SANDIA REPORT

SAND2012-9750
Unlimited Release
Printed November 2012

Incorporating the Min-Max Mesh Optimization Method within the Target-Matrix Paradigm

Patrick M. Knupp and David M. Day

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2012-9750
Unlimited Release
Printed November 2012

Incorporating the Min-Max Mesh Optimization Method within the Target-Matrix Paradigm

Patrick Knupp and David Day
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1318

Abstract

The Target-Matrix Optimization Paradigm (TMOP) is a method for improving mesh quality via node movement [11]. Through the use of appropriate pre-defined quality metrics and Target-matrices, TMOP provides the ability to define mesh quality in terms of element shape, volume, and edge orientation as a function of position within the mesh, thus facilitating high-level goals such as size-adapted meshing. The Paradigm is developed further in this paper by showing how it can be applied to the problem of improving the mesh in locations where the quality is the *worst*. Improving the worst quality portions of a mesh was investigated by Freitag [8], Amenta [1], and others, who showed that the problem can be formulated as a *min-max* optimization problem. The Target-paradigm is used to extend the previous work in two important respects. First, by using quality metrics from the Target-paradigm, quality can be user-specified via Target-matrices instead of the quality definition being hard-wired into the quality metric. Second, by making use of the sample point concept in TMOP, one can extend *min-max* mesh optimization to non-simplicial element types such as quadrilaterals. Numerical experiments were performed to illustrate these extensions and to investigate their utility. A non-gradient solver known as the *amoeba* was used to numerically compute the optimal solutions. This derivative-free method, in conjunction with the use of barrier metrics from TMOP, enables the solution of the min-max problem without the use of explicit constraints to guarantee that the optimal mesh is non-inverted. The experiments showed that the use of Target-metrics in min-max with triangle (or tetrahedral) element meshes is able to effectively improve the worst quality elements in terms of shape or in terms of size+shape. The experiments also showed that the mesh min-max extension

to quadrilateral element meshes can lead to acceptable meshes using barrier metrics from the Target-paradigm, provided that inter-element mesh smoothness is not important. Poor behavior was observed when using non-barrier Target-metrics in min-max with quadrilateral element meshes. For both simplicial and quadrilateral meshes, the min-sum method can be competitive with min-max, however.

Contents

List of Figures

Background and Introduction

Computational meshes are widely used in the solution of partial differential equations (PDEs) for simulations of continuum physics. They are also used in some visualization methods for rendering. Automatically generated meshes frequently suffer from defects such as inverted elements and poor quality. The quality of the mesh is important because it affects discretization accuracy, solver efficiency, and rendering effectiveness. Many mathematical techniques have been proposed for improving poor quality meshes. These include methods for improving the mesh even as it is generated (vertex insertion/deletion) [16] and post-processing methods in which mesh topology is modified (edge or element swapping) [9, 6] and/or vertex coordinates are modified (smoothing or optimization of an objective function that measures mesh quality) [3, 5, 7, 10, 14, 17, 4]. The latter set of methods are sometimes referred to as node-movement methods.

The fundamental concepts of the Paradigm were elucidated in [10], [11]. First, differentiable local mappings are assigned to each element of the mesh that is to be improved. Usually these mappings have the same form for each element type (e.g. the bilinear map from a square master element to an arbitrary quadrilateral). Second, a set of *sample points* within each master element is defined; these allow one to monitor quality *within* a mesh element (e.g., at the four corners of a quadrilateral element or at the mid-face of a quadratic hexahedral element). Third, the Jacobian matrix A derived from the mapping is evaluated at each sample point. The significance of the Jacobian matrix is that it contains geometric information about the mapping at the given sample point location; specifically, it can be used to measure lengths of tangents, local area or volume, and orientation of the tangent pair with respect to the global coordinate system. Fourth, for every sample point within the mesh, not only does there exist a Jacobian matrix, but also a matrix (of the same dimensions) which we call the *target-matrix*, W. The target-matrices are defined prior to optimization and serve as the mechanism by which the user supplies their own definition of mesh quality, in terms of the *desired* Jacobian matrix in the optimal mesh. Because a mesh typically contains thousands or millions of elements (and thus even more sample points), it is not practical for the user to specify the Target-matrices one by one. This issue is addressed through the use of automated *Target-calculators* which use high-level information about the meshing problem (e.g., characteristic length-scales). This information is related to the Target-matrices via their decomposition into matrix factors characterizing the shape, size, and orientation of the local tangent vectors. Given the pair of matrices A and W at a sample point, the matrix $T = AW^{-1}$, known as the *weighted* Jacobian matrix, provides a useful scaling of the Jacobian matrix. Fifth, the paradigm uses a set of local mesh quality metrics μ which are functions whose domain is a set of matrices and whose range is a subset of the real numbers. Usually the range is taken to be 0 to ∞, with 0 indicating optimal quality. For example, an important metric in the planar mesh case is the 2D non-barrier shape metric

$$\mu_1(T) = \| T \|_F^2 - 2\, det(T) \tag{1}$$

where T is the 2×2 weighted Jacobian matrix, $\| \cdot \|_F$ is the Frobenius matrix norm, and

$det(\cdot)$ is the matrix determinant. The domain of this metric is the set of all 2×2 real matrices. It can be shown that the range of this metric is 0 to ∞ for any T in the domain; further, $\mu_1 = 0$ if and only if $T_{2\times 2}$ is a scaled rotation matrix. If the scaled rotation is denoted by $s\,R$, with s scalar, then $T = sR$ means that $A = sRW$. Therefore, when the minimum is attained, the active Jacobian matrix in the optimal case is a scaled rotation times the target-matrix. This means that the minimizer will be impacted only by shape information within W and not by the scale or orientation information contained within W.

Finally, an objective function, which combines the values of the local mesh quality metric evaluated at the sample points, is needed to define an unconstrained optimization problem. To do this, the TMOP uses the concept of an objective function *template*. This approach is discussed in the next section since it was not fully elaborated in [11].

Objective Function Templates in the Target-matrix Paradigm

Let $\mathcal{S}$ be the set of all sample point indices within a given mesh and let $\mathcal{K} \subset \mathcal{S}$ be the subset of the sample point indices on which we wish to evaluate quality. The subset $\mathcal{K}$ could be, for example, the set of indices corresponding to sample points in a local mesh patch or it could be the set of indices corresponding to sample points within a given mesh element. Let K be the number of sample point indices within $\mathcal{K}$. Given $k \in \mathcal{K}$, the Jacobian matrix at sample point k is A_k.[1] Prior to optimization, a set of non-singular Target-matrices W_k is defined, one at each sample point, so we can form the weighted Jacobian matrix T_k. For a selected TMOP local quality metric $\mu(T)$, let $\mu_k = \mu(T_k)$ be the quality of the mesh at sample point k. To optimize mesh or patch quality one now needs an objective function to combine all the quality metric values μ_k into a single number representing the quality of the mesh (or patch). Because there is more than one way to create objective functions from metric values, we introduce the idea of an objective function template.

