the case of a non-degenerate triangle t, it follows from (1.3) that:

$$\frac{p_t}{r}=\frac{p_t^2}{\mathcal{A}_t}. \tag{5.1}$$

Therefore, this quality measure can be directly extended to q:

$$\zeta=\frac{p^2}{\mathcal{A}}. \tag{5.2}$$

[6] *i.e.*, left and right tangents at that point are not aligned.
[7] In fact, such an incircle exists if and only if $\alpha+\gamma=\beta+\delta$.

Similarily, the *aspect-ratio* can be extended to q:

$$\iota = \frac{p|q|_\infty}{\mathcal{A}}, \tag{5.3}$$

and these two measures are related *via* the following inequality:

$$\zeta = \frac{p(a+b+c+d)}{2\mathcal{A}} \leq \frac{4p|q|_\infty}{2\mathcal{A}} = 2\iota \tag{5.4}$$

with equality if and only if $p = 2|q|_\infty$, *i.e.* if and only if q is a rhombus.

5.1 *Extremum*

Combining (1.7) and (1.3) gives:

$$\zeta = \sqrt{\frac{p^4}{(p-a)(p-b)(p-c)(p-d) - abcd\cos^2\theta}} \tag{5.5}$$

and, as for triangles, it is much more convenient[8] to try to minimize $h = \frac{1}{\zeta^2}$ rather than to maximize ζ. Clearly, h is a function of only the five variables a, b, c, p and θ, and can be expressed as follows:

$$h(a,b,c,p,\theta) = \frac{(p-a)(p-b)(p-c)(a+b+c-p) - abc(2p-a-b-c)\cos^2\theta}{p^4} \tag{5.6}$$

and it is clear that, for any $a \in \mathbb{R}_+^*$, $h(\frac{a}{p}, \frac{b}{p}, \frac{c}{p}, 1, \theta) = h(a,b,c,p,\theta)$. This means that h only depends on four variables, given by three edges lengths to the semiperimeter ratios, plus the torsion angle θ. In other words, h is invariant through homothecy, as expected, thus ζ is non-dimensional. We therefore examine the variations of $\tilde{h} : \mathbb{Q}_1 \to \mathbb{R}$, where

$$\tilde{h}(x,y,z,\theta) = (1-x)(1-y)(1-z)(x+y+z-1) + xyz(x+y+z-2)\cos^2\theta \tag{5.7}$$

whose first order derivatives are:

$$\frac{\partial\tilde{h}}{\partial x}(x,y,z,\theta) = (2x+y+z-2)\left((1-y)(1-z) + yz\cos^2\theta\right) \tag{5.8}$$

$$\frac{\partial\tilde{h}}{\partial y}(x,y,z,\theta) = (x+2y+z-2)\left((1-x)(1-z) + xz\cos^2\theta\right) \tag{5.9}$$

$$\frac{\partial\tilde{h}}{\partial z}(x,y,z,\theta) = (x+y+2z-2)\left((1-x)(1-y) + xy\cos^2\theta\right) \tag{5.10}$$

$$\frac{\partial\tilde{h}}{\partial\theta}(x,y,z,\theta) = xyz(x+y+z-2)\sin 2t. \tag{5.11}$$

[8] and, obviously, equivalent.

According to (2.3), $x+y+z<2$, hence, since none of x, y nor z are null, the stationary-point condition implies that $2\theta \in \pi\mathbb{Z}$ thus, since $2\theta \in]0,\pi[$, necessarily $\theta = \frac{\pi}{2}$. Hence, $(x,y,z,\frac{\pi}{2})$ is a stationary-point if and only if

$$\left\{\begin{array}{lcl}(2x+y+z-2)(1-y)(1-z) &=& 0\\ (x+2y+z-2)(1-x)(1-z) &=& 0\\ (x+y+2z-2)(1-x)(1-y) &=& 0\end{array}\right. \tag{5.12}$$

which is equivalent, since neither x nor y nor z is equal to 1, to

$$\left\{\begin{array}{lcl}2x+y+z &=& 2\\ x+2y+z &=& 2\\ x+y+2z &=& 2\end{array}\right. \tag{5.13}$$

whose only solution is, clearly, $(\frac{1}{2},\frac{1}{2},\frac{1}{2})$. Therefore, $\tilde{h}$ has a unique stationary point, when the quadrilateral is a square. In order to check whether this case corresponds, as expected, to a *minimum*, one has to make sure that the hessian matrix is positive definite. The second order derivatives of $\tilde{h}$ are given by:

$$\frac{\partial^2\tilde{h}}{\partial x^2}(x,y,z,\theta) = 2(y+z-yz-1+yz\cos^2\theta) \tag{5.14}$$

$$\frac{\partial^2\tilde{h}}{\partial y^2}(x,y,z,\theta) = 2(x+z-xz-1+xz\cos^2\theta) \tag{5.15}$$

$$\frac{\partial^2\tilde{h}}{\partial z^2}(x,y,z,\theta) = 2(x+y-xy-1+yz\cos^2\theta) \tag{5.16}$$

$$\frac{\partial^2\tilde{h}}{\partial \theta^2}(x,y,z,\theta) = 2xyz(2-x-y-z)\cos 2t \tag{5.17}$$

$$\frac{\partial^2\tilde{h}}{\partial x\partial y}(x,y,z,\theta) = (1-z)(2x+2y+z-3)+z(2x+2y+z-2)\cos^2\theta \tag{5.18}$$

$$\frac{\partial^2\tilde{h}}{\partial x\partial z}(x,y,z,\theta) = (1-y)(2x+y+2z-3)+y(2x+y+2z-2)\cos^2\theta \tag{5.19}$$

$$\frac{\partial^2\tilde{h}}{\partial y\partial z}(x,y,z,\theta) = (1-x)(x+2y+2z-3)+x(x+2y+2z-2)\cos^2\theta \tag{5.20}$$

$$\frac{\partial^2\tilde{h}}{\partial x\partial \theta}(x,y,z,\theta) = yz(2-2x-y-z)\sin 2t \tag{5.21}$$

$$\frac{\partial^2\tilde{h}}{\partial y\partial \theta}(x,y,z,\theta) = xz(2-x-2y-z)\sin 2t \tag{5.22}$$

$$\frac{\partial^2\tilde{h}}{\partial z\partial \theta}(x,y,z,\theta) = xy(2-x-y-2z)\sin 2t \tag{5.23}$$

and, therefore, the hessian matrix of $\tilde{h}$ in $(\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{\pi}{2})$ is:

$$H_{\tilde{h}_\square} = -\frac{1}{8}\begin{pmatrix}4&2&2&0\\2&4&2&0\\2&2&4&0\\0&0&0&1\end{pmatrix} \tag{5.24}$$

and is clearly negative-definite; more precisely, the characteristic polynomial of the matrix

$$\begin{pmatrix} 4 & 2 & 2 & 0 \\ 2 & 4 & 2 & 0 \\ 2 & 2 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.25}$$

is $(X-1)(X-2)^2(X-8)$, thus the eigenvalues of $H_{\tilde{h}_\square}$ are $-\frac{1}{4}$ (double), $-\frac{1}{8}$ and -1. Hence, the stationary point is a *maximum*, meaning that ζ reaches its only *minimum* for squares; in addition, this *minimum*, denoted as $\zeta_\square$, is strict.
From this extremal property of ζ can be deduced another one for ι: firstly, since (5.4) is an equality only for rhombii, it is the case for squares, thus $\zeta_\square = 2\iota_\square$. Now, combining the minimization of ζ with (5.4), leads to:

$$\iota \geq \frac{\zeta}{2} \geq \frac{\zeta_\square}{2} = \iota_\square, \tag{5.26}$$

showing that, as ζ, ι is minimal for, and only for, squares.