An objective function *template* in TMOP is a function from R^K to the real numbers (or some subset thereof) and which is used to combine the values of the quality metric evaluated at the sample points in $\mathcal{K}$. A template that is commonly used in mesh optimization is based on the ℓ_p norm of a vector. Let $v \in R^K$ have components v_k. Then $\ell_p(v) = |v|_p$, i.e.,

$$\ell_p(v) = \left(\sum_{k=1}^{K} |v_k|^p \right)^{1/p} \tag{2}$$

If we let $v_k = \mu_k$ be the vector of quality values, then ℓ_p (or, more frequently, ℓ_p^p) becomes an objective function template. The value $p = 2$ is frequently used. Although ℓ_p qualifies as an objective function template, it suffers from a number of drawbacks. Consider, for example,

[1] It is important to note that if j and k are distinct sample points within the same mesh element, A_j and A_k are equal in the case of linear simplicial elements, but are not necessarily equal for linear quadrilateral and hexahedral elements, nor are they necessarily equal for quadratic triangle and tetrahedral elements.

the cube geometry with a structured mesh of N perfect cube elements. If the quality $\mu_k = 1$ for the perfect cube, then the quality of the entire mesh, under the ℓ_p template, would be N. The quality of the mesh thus depends on the number of elements it contains (for example, if the cube mesh has 8 perfect cube elements it has quality 8, whereas if it has 27 perfect cube elements in has quality 27). One would like to use a template which can provide a quality assessment that is independent of the number of elements in the mesh. This also would make the numerical optimization problem easier when there are 10^7 or more elements per mesh. Also, the absolute value within the template can lead to non-smooth optimization problems if the local quality metric is not restricted to non-negative values.

An alternative template in TMOP which addresses the issues raised above is based on the Hölder (or Power) Mean. Recall that the Hölder mean is defined by[2]

$$H_p(v) = \left(\frac{1}{K} \sum_{k \in \mathcal{K}} v_k^p \right)^{1/p} \tag{3}$$

For the purpose of using this mean in TMOP, we can restrict our discussion to the case $v_k \geq 0$ for all k and $p > 0$ because all the Target-metrics are (by design) non-negative. We call $H_1(v)$ the AVE (average) template and $H_2(v)$ the RMS (root-mean-square) template. Within the context of TMOP, we will again replace each v_k with the quality μ_k of the mesh evaluated at the k^{th} sample point within the mesh. A measure of the quality of the *mesh* or patch based on the mean is thus

$$H_p(v) = \left(\frac{1}{K} \sum_{k \in \mathcal{K}} \mu_k^p \right)^{1/p} \tag{4}$$

and H_p^p is a useful template for an objective function. The power mean has the following properties useful for aggregating local mesh quality values:

1. For $a \geq 0$, if the range of v_k is $[a, b]$ for all k, then the range of $H_p(v)$ is $[a, b]$. This is important because if the mean is used to measure the quality of a mesh, then the range of the mesh quality will be the same as the range of the local Target-metric used in the mean.

2. If $\gamma > 0$, then $H_p(\gamma\, v) = \gamma\, H_p(v)$. Thus, uniform scaling of the metric results in uniform scaling of the template.

3. If, for some k, the value of v_k decreases, then the value of the template $H_p(v)$ decreases (for any p).

4. If $p < q$, then $H_p(v) < H_q(v)$.

[2]It is also useful sometimes to write $H_p(v_1, v_2, ..., v_K)$ in place of $H_p(v)$.

The power mean template can be naturally extended to $p = \infty$ by defining

$$H_{+\infty}(v) \equiv \max_{k \in K} v_k \tag{5}$$

We call $H_{+\infty}$ the MAX (maximum) template.

Finally, we mention the use of *hierarchical* templates in TMOP. The Hölder Mean templates above were applied to the single vector v. This results in what we call a non-hierarchical template. One can also form *hierarchical* templates that are useful in measuring the quality of a mesh or patch containing non-simplicial elements in terms of the quality of its elements. Let $v = (v_1, v_2, \ldots, v_K)$ be a vector in R^K. Define $\Pi\, v$ to be a partition of the vector v into a set of sub-vectors such that each component of v appears in one and only one of the sub-vectors and such that every sub-vector has at least one component. Let the set of vectors in $\Pi\, v$ be $(\Pi\, v)_1$, $(\Pi\, v)_2$, ..., $(\Pi\, v)_N$. We can then define the following hierarchical objective function template

$$\Theta_{pq}(\Pi\, v) = H_p^p(H_q^q[(\Pi\, v)_1], H_q^q[(\Pi\, v)_2], ..., H_q^q[(\Pi\, v)_N]) \tag{6}$$

A hierarchical measure of mesh or patch quality is

$$Q_{pq}(\Pi\, v) = H_p(H_q[(\Pi\, v)_1], H_q[(\Pi\, v)_2], ..., H_q[(\Pi\, v)_N]) \tag{7}$$

As an example, let $v_k = \mu_k$ be the value of the local quality metric at some sample point in the mesh or patch. Let us partition the vector v by assigning to $(\Pi\, v)_j$ all the values of the local quality metric which came from the sample points within element j. Then we can measure the quality of the mesh or patch by computing $\Theta_{pq}(\Pi\, v)$. Note that if the mesh contains mixed element types then the length of each of the vectors $(\Pi\, v)_n$ will vary. This gives us a solid method for measuring the quality of hybrid meshes.

In this paper we use the following hierarchical templates: (1) $\Theta_{\infty,\infty}(\Pi\, v) = H_\infty(v)$ and (2) $\Theta_{1,1}(\Pi\, v)$ (equals $H_1(v)$ when the length of each vector in the partition is the same). In section 5.2.19, we also use (3) $\Theta_{\infty,1}(\Pi\, v)$, and (4) $\Theta_{p,p}(\Pi\, v)$ with $p = 2$ or 4. The latter is equivalent to $H_p^p(v)$ when the length of each vector in the partition is the same.

Objective Functions in the Target-matrix Paradigm

A mesh *objective function* is a function of the mesh coordinates and can be constructed from an objective function template. Define $\mathcal{X}(\mathcal{K}) = \{x_1, x_2, \ldots, x_K\}$, $x_k \in R^d$, to be the set of vertex coordinates associated with the sample points in $\mathcal{K}$.[3] Because the local active

[3] d is the dimension of the space, usually 2 or 3.

matrices A_k depend on $\mathcal{X}$ (or a subset thereof), so does the matrix T_k. In turn, this means that $\mu_k = \mu(T_k) = \mu(T_k(\mathcal{X}))$ depends on the mesh vertices. When $v_k = \mu_k$, we can write $v = v(\mathcal{X})$ and a TMOP (non-hierarchical) objective function F_p is the composition of the template H_p^p with the vector $v(\mathcal{X})$, raised to the power p. Explicitly,

$$F_p(\mathcal{X}) = H_p^p(v(\mathcal{X})) \tag{8}$$

H_P^p is used in the objective function rather than H_p because this yields a non-dense Hessian matrix, convenient for numerical optimization methods.

One can also create hierarchical objective functions F_{pq} using the hierarchical objective function templates.

$$F_{pq}(\mathcal{X}) = \Theta_{pq}(\Pi\, v(\mathcal{X})) \tag{9}$$

Since most TMOP local quality metrics values range from 0 to ∞, TMOP objective functions generally range from 0 to ∞, with zero being the optimal value. As a result, TMOP objective functions are *minimized*, as opposed to being *maximized*. The value $p = 1$ is most commonly used because it is numerically convenient and often yields significant improvement in the mesh quality. In this case, the optimal mesh is one which minimizes the *average* value of the mesh quality metric over the set of sample points.

It is often pointed out by researchers in the field that minimizing the maximum value of the mesh quality metric may be superior to minimizing the average value since the maximum value corresponds to the locations within the mesh having the *worst* quality. Use of the H_∞ 'mean' as the template in creating TMOP objective functions means that, in theory at least, worst case mesh quality can be addressed within the Target-matrix paradigm and, more significantly, that most of the power of the paradigm can be brought to bear in this setting. Demonstrating this fact computationally is a major focus of the present paper. A *min-max* objective function within the Target-matrix paradigm is

$$F_\infty(\mathcal{X}) = \max_{k \in K} \mu(T_k(\mathcal{X})) \tag{10}$$

with $\mu(T)$ a Target-metric, $\mathcal{K}$ the set of sample points within a local mesh patch. For simplicial meshes (triangles or tetrahedra) we use only one sample point per element.

Application of the min-max optimization problem to the improvement of mesh quality on local mesh patches (containing one free vertex) is, of course, not new. Theory associated with this application is discussed in [1], which showed that if the local measure of mesh quality (a function of the free vertex) has convex level sets, then the optimal position is unique.[4] Diachin [8] created a numerical optimization solver to compute the optimal solution to the Min-Max problem, based on an active set method and which did not use Target-metrics. The method, applied to meshes consisting of more than one free vertex, was

[4]VanderZee [12] showed that some of the 2D TMOP metrics are convex, notably the one in equation (1).

0. Initialize the termination criteria

1. Loop over $i = 1, 2, \ldots, i_{outer-max}$, with $i_{outer-max}$ being the number of passes over the global mesh

2. Loop over $j = 1, 2, \ldots, J$, with J being the total number of free vertices in the global mesh

3. Get the j^{th} local mesh patch

4. Perform $i_{inner-max}$ iterations of the min-max optimization solver on the j^{th} local mesh patch and update the position of the j^{th} free vertex accordingly

5. End loop over local mesh patches

6. End loop over outer iterations

The list of free vertices in step 2 is not sorted according to quality; thus the ordering of the vertices depends on the order in which the vertices were created during the meshing step. This algorithm is also used in the present paper, except that we used a non-gradient (i.e., derivative-free) solver in step 4 rather than the OptMS active set solver. The non-gradient solver is described next.

A Non-Gradient solver for the min-max problem

It is well known that the min-max problem can lead to non-differentiable objective functions and this is generally true for the meshing application - the objective function is not everywhere differentiable with respect to the free mesh vertices. In particular, it is not differentiable at the optimal solution. Therefore, one should not use gradient or Hessian-based methods for solving the mesh min-max problem. The local patch mesh min-max optimization problem is low-dimensional since each patch consists of one free vertex and thus only 2 or 3 variables. Moreover, the objective function evaluations are inexpensive. The min-max problem can, for example, be solved using an optimization method which uses only values of the objective function. Direct search methods for mesh optimization where introduced in [13]. An example of direct search is given in the present paper using a solver from [15]. Although such methods are recognized as being relatively slow compared to more sophisticated approaches, they are adequate for our purpose, which is simply to illustrate how the mesh min-max can be naturally incorporated into the Target-matrix paradigm. Even among methods which do not use gradient or Hessian information, there are a lot of possible solvers; we used amoeba, which we implemented in the Mesquite code [2] so that we could take advantage of all the Target-matrix functionality.

The Amoeba Algorithm Tailored to the Mesh Optimization Problem

The amoeba algorithm is well-known and so we do not describe it here (see [15], for example). However, we will describe in this sub-section the way in which Mesquite's mesh data is translated into data which the amoeba can use, along with a few modifications made to the algorithm to place it within the context of the Target-matrix mesh optimization paradigm as implemented in Mesquite. Our implementation differs from the description in Numerical Recipes in the following ways:

1. For barrier metrics, if a point of the simplex (or the new point) is infeasible, we set the value of the OF to a large number to represent infinity.
2. For non-barrier metrics, we ignore the feasible flag altogether, and never set the OF value to a large number to represent infinity.
3. Because we wanted to investigate convergence behavior, the termination criterion for the amoeba was changed to number of inner iterations.
4. The initial diameter of the amoeba was equal to the smallest edge-length in the mesh, times an empirically determined factor f. This factor is problem dependent. In general, if f is over-small or over-large the convergence rate of the solver is relatively slow, while over-large values of f can also cause the solver to try vertex positions outside the feasible region, which is especially troublesome for barrier metrics. Initialization of this scale factor could therefore be an issue in practical mesh optimization, but was not an issue in this research.

Numerical Experiments

This section describes numerical experiments that were performed to provide evidence that our solution algorithm is working correctly and to illustrate some of the results of extending the mesh min-max method via the Target-matrix paradigm.

Numerical Experiment Test Plan

The experiment test plan is summarized in Table . Experiments 1, 1a, 1b, 2, and 2a optimize a 2D triangle mesh consisting of a single local patch (one free vertex). Experiments 3 and 4, investigate behavior on a triangle mesh with two free vertices. The seven experiments 1-4 were performed mainly to confirm that the non-gradient solver was implemented correctly and was able to provide correct solutions to the min-max method on triangle meshes. Experiments 5, 5a, 6, 6a, and 7 use a triangle mesh with 33 free vertices. Experiments 8, 8a, 8b, 9, 9a, 10, and 10a optimize meshes consisting of quadrilateral elements. Finally,

experiment 11 optimizes a 3D tetrahedral mesh. Where appropriate, the results of each experiment are displayed in the form of (i) the initial mesh, (ii) the optimized mesh, (iii) the initial mesh quality, (assessed by the relevant metric), (iv) the optimized mesh quality, assessed with the same metric, and (v) for meshes with a single free vertex, a plot of the objective functions' level set contours, optimization path, and convergence history. For the experiments in this report, we report only fully converged results because our focus is on the correctness of results, rather than efficiency of the solver.

Inputs for the numerical experiments are given in Table . Key input variables are:

1. Max template (H_∞) or Ave template (H_1).

2. TMOP metics:

   (a) 2D non-barrier shape metric (μ_1 in equation 1), or

   (b) 2D barrier shape metric μ_2,[5] or

   $$\mu_2(T) = \frac{\| T \|_F^2}{2\, det(T)} - 1 \tag{11}$$

   (c) 2D non-barrier shape+size metric μ_3, or

   $$\mu_3(T) = \| T \|_F^2 - 2\sqrt{\| T \|_F^2 + 2\, det(T)} + 2 \tag{12}$$

   (d) 3D non-barrier shape metric μ_4, or

   $$\mu_4(T) = \| T \|^3 - 3\sqrt{3}\, det(T) \tag{13}$$

   (e) 3D barrier shape metric μ_5

   $$\mu_5(T) = \frac{\| T \|^3}{3\sqrt{3}\, det(T)} - 1 \tag{14}$$

3. Target-matrix: Except for Experiment 7, the Target-matrix is a constant matrix (i.e., one which is the same at all sample points). In most of the experiments which contain triangle elements, the Target-matrix corresponds to an equilateral triangle with unit area, and is given by the 2×2 matrix

   $$W_{eq2} = \left(\frac{2}{\sqrt{3}}\right)^{1/2} \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix} \tag{15}$$

   For the tetrahedral element mesh, W_{eq2} is replaced with W_{eq3}, which corresponds to an equilateral tetrahedron with unit volume. The experiments with quadrilateral elements used $I_{2\times2}$ (the identify matrix) as the Target-matrix..

[5] The domain of this metric is any 2×2 real matrix whose determinant is positive.