5.2 Asymptotic behavior

5.2.1 Parallelogram

In this case, $\alpha = \gamma$ and $\beta = \delta$ thus $\theta = \alpha$ or $\theta = \beta$. Moreover, α and β are supplementary, hence have the same sine. Therefore, the area of q is given by:

$$\mathcal{A} = ab\sin\theta = |q|_0|q|_\infty \sin\theta, \tag{5.27}$$

thus ζ can be expressed as:

$$\zeta = \frac{(|q|_0 + |q|_\infty)^2}{|q|_0|q|_\infty \sin\theta} = \frac{1}{\sin\theta}\left(\tau + \frac{1}{\tau} + 2\right) \tag{5.28}$$

and ι as:

$$\iota = \frac{(|q|_0 + |q|_\infty)|q|_\infty}{|q|_0|q|_\infty \sin\theta} = \frac{1}{\sin\theta}(\tau + 1). \tag{5.29}$$

5.2.2 Rectangle

If q is a rectangle, then (5.28) and (5.29) respectively become $\zeta = \tau + \frac{1}{\tau} + 2$ and $\iota = \tau + 1$. Therefore, ζ and ι are two functions of τ, strictly increasing over $[1, +\infty[$, respectively denoted as $\zeta_\square$ and $\iota_\square$. Moreover,

$$\zeta_\square(\tau) \underset{\tau\to+\infty}{\sim} \tau \underset{\tau\to+\infty}{\sim} \iota_\square(\tau), \tag{5.30}$$

which means that they asymptotically behave as τ for rectangles. In particular, $\zeta_\square$ and $\iota_\square$ diverge to $+\infty$ when τ does.

5.2.3 Rhombus

If q is a rhombus, then it follows from (5.28) that:

$$\zeta = 2\iota = \frac{4}{\sin\theta} \tag{5.31}$$

which gives a direct relation between these quality measures and the torsion of the rhombus, since they reduce to two functions $\zeta_\diamond$ and $\iota_\diamond$, depending only on θ, with $\zeta_\diamond = 2\iota_\diamond$. Obviously, they both strictly increase (resp. decrease) on $]0, \frac{\pi}{2}]$ (resp. $[\frac{\pi}{2}, \pi[$). They also diverge to $+\infty$ at the bounds of $]0,\pi[$ since, using the fact that $\sin\theta = \sin(\pi-\theta)$,

$$\zeta_\diamond(\theta) = 2\iota_\diamond(\theta) \underset{\tau\to 0^+}{\sim} \frac{4}{\theta} \tag{5.32}$$

$$\zeta_\diamond(\theta) = 2\iota_\diamond(\theta) \underset{\tau\to \pi^-}{\sim} \frac{4}{\pi-\theta}. \tag{5.33}$$

The relative sensitivities of ζ to stretching and torsion are now discussed, by solving $\zeta_\square(\tau) = \zeta_\diamond(\theta)$, *i.e*,

$$\tau + \frac{1}{\tau} + 2 = \frac{4}{\sin\theta} \quad \Longleftrightarrow \quad \frac{\tau^2\sin\theta + 2(\sin\theta - 2)\tau + \sin\theta}{\tau\sin\theta} = 0, \tag{5.34}$$

which can be solved either for τ or θ. In the former case, (5.34) reduces to finding the real roots of the polynomial

$$P_\zeta = X^2\sin\theta + 2(\sin\theta - 2)X + \sin\theta \tag{5.35}$$

whose determinant is

$$\Delta_{P_\zeta} = 4\left((\sin\theta - 2)^2 - \sin^2\theta\right) \tag{5.36}$$

$$= 16(1-\sin\theta) \tag{5.37}$$

$$\geq 0 \tag{5.38}$$

with equality if and only if $\theta = \frac{\pi}{2}$. In this case, 1 is the only root of P_ζ, *i.e.*, q is a square. Otherwise, $0 < 1-\sin\theta < 1$, hence P_ζ has two distinct real roots; in addition, $1 - \sin\theta < \sqrt{1-\sin\theta}$, thus

$$2 - \sin\theta - 2\sqrt{1-\sin\theta} < \sin\theta < 2 - \sin\theta + 2\sqrt{1-\sin\theta}. \tag{5.39}$$

Combining (5.39) with the fact that, by definition, $\tau \geq 1$, it follows that, for any given $\theta \in]0,\pi[$, (8.18) has the following unique solution:

$$\tau = \frac{2 - \sin\theta + 2\sqrt{1-\sin\theta}}{\sin\theta} = \frac{(\sqrt{1-\sin\theta}+1)^2}{\sin\theta}. \tag{5.40}$$

Moreover, solving (5.34) for θ obviously brings two possible solutions, $\arcsin\frac{4\tau}{(\tau+1)^2}$ and $\pi - \arcsin\frac{4\tau}{(\tau+1)^2}$. In fact, they both correspond to the same shape for q, since they are supplementary.

5.2.4 Triangular degeneracy

In the case where q degenerates towards a triangle q, it is obvious that both perimeter and area of q tend towards their respective counterparts in t. Hence, $\zeta(q)$, in the sense of quadrilateral quality, tends towards $\zeta(t)$, in the sense of triangle quality. In particular, ζ does not diverge during triangular degeneracy.
From both extremal properties and asymptotic behavior that have been demonstrated, one can conclude that ζ can be extended as a generic quality measure, suitable for both triangles and quadrilaterals; in addition, a continuous transition between these two kinds of elements is ensured. However, this particular property might be, in some respects, a major drawback since, for some applications, one might wish to avoid triangular degeneracy of quadrilaterals. In this case, the quality measure should diverge to infinity, rather than tend towards the quality of the limiting triangular element.

6 Remembering diagonals

The following quadrilateral quality measure is proposed by [7]:

$$Q = \frac{|q|_2 \, h_{\max}}{\min_i \mathcal{A}_i} \tag{6.1}$$

where $\mathcal{A}_i$ denotes the area of the triangle whose edges are those of q adjacent to vertex i and $h_{\max} = \max(AC, BD, |q|_\infty)$. In particular, $|q|_\infty \leq h_{\max}$.

6.1 *Extremum*

An application of the CAUCHY-SCHWARZ inequality:

$$\left(\forall (u_1, \ldots, u_n) \in \mathbb{R}_+^n\right) \quad \sum_{k=1}^{k=n} u_k \leq \sqrt{n \sum_{k=1}^{k=n} u_k^2}, \tag{6.2}$$

to the edge lengths of q shows that

$$a + b + c + d \leq \sqrt{4\left(a^2 + b^2 + c^2 + d^2\right)} \tag{6.3}$$

thus $p \leq |q|_2$, with equality if and only if q is a rhombus. In addition,

$$\min_i \mathcal{A}_i \leq \frac{\mathcal{A}}{2} \tag{6.4}$$

with equality if and only if q is a parallelogram. Hence,

$$\frac{p}{\mathcal{A}} \leq \frac{|q|_2}{2 \min_i \mathcal{A}_i} \tag{6.5}$$

with equality if and only if q is a rhombus. Therefore, if q is not a rhombus,

$$\iota = \frac{p|q|_\infty}{\mathcal{A}} < \frac{|q|_2 |q|_\infty}{2 \min_i \mathcal{A}_i} \leq \frac{|q|_2 \, h_{\max}}{2 \min_i \mathcal{A}_i} = \frac{Q}{2}. \tag{6.6}$$

$$\frac{\zeta}{2} \leq \iota = \frac{p|q|_\infty}{\mathcal{A}} \leq \frac{|q|_2 h_{\max}}{2\min_i \mathcal{A}_i} = \frac{Q}{2}. \tag{6.7}$$

and, in particular, $\zeta \leq Q$.