Exp No.	Template	Metric	Target	Element	#Free	Mesh/Patch
1	Max	μ_1	W_{eq2}	Tri	1	Hexagon
1a	Max	μ_1	W_{eq2}	Tri	1	Horseshoe
1b	Ave	μ_1	W_{eq2}	Tri	1	Horseshoe
2	Max	μ_2	W_{eq2}	Tri	1	Hexagon
2a	Max	μ_2	W_{eq2}	Tri	1	Horseshoe
3	Max	μ_1	W_{eq2}	Tri	2	EqualTri2
4	Max	μ_2	W_{eq2}	Tri	2	EqualTri2
5	Max	μ_1	W_{eq2}	Tri	33	Circle
5a	Ave	μ_1	W_{eq2}	Tri	33	Circle
6	Max	μ_3	λW_{eq2}	Tri	33	Circle
6a	Ave	μ_3	λW_{eq2}	Tri	33	Circle
7	Max	μ_3	$\lambda(r) W_{eq2}$	Tri	33	Circle
8	Max	μ_1	$I_{2\times2}$	Quad	1	Square
8a	Max	μ_1	$I_{2\times2}$	Quad	1	Horseshoe
8b	Ave	μ_1	$I_{2\times2}$	Quad	1	Horseshoe
9	Max	μ_2	$I_{2\times2}$	Quad	1	Horseshoe
9a	Ave	μ_2	$I_{2\times2}$	Quad	1	Horseshoe
10	Max/Ave	μ_1, μ_2, μ_3	$I_{2\times2}$	Quad	21	UnstrQuad
10a	Max/Ave	μ_1	$I_{2\times2}$	Quad	2	Small Quad
11	Max/Ave	μ_4, μ_5	W_{eq3}	Tet	1322	Tire

Table 1. Numerical Experiment Test Plan

4. Element Type: Tri (triangles), Quad (quadrilaterals), or Tet (tetrahedral).[6] Triangles have one sample point within each element, quadrilaterals have four, and tetrahedra have one.

5. #Free is the number of free vertices in the global mesh (every local patch has one free vertex).

6. Mesh or Patch: A brief description of the mesh or patch is given.

All of the experiments were performed using the Non-gradient (amoeba) solver.

Experiments Confirming the Non-gradient Solver

Experiments 1-4 confirm that the non-gradient solver, in conjunction with some basic Target-metrics, was implemented correctly.

[6]Hexahedral elements were not investigated because the results of our investigations using quadrilateral elements were somewhat disappointing.

Numerical Experiment 1 Results

Figure 1 shows the initial and final computed meshes resulting from Experiment 1. The triangles in the initial mesh have different shapes and none are equilateral, whereas all the triangles in the computed optimal mesh are equilateral. This result is consistent with our expectations, which are based on (1) the symmetric arrangement of the patch boundary vertices, (2) use of a Target-matrix corresponding to an equilateral triangle, and (3) use of a Shape metric (which is invariant to element size and orientation). The maximum quality in the optimal mesh is zero, indicating that $A_k = sRW_k$ for each element k and for some unspecified scale and rotation factors. On the left, Figure 2 shows the level sets corresponding to the patch and non-barrier shape metric μ_1. The level sets are concentric circles around the central point of the hexagon patch; due to the high symmetry of the patch, the level set contours are smooth in this experiment. The path of the free vertex vs. iteration count is also shown; one sees the vertex move from the initial point in a counter-clockwise direction around the optimal point for a few iterations before it zooms in on the optimum location. On the right, Figure 2 shows the logarithm of the value of the objective function vs. the iteration count. In 57 inner (amoeba) iterations, the value of the objective function decreases from $\approx 10^0$ to $\approx 10^{-16}$. In experiment 1, the emprically-determined value for the initial size of the amoeba simplex was 4*the minimum edge-length. We conclude from this experiment that the solver was able to converge to the correct solution on this extremely simple, but useful, test problem.

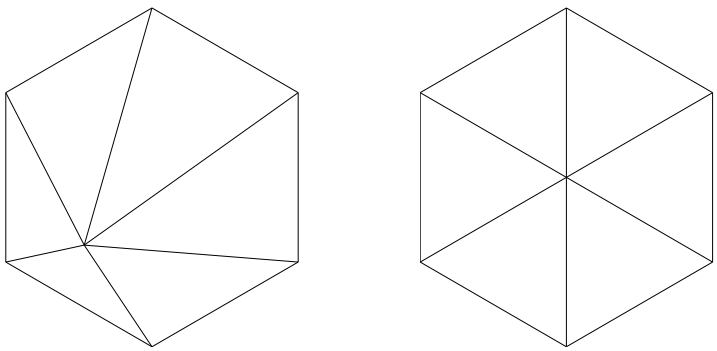

Figure 1. Mesh Patch in Experiment 1: (Left) Initial (Right) Optimized

Element Quality:
Initial Mesh μ_1: min=1.47, ave=1.47, max=1.47
Optimized Mesh μ_1: min=0, ave=0, max=0

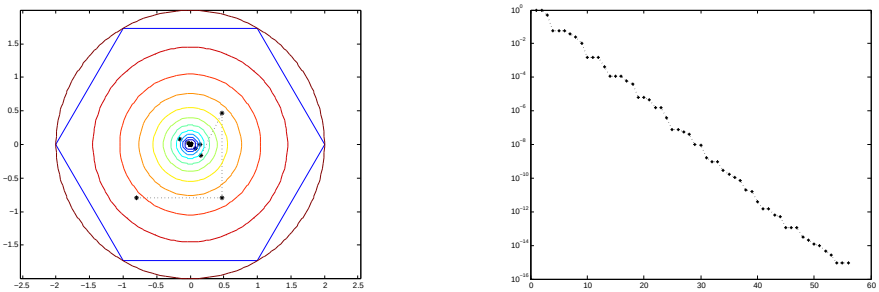

Figure 2. Experiment 1 (Left) Objective Function Contour Plot and Optimization Path (Right) Objective Function Value vs. Inner Iteration Count.
The minimum value, $\approx 10^{-16}$ (i.e., machine zero), was attained after 57 inner iterations.

Numerical Experiment 1a Results

Experiment 1a has the same set-up as Experiment 1 except that the patch is changed from the hexagon to the so-called *Horseshoe*. The significance of the latter patch is that it is non-convex. When this patch is used in conjunction with the non-barrier quality metric μ_1 (and with many other metrics), the optimal point lies outside the set of points at which the mesh is non-inverted. As Figure 3 shows, both the initial and the optimal meshes in this problem are inverted. Figure 4 (left) shows the level sets in this problem. The level sets are convex, but non-smooth. The level set plot confirms that the miminum of the objective function lies outside the patch domain, which explains why the optimal mesh is inverted. Note also that the patch does not admit a solution that consists of all elements having the desired equilateral shape. As a result, the values of the μ_1 quality metric are not zero on the optimal mesh. We do see from these quality metrics, however, that both the average and the worst case quality are improved by the optimization. The objective function vs. iteration count is given on the right in Figure 4. It shows the value of the objective function decreasing from ≈ 3 to $\approx 10^{-5}$ in 54 iterations. This experiment shows that the solver finds the correct optimal position and it also shows, incidentally, that inverted meshes can be generated when using the maximum template with non-barrier metrics.

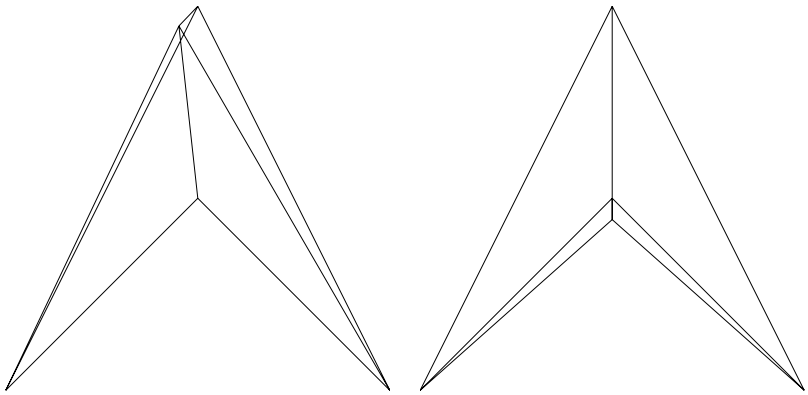

Figure 3. Mesh Patch in Experiment 1a: (Left) Initial (Right) Optimized (The two elements at the bottom of the optimized mesh are inverted because the optimal location of the free vertex lies outside the domain.)

Element Quality:
Initial Mesh μ_1: min=2.18, ave=3.93, max=4.20
Optimized Mesh μ_1: min=2.41, ave=2.41, max=2.41

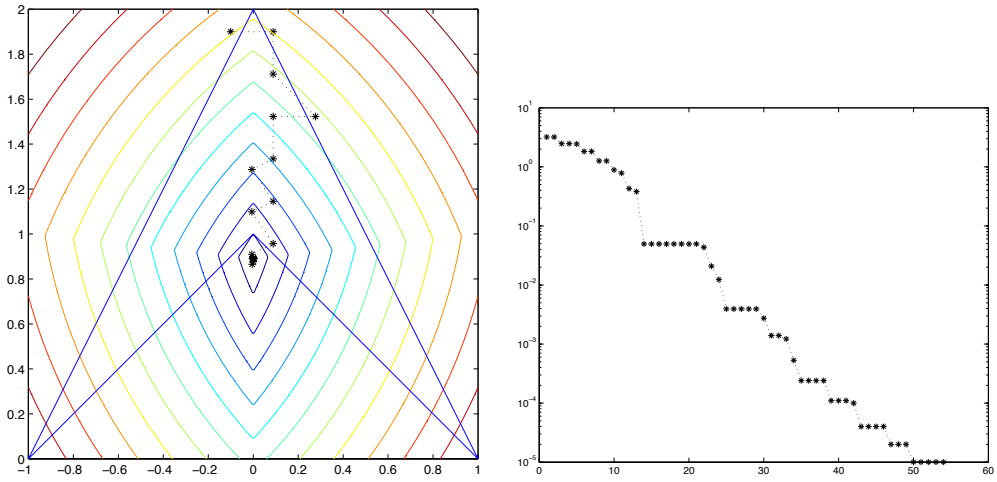

Figure 4. Experiment 1a (Left) Objective Function Contour Plot and Optimization Path
(Right) Objective Function Value vs. Inner Iteration Count. The minimum value, $\approx 10^{-5}$, was attained after 50 inner iterations.

Numerical Experiment 1b Results

Experiment 1b has the same set-up as Experiment 1a except that instead of using the maximum template, we use the Hölder Mean (H_p) template (with $p = 1$) so that the objective function is simply the linear average of the quality of the four elements in the patch. The initial mesh patch is the same as in experiment 1a, while the optimized mesh in 1b is similar to, but not exactly the same as in 1a (see Figure 5). Level set contours for Experiment 1b are shown in Figure 6 (left). Comparing the contours to those in Experiment 1a, we see that in 1b, the contours are smooth, as well as convex, and that the optimal point is slightly lower than it was in 1a. The resulting optimal mesh is thus inverted in both 1a and 1b. In spite of that, one sees from the values of the quality metrics that both the average and the worst case quality are improved by the optimization. The worst case quality is not as improved as much as it was in Experiment 1a since the maximum template was not used. The path of the free vertex and plot of objective function value vs. iteration count (Figure 6, right) are similar to those in Experiment 1a. This experiment shows that the solver again finds the correct optimal position and that the solver can work correctly with the Mean template as well as the Maximum. The experiment also shows, incidentally, that the choice of objective function template can affect both the location of the optimum and the smoothness of the level set contours, even when the same quality metric is employed.

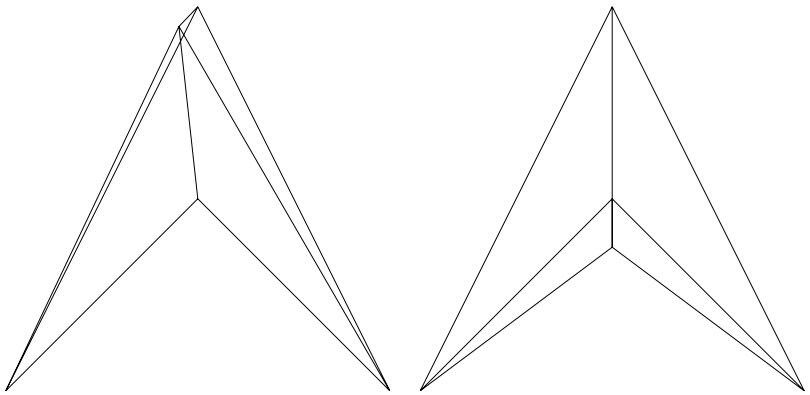

Figure 5. Mesh Patch in Experiment 1b: (Left) Initial (Right) Optimized (The optimized mesh contains two inverted elements.)

Element Quality:
Initial Mesh μ_1: min=2.18, ave=3.93, max=4.20
Optimized Mesh μ_1: min=2.19, ave=2.39, max=2.59

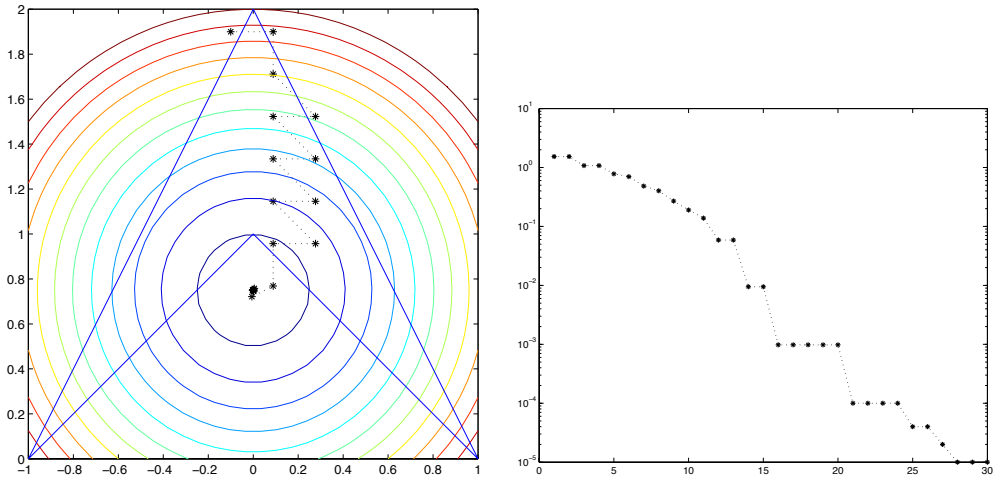

Figure 6. Experiment 1b (Left) Objective Function Contour Plot and Optimization Path
(Right) Objective Function Value vs. Inner Iteration Count. The minimum value, $\approx 10^{-5}$, was attained after 30 inner iterations.

Numerical Experiment 2 Results

The set-up in Experiment 2 is identical to Experiment 1 except that the local quality metric was changed from the non-barrier shape metric μ_1 to the barrier shape metric μ_2. The primary purpse of the test was to verify that the non-gradient solver can work correctly when using a barrier metric as the basis of the objective function.