6.2 Asymptotic behavior

6.2.1 Parallelogram

If q is a parallelogram, then

$$(\forall i \in \{A,B,C,D\}) \quad \min_i \mathcal{A}_i = \frac{\mathcal{A}}{2} = \frac{|q|_0|q|_\infty \sin\theta}{2} \tag{6.8}$$

and, using AL KASHI's Theorem, it is obvious that

$$BD = \sqrt{a^2+b^2-2ab\cos\alpha} \tag{6.9}$$

$$AC = \sqrt{a^2+b^2-2ab\cos\beta} = \sqrt{a^2+b^2+2ab\cos\alpha} \tag{6.10}$$

since α and β are supplementary. Hence,

$$\max(AC,BD) = \sqrt{a^2+b^2+2ab\,|\cos\alpha|} > \max(a,b) = |q|_\infty \tag{6.11}$$

thus

$$h_{\max} = \sqrt{a^2+b^2+2ab\,|\cos\alpha|} = \sqrt{|q|_0^2+|q|_\infty^2+2|q|_0|q|_\infty|\cos\theta|}. \tag{6.12}$$

Finally,

$$|q|_2 = \sqrt{2(a^2+b^2)} = \sqrt{2(|q|_0^2+|q|_\infty^2)} \tag{6.13}$$

therefore

$$Q = \frac{2\sqrt{2(|q|_0^2+|q|_\infty^2)(|q|_0^2+|q|_\infty^2+2|q|_0|q|_\infty|\cos\theta|)}}{|q|_0|q|_\infty\sin\theta} \tag{6.14}$$

$$= \frac{2\sqrt{2(1+\tau^2)(1+\tau^2+2\tau|\cos\theta|)}}{\tau\sin\theta}. \tag{6.15}$$

6.2.2 Rectangle

In this case, (6.15) shows that Q becomes the following function of the stretching factor:

$$Q_{\square}(\tau) = \frac{2\sqrt{2(1+\tau^2)(1+\tau^2)}}{\tau} = 2\sqrt{2}\left(\tau+\frac{1}{\tau}\right). \tag{6.16}$$

Clearly, $Q_{\square}$ strictly increases over $[1,+\infty[$ and

$$Q_{\square}(\tau) \underset{\tau\to+\infty}{\sim} 2\sqrt{2}\tau \tag{6.17}$$

hence, in particular, $\lim_{\tau\to+\infty} Q_{\square}(\tau) = +\infty$.

6.2.3 Rhombus

In the case where q is a rhombus, then it follows from (6.15) that Q can be simplified in a function of the sole torsion:

$$Q_\diamond(\theta) = \frac{4\sqrt{2(1+|\cos\theta|)}}{\sin\theta} \tag{6.18}$$

which diverges to $+\infty$ in both 0^+ or π^-. More precisely,

$$Q_\diamond(\theta) \underset{\tau\to 0^+}{\sim} \frac{8}{\theta} \tag{6.19}$$

$$Q_\diamond(\theta) \underset{\tau\to \pi^-}{\sim} \frac{8}{\pi-\theta}. \tag{6.20}$$

6.2.4 Triangular degeneracy

In case of a triangular degeneracy of q towards a triangle t,

$$(\exists i \in \{A,B,C,D\}) \quad \lim_{q\to t} \mathcal{A}_i = 0 \tag{6.21}$$

which implies in particular that $\min_i \mathcal{A}_i$ tends to 0, while neither $|q|_2$ nor $h_{\max}$ do; thus, by definition, Q diverges to $+\infty$. In other words, Q detects triangular degeneracies, while neither ζ nor ι do. Consequently, Q is not continuous with any underlying triangle quality measure, unlike ζ and ι.

Remark 6.1. Clearly, this behavior is due to the fact that the denominator of Q is no longer the entire area of the quadrilateral, but the minimal triangular area. Variations of ζ and ι can be designed consequently, if the goal is to detect triangular degeneracy.

An example of how ζ, ι and Q behave in case of triangular degeneracy is provided by the following example:

Example 6.1. Let $\diamondsuit_x$ denote the kite such that ABC is a unitary equilateral triangle, while ACD is iscosceles in D, with $AD = CD = x$, as illustrated Figure 6. Obviously,

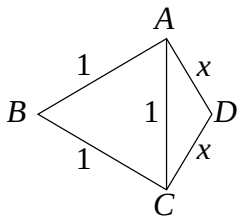


Figure 6: Kite $\diamondsuit_x$.

x must belong to $\left]\frac{1}{2},+\infty\right[$ but, since the aim is here to examine the case of a triangular degeneracy, the interval is limited to $\left]\frac{1}{2},1\right[$. It is straightforward to determine ζ, ι and

$\mathcal{Q}$ as function of x when $q = \diamondsuit_x$. In fact,

$$\begin{aligned}
\min_i \mathcal{A}_i(x) &= \frac{1}{4}\sqrt{x^2 - \frac{1}{4}} && (6.22)\\
p(x) &= 1 + x && (6.23)\\
|q|_2(x) &= \sqrt{2(1+x^2)} && (6.24)\\
|q|_\infty(x) &= 1 && (6.25)\\
\mathcal{A}(x) &= \frac{\sqrt{3}}{4} + \frac{1}{4}\sqrt{x^2 - \frac{1}{4}}. && (6.26)
\end{aligned}$$

Hence, when $x \to \frac{1}{2}$, both ζ and ι tend towards finite values (respectively, $3\sqrt{3}$ and $2\sqrt{3}$), while $\mathcal{Q} \to +\infty$.

7 Adaptation of κ_2 to right isosceles reference triangles

An interesting approach to estimate triangle quality has been proposed by various authors (*cf.* [2, 6, 9]), based on the singular values of a matrix expressing the affine transformation between the mesh element and a given reference element. More precisely, these works have focused on the case where the reference element is a regular simplex, since this element is generally considered to be the best possible for isotropic simplicial meshes. An in-depth examination of the variations of such quality measures, as well as a comparison with other quality measures has been made in [11].
It seems that the full derivation, in the case where the reference element is a right isosceles triangle, appears for the first time in [10]. One reason is that such elements do not generally correspond to the kind of triangles that are desired in the context of finite element analysis. However, in the goal of extending this measure to quadrilateral meshes, such elements become naturally the desired ones. Since the goal of the present article is to provide a comprehensive analysis of quadrangle quality measures, this derivation is recalled here.

7.1 Construction

As described in [2], the *edge-matrix* of a triangle t is defined by:

$$T_0 = \begin{pmatrix} x_F - x_E & x_G - x_E \\ y_F - y_E & y_G - y_E \end{pmatrix} \tag{7.1}$$

and let W be the edge-matrix of a reference isosceles right triangle, for example

$$W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{7.2}$$

meaning that W is, simply, the identity matrix of $\mathbb{R}^2$. Hence, $T_0 W^{-1} = T_0$ is the matrix that maps the reference element into t. Using the same ideas as in [2], it is possible

to define matrix-norms based on the singular values σ of T_0. Obviously, the symmetry which arises when W is an equilateral triangle vanishes with this new reference element. In particular, t is considered as being optimal only if the right angle is in A. This property allows a strict control, not only over the shape of t, but also on the vertex at which the right angle should be. The singular values are given by the positive square-roots of the eigenvalues of the positive definite matrix $T_0^T T_0$. Now,

$$T_0^T T_0 = \begin{pmatrix} g^2 & \overrightarrow{EF} \cdot \overrightarrow{EG} \\ \overrightarrow{EF} \cdot \overrightarrow{EG} & f^2 \end{pmatrix}, \tag{7.3}$$

where $\cdot$ denotes the usual scalar product. The singular values σ of T_0 are thus obtained from the characteristic equation of $T_0^T T_0$ as

$$\sigma^4 - (f^2+g^2)\sigma^2 + f^2 g^2 - \left(\overrightarrow{EF} \cdot \overrightarrow{EG}\right)^2 = 0 \tag{7.4}$$

or, equivalently,

$$\sigma^4 - (f^2+g^2)\sigma^2 + 4\mathcal{A}_t^{\,2} = 0. \tag{7.5}$$

Hence,

$$\sigma_1^2 + \sigma_2^2 = f^2 + g^2 \tag{7.6}$$

and $\sigma_1\sigma_2 = 2\mathcal{A}_t$ where σ_1^2 and σ_2^2 ($0 < \sigma_1 \leq \sigma_2$) are the two roots of (7.5). A quality measure can be constructed from the condition number of any unitarily invariant norm of the matrix T_0 (*cf.* [2]). One such family is derived from the SCHATTEN p-norms defined by:

$$N_p(T_0) = \left(\sigma_1^p + \sigma_2^p\right)^{1/p}, \quad p \in [1, +\infty[. \tag{7.7}$$

The case $p = 2$ is the FROBENIUS norm, the limiting case $p \to \infty$ is the spectral norm and the case $p = 1$ is the trace norm. A non-normalized quality measure is given by the condition number $\kappa_p(T_0)$ which is defined as

$$\kappa_p(T_0) = \left[\left(\sigma_1^p + \sigma_2^p\right)\left(\sigma_1^{-p} + \sigma_2^{-p}\right)\right]^{1/p}. \tag{7.8}$$

In the particular case $p = 2$, using (1.2), it follows that

$$\kappa_2(T_0) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1\sigma_2} = \frac{f^2+g^2}{2\mathcal{A}_t} = \frac{f^2+g^2}{fg\sin\eta} \tag{7.9}$$

thus $\kappa_2(T_0)$ depends only on metric and angular parameters of t, and therefore can be denoted unambiguously as a function of t. In order to avoid the confusion with the "classic" κ_2 triangle quality measure, with equilateral reference element, a slighty different notation shall be used, *e.g.*,

$$\kappa_2^{\perp}(t) = \kappa_2(T_0). \tag{7.10}$$

Now, assuming that $\xi = \frac{f}{g}$, which is allowed since t is non degenerate and thus $g \neq 0$, it then follows that

$$\kappa_2^{\perp}(t) = \frac{\xi^2+1}{\xi\sin\eta} = \left(\xi + \frac{1}{\xi}\right)\frac{1}{\sin\eta} \tag{7.11}$$

or, in entirely angular terms, since (1.2) shows that $\xi = \frac{\sin\phi}{\sin\psi}$,

$$\kappa_2^{\perp}(t) = \left(\frac{\sin\phi}{\sin\psi} + \frac{\sin\psi}{\sin\phi}\right)\frac{1}{\sin\eta} = \frac{\sin^2\phi + \sin^2\psi}{\sin\eta\sin\phi\sin\psi}. \tag{7.12}$$

Remark 7.1. As demonstrated in [11], the following angular identity arises in the context of equilateral reference elements:

$$\kappa_2(t) = \frac{\sin^2\eta + \sin^2\phi + \sin^2\psi}{\sin\eta\sin\phi\sin\psi} \tag{7.13}$$

and is invariant through angle permutation[9], unlike (7.12). This is intuitively clear, since equilateral triangles are $\mathfrak{S}_3$-invariant, while right isosceles triangles are not. Therefore, since η, ϕ and ψ are supplementary, the surface $z = \kappa_2(t)$ can be equivalently represented as a function of any angle pair of t, while this is no longer true for $z = \kappa_2^{\perp}(t)$, for which at least one angle must be specifically chosen. In particular, if one angle is chosen to be η (the one which is right, ideally, for $\kappa_2^{\square}$), then the second is arbitrary and can be, *e.g.*, ϕ. In this case, it follows from (7.12) and (7.13), respectively,

$$\kappa_2^{\perp}(t) = \frac{\sin^2\phi + \sin^2(\eta+\phi)}{\sin\eta\sin\phi\sin(\eta+\phi)} \tag{7.14}$$

and

$$\kappa_2(t) = \frac{\sin^2\eta + \sin^2\phi + \sin^2(\eta+\phi)}{\sin\eta\sin\phi\sin(\eta+\phi)}. \tag{7.15}$$

7.2 *Extremum*

According to (7.11), consider the mapping

$$\begin{array}{rccc} k: & \mathbb{R}_+^* \times]0,\pi[& \longrightarrow & \mathbb{R}_+^* \\ & (\xi,\eta) & \longmapsto & \left(\xi + \frac{1}{\xi}\right)\frac{1}{\sin\eta} \end{array} \tag{7.16}$$

which is $\mathcal{C}^\infty$ over the open domain $\mathbb{R}_+^* \times]0,\pi[$; hence, any local *extremum* of k is attained at a stationary point. The first order derivatives are:

$$\frac{\partial k}{\partial \xi}(\xi,\eta) = \left(1 - \frac{1}{\xi^2}\right)\frac{1}{\sin\eta} \tag{7.17}$$

$$\frac{\partial k}{\partial \eta}(\xi,\eta) = -\left(\xi + \frac{1}{\xi}\right)\frac{\cos\eta}{\sin^2\eta} \tag{7.18}$$

and, given the definition domain, the only stationary point is $(1, \frac{\pi}{2})$. Again, the nature of this point can be examined by means of the hessian matrix of k, assembled with the

[9] or, in less rigorous but simpler terms, symmetrical.

second order derivatives of k:

$$\frac{\partial^2 k}{\partial \xi^2}(\xi,\eta) = \frac{2}{\xi^3 \sin\eta} \tag{7.19}$$

$$\frac{\partial^2 k}{\partial \eta^2}(\xi,\eta) = \left(\xi + \frac{1}{\xi}\right) \frac{\sin^2\eta + 2\cos^2\eta}{\sin^3\eta} \tag{7.20}$$

$$\frac{\partial^2 k}{\partial \xi \partial \eta}(\xi,\eta) = -\left(1 - \frac{1}{\xi^2}\right) \frac{\cos\eta}{\sin^2\eta} \tag{7.21}$$

which gives, when $(\xi,\eta) = (1,\frac{\pi}{2})$,

$$\frac{\partial^2 k}{\partial \xi^2}\left(1,\frac{\pi}{2}\right) = \frac{\partial^2 k}{\partial \eta^2}\left(1,\frac{\pi}{2}\right) = 2 \tag{7.22}$$

$$\frac{\partial^2 k}{\partial \xi \partial \eta}\left(1,\frac{\pi}{2}\right) = 0. \tag{7.23}$$

Thus, the hessian determinant is equal to $4 > 0$ and the first diagonal entry is $2 > 0$. Hence, the hessian matrix is locally positive definite around the critical point, which therefore corresponds to a strict local *minimum* of k. Since k is $\mathcal{C}^\infty$ over its open and connected definition domain, the uniqueness of the critical point ensures that this *minimum* is, also, absolute. In other words, $\kappa_2^\perp(t)$ is minimal only for right ($\eta = \frac{\pi}{2}$) isosceles ($\xi = 1 \Leftrightarrow f = g$) triangles. In this case, the value of $\kappa_2^\perp(t)$ is, obviously 2, which provides the normalization coefficient.

Remark 7.2. If t is equilateral, then $\kappa_2^\perp(t) = \frac{4}{\sqrt{3}} \gtrapprox 2.31 > 2$, as expected since the reference element is no longer equilateral.

7.3 Asymptotic behavior

A needle degeneracy occurs when one, and only one, of the angles of t tends to 0 (*cf.* [11]). This implies none of them tends to π, otherwise there would be two angles tending to 0. Hence, one and only one of the sines in (7.12) tends to 0, implying that the numerator tends to a non-zero value, while the denominator tends to 0. Therefore, $\kappa_2^\perp(t) \to +\infty$.
In the case where t flattens, *i.e.*, one of its angles tends to π, either $\eta \to \pi$ (flattens in E), or $\eta \to 0$ (flattens in F or G). In both cases, $\sin\eta \to 0$ thus, combined with the fact that $\xi + \frac{1}{\xi} \geq 2$, it follows from (7.9) that $\kappa_2^\perp(t) \to +\infty$.

Remark 7.3. In the case where t is a right triangle, but not necessarily right isosceles in E, it follows from (7.11) that:

- if t is right in E, then $\kappa_2^\perp(t) = \frac{f^2+g^2}{fg}$, which equals 2 if and only if $f = g$, and tends to $+\infty$ as either $\frac{f}{g}$ or $\frac{g}{f}$ does (needle degeneracy);
- if t is right in F, then $\kappa_2^\perp(t) = \frac{e^2+2g^2}{eg}$, which at best equals $2\sqrt{2}$, when $e = g\sqrt{2}$, and tends to $+\infty$ as either $\frac{e}{g}$ or $\frac{g}{e}$ does (needle degeneracy). In particular, if t is also isosceles, then $\kappa_2^\perp(t) = 3$. For symmetry reasons, the results are obviously the same if t is right in G.