Figure 7 shows the initial and final computed meshes resulting from Experiment 2. These results are identical to those found in Experiment 1 even though a different metric has been used. This is expected due to the high symmetry of the patch and in spite of the fact that the level set contour plots for the two metrics are otherwise different (compare Figures 2 and 8). In the latter case, the metric is only defined for $det(T) > 0$, so the contours do not extend outside the domain. They, in fact, become infinte on the boundary of the domain. Therefore, the initial mesh must be non-inverted in order to apply this (and other barrier) metrics. Figure 8 (right) shows that the value of the objective function in Experiment 2 decreases from $\approx 10^0$ to $\approx 10^{-16}$ in 53 iterations.

In experiment 2, the emprically-determined value for the initial size of the amoeba simplex was 4*the minimum edge-length.

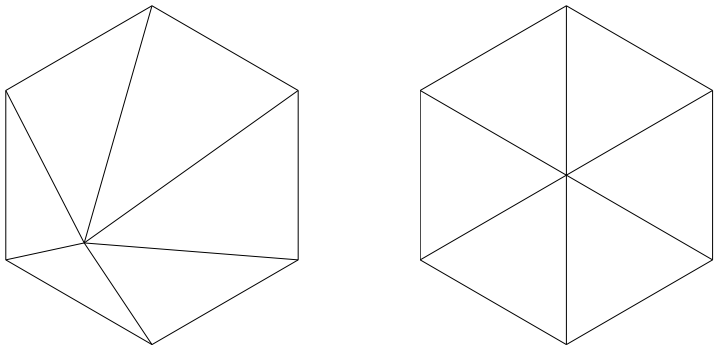

Figure 7. Mesh Patch in Experiment 2: (Left) Initial (Right) Optimized

Element Quality:
Initial Mesh μ_2: min=0.131, ave=0.282, max=0.324
Optimized Mesh μ_2: min=0, ave=0, max=0

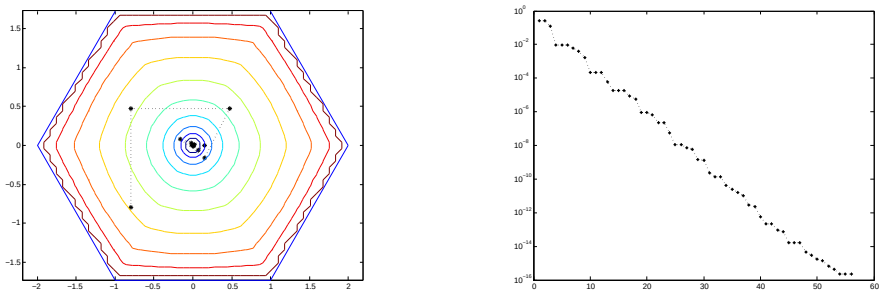

Figure 8. Experiment 2 (Left) Objective Function Contour Plot and Optimization Path
(Right) Log of Objective Function Value vs. Inner Iteration Count.
The minimum value, $\approx 10^{-16}$ (i.e., machine zero), was attained after 53 inner iterations.

Numerical Experiment 2a Results

The set-up in Experiment 2a was the same as in Experiment 2 except that the hexagon patch was replaced by the Horseshoe patch. The primary purpose of this experiment was to verify that the solver can work correctly on a non-convex patch when using a barrier metric. In this problem, the position of the free vertex in the initial mesh was changed from its position in Experiments 1a and 1b so that the initial mesh is non-inverted, as required when using a barrier metric. See Figure 9 for a look at the initial and optimized meshes. In this case, the optimal mesh is non-inverted, as one would expect from using a barrier metric. The level set contours in this problem are shown in Figure 10. Because this is a barrier metric, the metric is only defined on the feasible region, and thus the contours lie in a subset of the patch domain. Moreover, the optimum lies inside the feasible region. It is interesting to compare the level set contours in Experiment 2a to those in 1a; the difference between the two is due solely to the change to a barrier metric from a non-barrier metric.

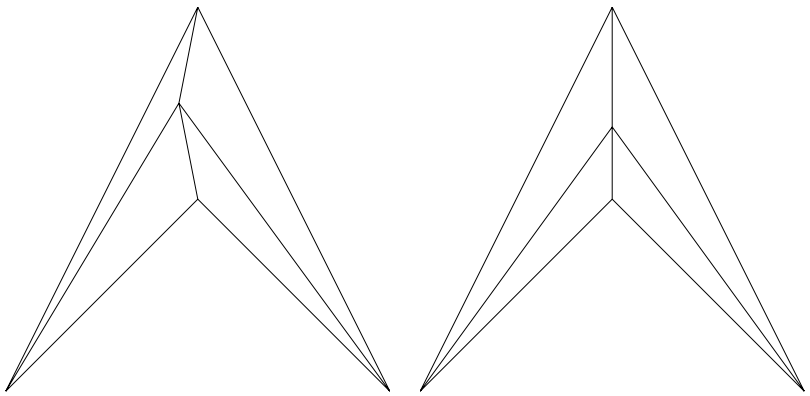

Figure 9. Mesh Patch in Experiment 2a: (Left) Initial (Right) Optimized. There are no inverted elements.

Element Quality:
Initial Mesh μ_2: min=1.56, ave=3.57, max=7.00
Optimized Mesh μ_2: min=2.85, ave=2.85, max=2.85

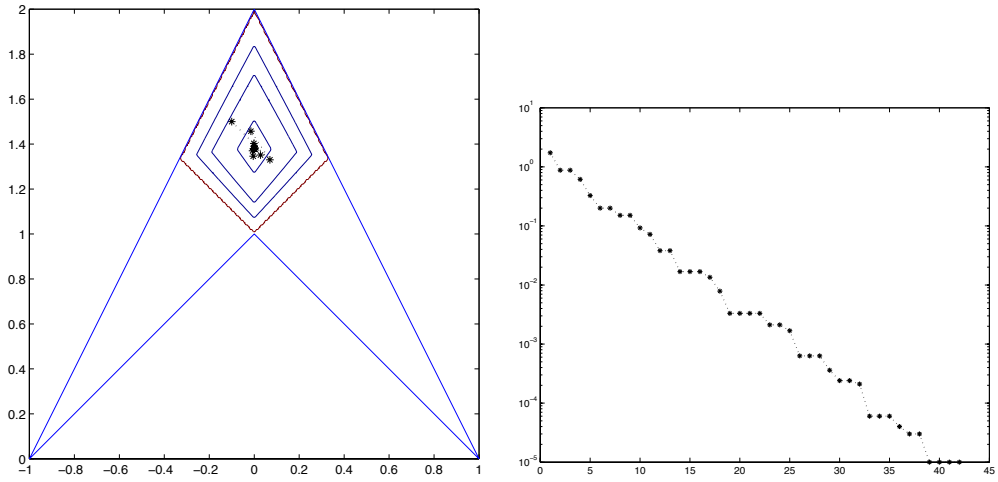

Figure 10. Experiment 2a (Left) Objective Function Contour Plot and Optimization Path
(Right) Objective Function Value vs. Inner Iteration Count. The minimum value, $\approx 10^{-5}$, was attained after 42 inner iterations.

Numerical Experiment 3 Results

The set-up in Experiment 3 is the same as in Experiment 1 except that we change to a mesh having *two* free vertices. To converge this problem requires increasing the outer iteration criterion from 1 (used in all the single free vertex meshes) to a larger value (2 suffices in this case). Once again, the solution to this problem is known (all equilateral triangles), therefore it is easy to ascertain whether or not the non-gradient solver was able to find this solution. Figure 11 shows the initial and final computed meshes resulting from Experiment 3. Judging from this result, the derivative free (non-gradient) solver and outer loop over the patches worked correctly on this problem.