Figure 7 provides a graphic comparison between the respective behaviors of κ_2 (*cf.* [11] for details) and $\kappa_2^\perp$. In fact, three different $z = \kappa_2^\perp$ surfaces can be defined, depending on which couple of angular variables is picked, because of the asymmetry explained in Remark 7.1. Therefore, at least concerning $\kappa_2^\perp$, either an η-axis must be specified, or neither of the two axes[10] actually are associated to η. In Figure 7, η is in abscissa and, as expected, the two surfaces have different bottoms: $(\frac{\pi}{2}, \frac{\pi}{4}, 1)$ for $z = \kappa_2^\perp$ and $(\frac{\pi}{3}, \frac{\pi}{3}, 1)$ for $z = \kappa_2$.

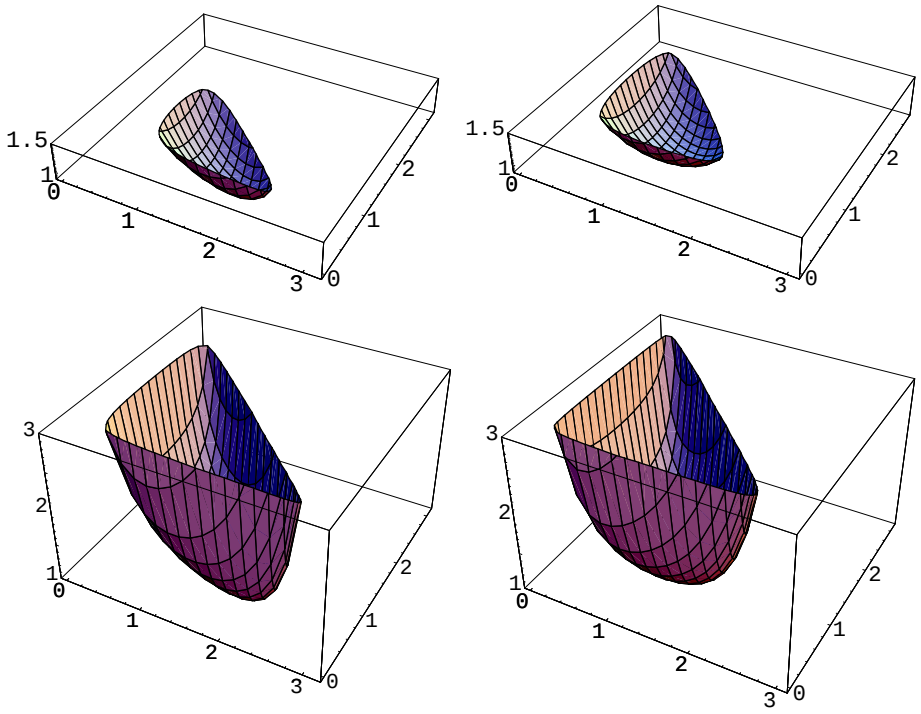


Figure 7: Surfaces $z = \frac{\kappa_2^\perp}{2}$ (left) and $z = \frac{\kappa_2}{2}$ (right) as functions of η (abscissa) and either ϕ or ψ (ordinate), for $z < 1.5$ (up) and $z < 3$ (down).

8 Extending κ_2 to quadrilaterals

Section 7 provides a matrix-based triangle quality measure to the case where the reference element is a right isosceles triangle with, in addition, a specific control over which edge is the hypotenuse. The main motivation of this modification is to allow, in a second step, the adaptation of κ_2 to quadrilaterals.

[10] in fact, in this case, η is implicitly represented along the $x + y = \pi$ axis.

8.1 Construction

Considering the generic planar quadrilateral q, four different triangles might be evaluated by means of $\kappa_2^\perp$: $t_A = DAB$, $t_B = ABC$, $t_C = BCD$ and $t_D = CDA$, with respective edge-matrices T_A, T_B, T_C and T_D. Now, it follows from (7.1) that

$$T_A + T_B + T_C + T_D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{8.1}$$

i.e.,

$$T_3 = -T_0 - T_1 - T_2. \tag{8.2}$$

In other words, it is not necessary to evaluate the four edge-matrices at each vertex of the quadrilateral, since any of them is a linear combination of the three other ones. This simply means that, given three vertex angles and edge ratios, the quadrilateral is fully determined, up to homothecy.
Now, considering $\kappa_2^\perp(t_i)$, as it has been previously modified for right isosceles triangles, the qualities of each of these four triangles are, respectively,

$$\kappa_2^\perp(t_A) = \frac{a^2+d^2}{ad\sin\alpha}, \quad \kappa_2^\perp(t_B) = \frac{a^2+b^2}{ab\sin\beta}, \tag{8.3}$$

$$\kappa_2^\perp(t_C) = \frac{b^2+c^2}{bc\sin\gamma}, \quad \kappa_2^\perp(t_D) = \frac{c^2+d^2}{cd\sin\delta}. \tag{8.4}$$

According to (8.1), $\kappa_2^\perp(t_D)$ is related to $\kappa_2^\perp(t_A)$, $\kappa_2^\perp(t_B)$ and $\kappa_2^\perp(t_C)$, but this dependency is no longer linear, since singular values and, hence, polynomial equations, are involved. Therefore, although it might appear as more elegant to design a quadrilateral quality measure, depending only on three of the underlying triangle qualities, it is certainly much more costly. For this reason, a more realistic and certainly more efficient idea is to take into account the four qualities. For instance, one might take the worst of these qualities, in other words their max norm:

$$\kappa_2^\infty(q) = \max\left(\kappa_2^\perp(t_A), \kappa_2^\perp(t_B), \kappa_2^\perp(t_C), \kappa_2^\perp(t_D)\right). \tag{8.5}$$

Remark 8.1. As always, the problem with the max norm is that, provided the worst triangle remains the same, it is intrisically unable to take into account quality variations of the three other ones, as illustrated by Figure 8: κ_2^∞ cannot detect the fact than a quadrilateral is "less" distorted than the other.

A natural attempt to address the case mentioned in Remark 8.1 is to consider the arithmetic mean instead of the max norm:

$$\kappa_2^1(q) = \frac{\kappa_2^\perp(t_A) + \kappa_2^\perp(t_B) + \kappa_2^\perp(t_C) + \kappa_2^\perp(t_D)}{4}. \tag{8.6}$$

Remark 8.2. The choice of the arithmetic mean is arbitrary without any further justification. One might, *e.g.*, prefer to use the euclidean norm instead.

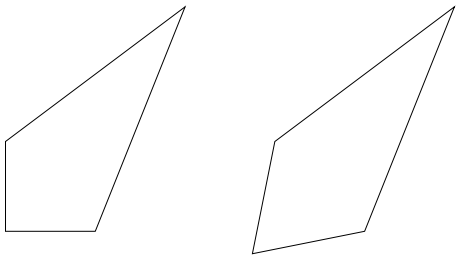


Figure 8: Both quadrilaterals share the same κ_2^∞.

8.2 *Extremum*

It has been proved in Section 7 that

$$(\forall\, i \in \{A,B,C,D\}) \quad \kappa_2^\perp(t_i) \geq 2 \tag{8.7}$$

with equality if and only if t_i is a right isosceles triangle. Hence, by definition,

$$\kappa_2^1(q) \geq \frac{2+2+2+2}{4} = 2 \tag{8.8}$$

with equality if and only if all $\kappa_2^\perp(t_i)$ are equal to 2. In addition,

$$\kappa_2^\infty(q) = 2 \iff (\forall\, i \in \{A,B,C,D\}) \quad \kappa_2^\perp(t_i) = 2 \tag{8.9}$$

whence both κ_2^∞ and κ_2^1 reach their unique absolute *minimum* only when q is a square. Moreover, they have the same *minimum*, 2, which provides the normalization coefficient.

8.3 Asymptotic behavior

8.3.1 Parallelogram

As explained in Paragraph 5.2, when q is parallelogram it follows that $\sin\alpha = \sin\beta = \sin\theta$; in addition, $a = c$ and $b = d$, whence

$$(\forall\, i \in \{A,B,C,D\}) \quad \kappa_2^\perp(t_i) = \frac{a^2+b^2}{ab\sin\theta} \tag{8.10}$$

from which it follows that

$$\kappa_2^\infty(q) = \kappa_2^1(q) = \frac{a^2+b^2}{ab\sin\theta} = \frac{|q|_2^2}{2\mathcal{A}} \tag{8.11}$$

or, equivalently,

$$\kappa_2^\infty(q) = \kappa_2^1(q) = \frac{|q|_0^2+|q|_\infty^2}{|q|_0|q|_\infty\sin\theta} \tag{8.12}$$

from which the following expression in terms of stretching and torsion of κ_2^1 for parallelograms arises:

$$\kappa_2^\infty(q) = \kappa_2^1(q) = \frac{1}{\sin\theta}\left(\tau + \frac{1}{\tau}\right). \tag{8.13}$$

Remark 8.3. In the context of the κ_2 triangle quality measure, with an equilateral reference element, it is shown in [11] that the following identity holds:

$$\kappa_2(t) = \frac{|t|_2^2}{2\sqrt{3}\mathcal{A}_t}, \tag{8.14}$$

which appears to be similar to (8.11), up to a constant factor. It is quite satisfactory to obtain the same result, up to a constant factor, for triangles[11] and parallelograms.

8.3.2 Rectangle

If q is a rectangle, then (8.13) can obviously be simplified as follows:

$$\kappa_2^\infty(q) = \kappa_2^1(q) = \tau + \frac{1}{\tau} \tag{8.15}$$

thus κ_2^∞ and κ_2^1 reduce to the same function, denoted as $\kappa_{2\square}$, which depends only on the stretching factor of q, and is obviously strictly increasing over $[1, +\infty[$. In addition, its asymptotic behavior is given by

$$\kappa_{2\square}(\tau) \underset{\tau\to+\infty}{\sim} \tau \underset{\tau\to+\infty}{\sim} \zeta_{\square}(\tau), \tag{8.16}$$

which implies, in particular, that $\kappa_{2\square}$ diverges towards $+\infty$ when τ tends to $+\infty$.

8.3.3 Rhombus

It is also interesting to notice that, in the case where q is a rhombus, (8.13) show that both κ_2^∞ and κ_2^1 reduce to a function of θ:

$$\kappa_2^\infty(q) = \kappa_2^1(q) = \kappa_{2\diamond}(\theta) = \frac{2}{\sin\theta} \tag{8.17}$$

which strictly decreases (resp. decreases) on $]0, \frac{\pi}{2}]$ (resp. $[\frac{\pi}{2}, \pi[$) and is equivalent to $\frac{2}{\theta}$ (resp. $\frac{2}{\pi-\theta}$) when $\theta \to 0^+$ (resp. $\theta \to \pi^-$). In particular, $\kappa_{2\diamond}$ diverges to $+\infty$ when θ tends to either 0^+ or π^-.
The relative sensitivities of κ_2^∞ and κ_2^1 to stretching and torsion are now discussed, by solving $\kappa_{2\square}(\tau) = \kappa_{2\diamond}(\theta)$, *i.e*,

$$\tau + \frac{1}{\tau} = \frac{2}{\sin\theta} \quad \Longleftrightarrow \quad \frac{\tau^2\sin\theta - 2\tau + \sin\theta}{\tau\sin\theta} = 0 \tag{8.18}$$

and solving for τ reduces to finding the roots of the polynomial $X^2\sin\theta - 2X + \sin\theta$. Obviously, these roots are $\frac{1\pm|\cos\theta|}{\sin\theta}$, corresponding to a double root only when $\theta = \frac{\pi}{2}$

[11] in the usual case, when the reference element is equilateral.

and, hence, $\tau = 1$, *i.e.*, if and only if q is a square. Conversely, $\tau = 1$ implies $\sin\theta = 1$ thus $\theta = \frac{\pi}{2}$. In the case where $\theta \neq \frac{\pi}{2}$, it is clear that $1 + |\cos\theta| > \sin\theta > 0$, hence $\frac{1+|\cos\theta|}{\sin\theta}$ complies with the fact that $\tau > 1$. Now, $\theta \in]0,\pi[\setminus\{\frac{\pi}{2}\}$ implies $0 < \sin\theta < 1$ thus $\sin^2\theta < \sin\theta$ and

$$1 - |\cos\theta| < 1 - \cos^2\theta = \sin^2\theta < \sin\theta. \tag{8.19}$$

Therefore, $\frac{1+|\cos\theta|}{\sin\theta} < 1$, hence for any $\theta \in\ ]0,\pi[$, (8.18) has the following unique solution:

$$\tau = \frac{1+|\cos\theta|}{\sin\theta}. \tag{8.20}$$

Moreover, solving (8.18) for θ obviously brings two possible solutions, $\arcsin\frac{2\tau}{\tau^2+1}$ and $\pi - \arcsin\frac{2\tau}{\tau^2+1}$. In fact, they both correspond to the same shape for q, since they are supplementary.
Figure 9 represents the respective stretching factor *vs.* torsion sensitivities of ζ and κ_2^1, using (5.40) and (8.20). Both graphs have a salient point in $(\frac{\pi}{2},1)$, since neither the absolute value (for ζ) nor the square root (for κ_2^1) functions are $\mathcal{C}^1$ in 0.

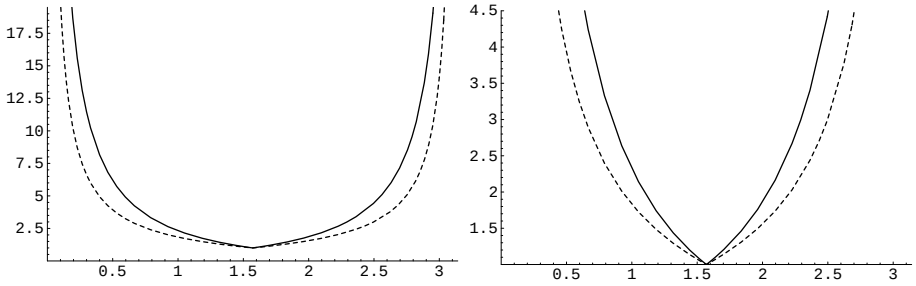


Figure 9: Graphs at two different scales providing the respective sensitivities of ζ (dotted) and κ_2^1 (solid) to stretching *vs.* torsion.

8.3.4 Triangular degeneracy

In case of a triangular degeneracy of q, *i.e.*, at least one of its angles tends to π, it follows immediately from (8.3) and (8.4) that at least one of the $\kappa_2^\perp(t_i)$ tends to $+\infty$, as do $\kappa_2^\infty(q)$ and $\kappa_2^1(q)$. In other words, both κ_2^∞ and κ_2^1 diverge in case of triangular degeneracy.

Remark 8.4. Another quality measure could be defined by generalizing (8.11) to all quadrilaterals. In the case of a triangular degeneracy, this measure would not diverge because the denominator (the area) would not tend to 0, while the numerator (the sum of all squared edge lengths) remains bounded. Moreover, it would not be continuous with the κ_2 triangle quality measure, even after adjusting normalization coefficients, since $\mathcal{A} \to \mathcal{A}_t$ while $|q|_2^2 \not\to |t|_2^2$. This makes a clear difference with ζ, for which triangular continuity is ensured by the fact that $p \to p_t$.

9 Numerical results

Because of the four-dimensionality of Q_1, it is impossible to provide comprehensive plottings of quadrangle quality measures, *e.g.* as in [11]. The alternate approach used here is to test these measures over a representative set of quadrilaterals. In addition, τ, σ, ζ, ι, Q, κ_2^1 and κ_2^∞ are tested on various quadrilateral meshes.

9.1 Reference quadrilaterals

The proposed set of reference elements consists in the previously introduced quadrilaterals □, ⓠ, ◊ and ◇$_x$(with $x = 0.51$ and $x = 0.501$), along with:

- and , rectangles with respective stretching factors of 2 and 100;
- q_1 and q_2, denoting the two quadrilaterals of Figure 8;
- ◊, a rhombus such as $AB = 5AC$.