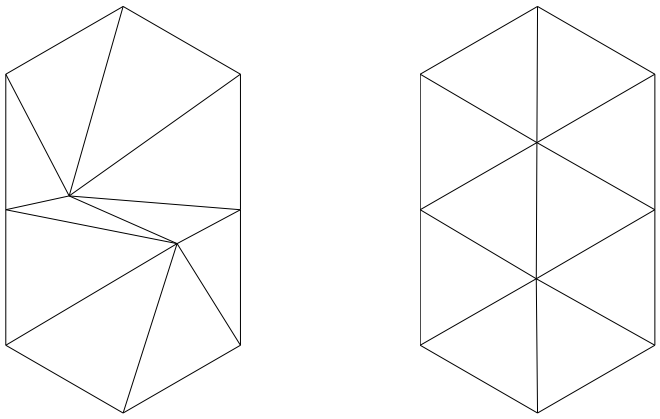

Figure 11. Mesh Patch in Experiment 3: (Left) Initial (Right) Optimized

Element Quality:
Initial Mesh μ_1: min=1.03, ave=1.74, max=4.19
Optimized Mesh μ_1: min=0, ave=0, max=0

Numerical Experiment 4 Results

The set-up in Experiment 4 is that same as in Experiment 3 except that we changed from using μ_1 to using the barrier metric μ_2. Once again, the solver is able to correctly compute the correct solution (see Figure 12).

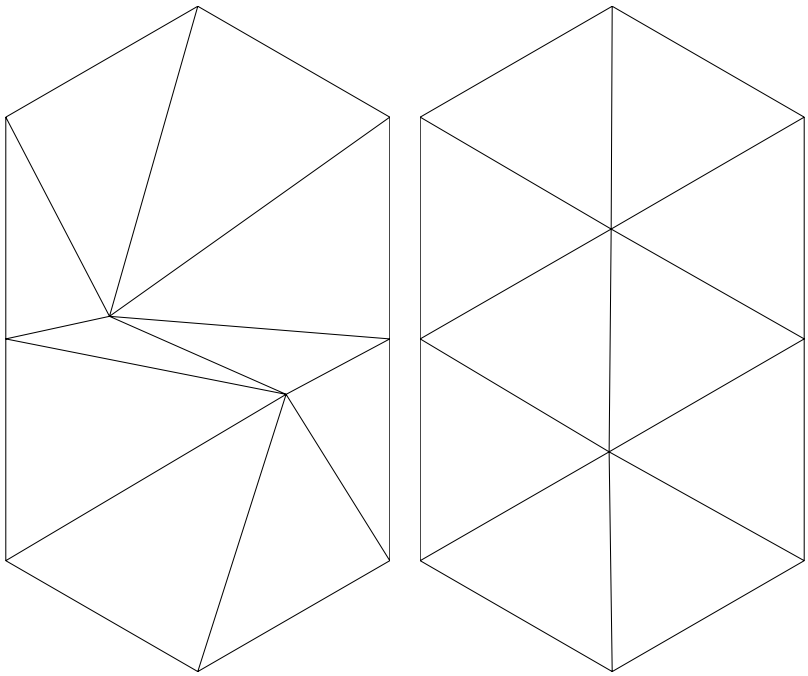

Figure 12. Mesh Patch in Experiment 4: (Left) Initial (Right) Optimized

Element Quality:
Initial Mesh μ_2: min=0.100, ave=0.474, max=0.788
Optimized Mesh μ_2: min=0.000082, ave=0.000219, max=0.000346

Experiments Illustrating Target-metrics within the Min-max Method

Experiments 5-7 illustrate the min-max method on triangle meshes with multiple vertices, with its extension to Target-paradigm quality metrics (barrier, non-barrier, shape, shape+size).

Numerical Experiment 5 Results

The set-up in Experiment 5 is the same as in Experiment 1 except that the hexagonal patch with one free vertex is replaced by a multi-vertex triangle mesh on a circular domain. Figure 13 shows the initial and optimized meshes on the Left and Right, respectively. On the whole, the elements in the optimized mesh are closer to being equilateral than they are in the initial mesh. The objective function decreases from 3.73 on the initial mesh to 0.538 on the optimized mesh, and the average quality also decreases. In Table 2 we report the values of the objective function on the optimized mesh for various values of the inner termination criterion, the outer criterion, and the relative size of the amoeba in the Non-gradient solver. By design, each entry in the table takes roughly the same amount of computational work because the product of the inner and outer termination criteria is always 256. Each row in the table reflects a fixed amoeba size. The table shows that in Experiment 5, for a fixed amount of computational work (256), the objective function is the smallest when either 32 or 16 inner iterations are performed, in concert with 8 or 16 outer iterations respectively, regardless of the size of the amoeba. Clearly, these ideal iteration limits will be problem dependent and will, among other things, depend on the number of vertices in the mesh.

We now take a closer look at the quality of the initial and optimal meshes in Experiment 5 in terms of the distribution of quality. Quality distribution can be displayed in terms of histograms which show the number of elements in the global mesh whose quality is between two numbers which define a quality bin. For example, we show in Figure 14 a quality histogram for the initial mesh, measured using the non-barrier shape metric μ_1. The histograms shows 23 triangle elements having quality between (0, 0.4), 15 between (0.4,0.8), and so on. There is one element in the worst quality bin (3.6-4.0), corresponding the the maximum quality of 3.72948. The histogram for the optimized mesh in Experiment 5 is shown in Figure 15; optimzation was based on the MAX template. The worst quality element in this mesh has quality 0.525371. Notice that the bins in this histogram are not the same as they were for the initial mesh. In order to effectively compare the two hitsograms we show a *common-scale* histogram in Figure 16. The figure shows that the worst quality elements have improved so that they now belong to the second best quality bin. The standard deviation of the histogram for the optimal mesh is much smaller than it is for the initial mesh.

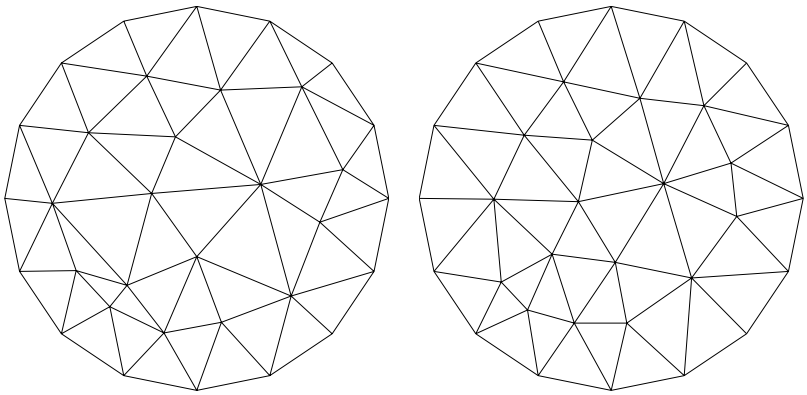

Figure 13. Mesh Patch in Experiment 5: (Left) Initial (Right) Optimized w/MAX

Element Quality:
Initial Mesh μ_1: min=0.005, ave=0.591, max=3.73
Optimized Mesh μ_1: min=0.034, ave=0.385, max=0.525

```
ElementMaxQM(TShapeNB1) histogram: (log10 plot)
(  0-0.4) |==================================================23
(0.4-0.8) |==================================================15
(0.8-1.2) |6
(1.2-1.6) |1
(1.6-2  ) |1
(  2-2.4) |1
(2.4-2.8) |0
(2.8-3.2) |0
(3.2-3.6) |0
(3.6-4  ) |1

 Histogram Statistics:
    minimum       average         rms          maximum    std.dev.
    0.00510058    0.591426    0.889919     3.72948       0.664959
```

Figure 14. Quality Histogram for Initial Mesh, using μ_1.