	τ	σ	$\frac{1}{4}\zeta$	$\frac{1}{2}\iota$	$\frac{1}{4\sqrt{2}}Q$	$\frac{1}{2}\kappa_2^1$	$\frac{1}{2}\kappa_2^\infty$
□	1	1	1	1	1	1	1
	2	1	1.125	1.5	1.25	1.25	1.25
	100	1	25.5	50.5	50	50.01	50.01
◊	1	1.5	1.155	1.155	1.414	1.155	1.155
◊	1	7.841	5.025	5.025	7.071	5.025	5.025
◇$_{0.51}$	1.961	1.261	1.18	1.562	5.585	1.588	2.588
◇$_{0.501}$	1.996	1.415	1.255	1.672	17.68	2.969	7.933
ⓠ	7.4	1	1.885	3.389	33.9	2.957	3.604
q_1	2.693	1.484	1.437	2.152	4.456	1.598	1.929
q_2	2.112	1.824	1.498	2.085	3.428	1.602	1.929

Table 1: Qualities of a representative set of quadrilaterals.

The corresponding normalized quality measures are given Table 1, from which several remarks can be made:

– as mentioned before, neither τ nor σ make sense as quality measures in a general context; however, when considered together, they provide valuable information about quadrilateral shape along two "dimensions" of Q_1;

– as expected, neither ζ nor ι detect the triangular degeneracies which appear in ◇$_{0.51}$ and, more markedly, in ◇$_{0.501}$. The fact that, nevertheless, ι grants worse

qualities to these quadrangles than ζ does is due to the "unsmoothing effect" implied by the presence of $|q|_\infty$ in the former;

– as expected, κ_2^1 and κ_2^∞ are disconnected as soon as asymmetry appears. In this case, κ_2^∞ is more sensitive to distortion than κ_2^1. Moreover, although κ_2^1 is supposed to be more discriminant than κ_2^∞ when the worst part[12] of q is invariant, this difference is not significant between q_1 and q_2;

– $\mathcal{Q}$ is extremely sensible, which is not a problem *per se*, but in this case one can wonder whether it is justified to obtain, roughly, $\mathcal{Q}(\textcircled{@}) \approx 10\mathcal{Q}(q_2)$. In addition, $\mathcal{Q}(q_1)$ is significantly worse than $\mathcal{Q}(q_2)$ as opposed to, *e.g.*, the results obtained with κ_2-based quality measures. Depending on application, one might decide which of q_1 or q_2 should be considered as best, but $\mathcal{Q}$ seems to be slightly unbalanced overall

9.2 Quadrilateral meshes

An other perspective on mesh quality measures is to examine how they behave on actual meshes. In this goal, it will be made use of six different quadrangular meshes, denoted as L_1 to L_6, of a given planar domain, as illustrated by Figure 10. These meshes differ in regularity, density an isotropy, and these variations should appear in terms of mesh quality. There are numerous ways to evaluate the quality of an entire mesh from that of its elements; discussing them is beyond the scope of this article. The most natural approach consists in considering the range[13] and the arithmetic mean of element quality measures across the entire mesh. The corresponding results are given in Table 2. From these numerous results, it appears that:

– clearly, $\mathcal{Q}$ and κ_2^∞ are much more discriminant than the other quality estimates. For example, both L_1 and L_4 contain very distorted elements (as confirmed by worst torsion and skewness) and the respective *maxima* of $\mathcal{Q}$ and κ_2^∞ across these meshes are much greater than for the other quality measures. In fact, these specific examples exhibit a strong correlation between $\mathcal{Q}$ and κ_2^∞ which is not a general rule, as exhibited by Table 1;

– the unbalanced behavior of $\mathcal{Q}$, suspected from the results of Table 1, is not confirmed here;

– on the contrary, κ_2^1 and κ_2^∞ are largely disconnected: the trend observed in Table 1 for some configurations is confirmed across entire meshes;

– L_3 is considered as very good mesh by ζ but not by $\mathcal{Q}$ nor κ_2^∞. A close visual examination of this mesh shows that only triangular degeneracies can be observed, explaining why ζ does not detect them.

[12] in the sense of the κ_2 triangle quality measure.
[13] *i.e.*, the best and the worst element qualities.

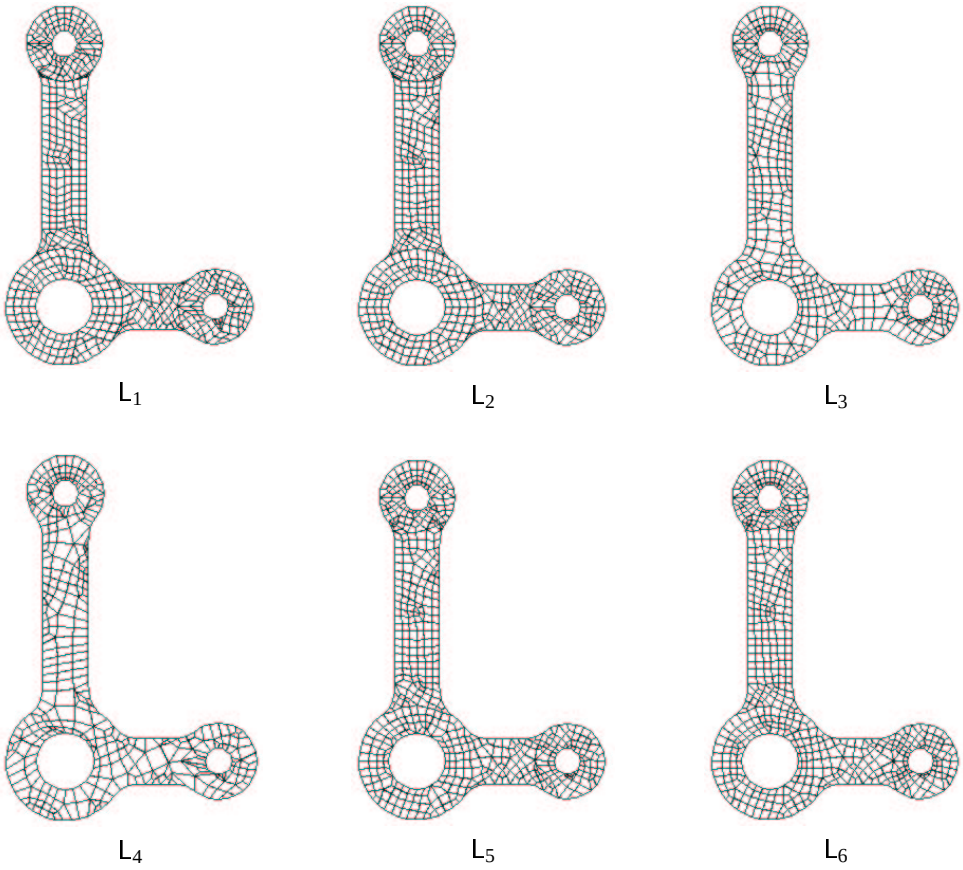


Figure 10: Six different quadrilateral meshes of the same domain. *Data courtesy of* INRIA Rocquencourt, France.

	$\overline{\tau}$	$\overline{\sigma}$	$\frac{1}{4}\overline{\zeta}$	$\frac{1}{2}\overline{\iota}$	$\frac{1}{4\sqrt{2}}\overline{Q}$	$\frac{1}{2}\overline{\kappa_2^1}$	$\frac{1}{2}\overline{\kappa_2^\infty}$
	− +	− +	− +	− +	− +	− +	− +
L_1	1.66 1 5.81	1.31 1 12.1	1.2 1 10.3	1.5 1 15.5	2.51 1 103	1.35 1 27.2	2.64 1 162
L_2	1.65 1.02 6.23	1.2 1 5.38	1.12 1 4.78	1.39 1.01 7.82	1.93 1.02 18.8	1.2 1 5.7	2.02 1 24.1
L_3	1.63 1.04 3.48	1.13 1 2.13	1.07 1 1.6	1.33 1.02 2.34	1.66 1.03 7.05	1.12 1 2.14	1.8 1 4.48
L_4	1.94 1.02 6.39	1.19 1 2.65	1.15 1 3.47	1.52 1.03 6.21	2.22 1.06 36	1.26 1 5.51	2.24 1 16.7
L_5	1.56 1.03 4.43	1.14 1 2.1	1.07 1 2.04	1.3 1.02 3.57	1.66 1.03 7.08	1.12 1 2.44	1.78 1 6.29
L_6	1.51 1.02 3.7	1.1 1 2.07	1.05 1 1.49	1.26 1.01 2.35	1.59 1.02 7.12	1.09 1 2.13	1.69 1 4.48

Table 2: Arithmetic mean and range of quality measures across six different quadrilateral meshes of the same domain.

Conclusions and future work

The results demonstrated in this article concerning quadrilateral quality measures are summarized in Table 3. Column "$= 1$" indicates which particular element optimizes the normalized quality; columns "$\boxed{\infty}$ "and $\Diamond$ provides the asymptotic behavior of the normalized quality when, respectively, the element is a rectangle with stretching factor τ tending to $+\infty$ and a rhombus with torsion θ tending to 0; column $\Diamond$; column "$\rightarrow \triangle$" indicates whether measure diverges or not in the case of a triangular degeneracy of the quadrilateral. Depending on the specific needs of the user, Table 3 allows one to choose which quality measure fits one's specific needs. In particular whether divergence to $+\infty$ in the case of a triangular degeneracy is desireable or not. Moreover, it seems that, in the case of Q, considering @ as five times worse than $\diamondsuit_{0.51}$ is certainly subject to caution. Overall, κ_2^1 and κ_2^∞ appear to be the best-behaved planar quadrilateral quality measures, in the case where triangular degeneracy detection is desireable.

Among other interesting questions remaining to be addressed are:

- point-placement strategies; in other words, given three vertices and a quality measure, what is (are) the optimal fourth vertex position(s) ?
- possible extensions to quadrilateral-sided hexahedrons;
- the correlation of these results to actual numerical quadrilateral-based numerical finite element or finite volume computations;

	$= 1$	$\boxed{\infty}$	$\Diamond$	$\rightarrow \triangle$
τ	rhombus	$\sim \tau$	$= 1$	no
σ	inscribed	$= 1$	$= \frac{\pi}{2\theta}$	no
$\frac{1}{4}\zeta$	square	$\sim \frac{\tau}{4}$	$\sim \frac{1}{\theta}$	no
$\frac{1}{2}\iota$	square	$\sim \frac{\tau}{2}$	$\sim \frac{1}{\theta}$	no
$\frac{1}{4\sqrt{2}}Q$	square	$\sim \frac{\tau}{2}$	$\sim \frac{\sqrt{2}}{\theta}$	yes
$\frac{1}{2}\kappa_2^1$	square	$\sim \frac{\tau}{2}$	$\sim \frac{1}{\theta}$	yes
$\frac{1}{2}\kappa_2^\infty$	square	$\sim \frac{\tau}{2}$	$\sim \frac{1}{\theta}$	yes

Table 3: Summary of quadrilateral quality measures.

– no theoretical reason prevents the use of these quality estimates in a non-euclidean context; however, actual implementations, *e.g.*, , in the case of non-unifrom riemanian metric, are certainly a challenge.

Acknowledgments

The author thanks:

– INRIA Rocquencourt, France, and in particular L. MARÉCHAL for having provided the quadrilateral meshes used in Subsection 9.2;

– C.A. KENNEDY for his critical and constructive reviews of this article.

References

[1] T.J. BAKER, Element quality in tetrahedral meshes, *Proc. 7th Int. Conf. Finite Element Methods in Flow Problems*, 1018-1024, Huntsville, U.S.A., 1989.

[2] T.J. BAKER, Deformation and quality measures for tetrahedral meshes, *Proc. ECCOMAS 2000*, Barcelona, Spain, September 2000.

[3] P.G. CIARLET and P.A. RAVIART, General LAGRANGE and HERMITE interpolation in $\mathbb{R}^n$ with applications to finite element methods, *Arch. Rational Mech. Anal.* **46**, 177-199, 1972.

[4] H.S.M. COXETER, Regular Polytopes, MacMillan, New York, 1963.

[5] J. DOMPIERRE *et al.*, Proposal of benchmarks for 3D Unstructured Tetrahedral Mesh Optimization, *Proc. 7th Int. Meshing Roundtable*, Sandia National Laboratories, Dearborn, Mi, USA, October 1998.

[6] L.A. FREITAG and P.M. KNUPP, Tetrahedral element shape optimization *via* the Jacobian determinant and condition number, *Proc. 8th Int. Mesh. Roundtable*, South Lake Tahoe, U.S.A., October 1999.

[7] P.J. FREY and P.L. GEORGE, Mesh Generation, Hermes Science Publishing, Oxford & Paris, 2000.

[8] P.M. KNUPP, Algebraic mesh quality metrics for unstructured initial meshes, *Finite Elements in Analysis and Design* **39**, 217-241, 2003.

[9] A. LIU and B. JOE, On the shape of tetrahedra from bisection, *Math. Comp.* **63** 207, 141-154, July 1994.

[10] PH.P. PÉBAY, Some Results About the Quality of Planar Mesh Elements *INRIA Research Report*, **4436**, Rocquencourt, April 2002.

[11] P.P. PÉBAY and T.J. BAKER, A comparison of triangle quality measures, *Proc. 10th Int. Meshing Roundtable*, Sandia National Laboratories, Newport Beach, Ca, USA, October 2001.

[12] V.T. PARTHASARATHY, C.M. GRAICHEN and A.F. HATHAWAY, A comparison of tetrahedral quality measures, *Fin. Elem. Anal. Des.* **15**, 255-261, 1993.

[13] J. ROBINSON, Some new distortion measures for quadrilaterals, *Finite Elements in Analysis and Design* **3**, 183-197, 1987.

Sundance: A HIGH-LEVEL TOOL FOR PDE-CONSTRAINED SIMULATION AND OPTIMIZATION

KEVIN R. LONG*

Sandia National Laboratories
MS 9217 P.O. Box 969
Livermore, CA 94551, U.S.A.

ABSTRACT: In this talk we describe Sundance, a software system for rapid development of parallel PDE simulation and optimization problems. Sundance was originally designed with PDE-constrained optimization in mind, but the features that make it suitable for use in PDE-constrained optimization turn out to also make it useful as a general-purpose research tool in PDE simulation.

Modern algorithms for PDE-constrained optimization require the application of operators and the solution of systems of equations that are different from those used in a single solution of the PDE, and require a closer degree of interaction between the PDE solver and optimizer than is usually available. Even with modern PDE frameworks such as Sandia's SIERRA and Nevada, it involves considerable development effort to obtain the quantities required for PDE-constrained optimization, and consequently the best algorithms for PDE-constrained optimization have been little used outside of academic research projects. Indeed, even research into these algorithms is hindered because of the considerable startup costs involved in writing a non-traditional PDE-simulator.

To facilitate research into, and application of, PDE-constrained optimization we have developed a general-purpose PDE solver system that has been designed from the ground up with large-scale, parallel, PDE-constrained optimization in mind. This system, called Sundance, accepts a system of coupled PDEs and boundary conditions written in symbolic form that is close to the notation in which a scientist or engineer would normally write them with pencil and paper. Each function or variation appearing in this symbolic description is annotated with a specification of the finite-element basis with which that object will be discretized. This information, along with a mesh, is then used by Sundance to assemble the implied discretized operators.

The symbolic interface to Sundance makes it quite simple to develop a high-performance code to create the non-traditional operators seen in PDE-constrained optimization. Furthermore, this symbolic interface is essentially a rapid-development tool that can be used for exploration of new solver algorithms, physical models, or finite-element formulations that are not available in standard codes.

KEY WORDS:

*krlong@ca.sandia.gov