Rel Size	(256,1)	(128,2)	(64,4)	(32,8)	(16,16)	(8,32)	(4,64)	(2,128)
1	1.13	0.718	0.640	*0.557*	0.712	0.887	1.50	72
0.5	1.13	0.718	0.641	0.641	*0.621*	1.38	1.51	72
0.25	1.13	0.718	0.641	*0.538*	0.599	1.06	1.40	72
0.125	1.13	0.718	0.634	0.644	*0.596*	0.961	1.20	72

Table 2. Experiment 5: Value of Objective Function after Optimization vs. Relative Size of Amoeba and (inner,outer) iteration limits.

Product of inner and outer is 256 to approximately represent a fixed amount of work.

For the set of relative sizes of the amoeba, the combination (32,8) or (16,16) generally give the smallest value of the objective function.

```
ElementMaxQM(TShapeNB1) histogram:
(   0-0.06) |======3
(0.06-0.12) |=========4
(0.12-0.18) |====2
(0.18-0.24) |=========4
(0.24-0.3 ) |0
( 0.3-0.36) |=========4
(0.36-0.42) |====2
(0.42-0.48) |===============7
(0.48-0.54) |================================================22
(0.54-0.6 ) |0

Histogram Statistics:
  minimum     average     rms            maximum     std.dev.
  0.0341927   0.385186    0.420712    0.525371      0.169206
```

Figure 15. Quality Histogram for Optimal Mesh (MAX template), using μ_1.

```
ElementMaxQM(TShapeNB1) histogram (initial mesh): (log10 plot)
(  0-0.4) |==================================================23
(0.4-0.8) |==================================================15
(0.8-1.2) |6
(1.2-1.6) |1
(1.6-2  ) |1
(  2-2.4) |1
(2.4-2.8) |0
(2.8-3.2) |0
(3.2-3.6) |0
(3.6-4  ) |1


ElementMaxQM(TShapeNB1) histogram (optimal mesh): (log10 plot)
(  0-0.4) |==================================================19
(0.4-0.8) |==================================================29
(0.8-1.2) |0
(1.2-1.6) |0
(1.6-2  ) |0
(  2-2.4) |0
(2.4-2.8) |0
(2.8-3.2) |0
(3.2-3.6) |0
(3.6-4  ) |0
```

Figure 16. Common-scale histograms for the initial and optimal meshes from Experiment 5, using μ_1.

Numerical Experiment 5a Results

The set-up in Experiment 5a is the same as in Experiment 5 except that we use the AVE template instead of the MAX. Figure 17 shows the mesh optimized using the AVE template. The mesh has good quality and is an improvement over the initial mesh. Figure 18 shows that the mesh optimized with the AVE template is nearly indistinguishable visually from the mesh optimized with the MAX (i.e., the optimized mesh in Experiment 5). The optimal mesh in Experiment 5a looks very slightly *smoother* than the one in Experiment 5.

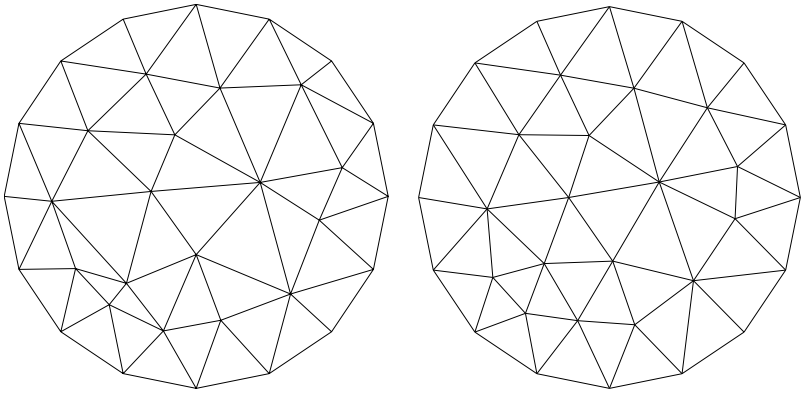

Figure 17. Mesh Patch in Experiment 5a: (Left) Initial (Right) Optimized w/AVE

Element Quality:
Initial Mesh μ_1: min=0.005, ave=0.591, max=3.73
Optimized Mesh μ_1: min=0.002, ave=0.305, max=1.02

We showed in Figure 14 a quality histogram for the initial mesh in Experiment 5a, measured using the non-barrier shape metric μ_1. The histogram for the optimized mesh in Experiment 5a is shown in Figure 19; optimization was based on the AVE template. The worst quality element has a value of 1.01999. Next, Figure 20 shows common-scale histograms for the two meshes (initial and optimal) in Experiment 5a. Finally, we show in Figure 21 the quality histogram for all three meshes in Experiments 5 and 5a. We see that, while MAX results in better *worst quality* than AVE, the worst quality resulting from optimizing using the AVE template is still much better than it was in the initial mesh. Moreover, the average quality is better with AVE template than it is with Max template.

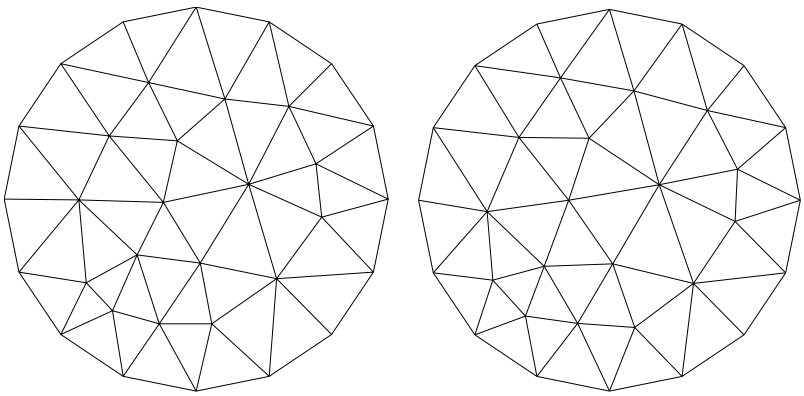

Figure 18. Optimized Meshes from Experiments 5 and 5a: (Left) Optimized w/MAX (Right) Optimized w/AVE

Element Quality:
Left Mesh μ_1: min=0.034, ave=0.385, max=0.525
Right Mesh μ_1: min=0.002, ave=0.305, max=1.02

```
 ElementMaxQM(TShapeNB1) histogram: (log10 plot)
 (  0-0.2) |==================================================20
 (0.2-0.4) |==================================================15
 (0.4-0.6) |6
 (0.6-0.8) |3
 (0.8-1  ) |3
 (  1-1.2) |1
 (1.2-1.4) |0
 (1.4-1.6) |0
 (1.6-1.8) |0
 (1.8-2  ) |0

Histogram Statistics:
 minimum     average     rms             maximum     std.dev.
 0.0018808   0.305241    0.397133    1.01999         0.254052
```

Figure 19. Quality Histogram for Optimal Mesh (AVE template), using μ_1.

```
Common Scale Histograms for Mean Obj functon max metric

************** Common-scale Histograms **************

   ElementMaxQM(TShapeNB1) histogram (initial mesh): (log10 plot)
   (  0-0.4) |==================================================23
   (0.4-0.8) |==================================================15
   (0.8-1.2) |6
   (1.2-1.6) |1
   (1.6-2  ) |1
   (  2-2.4) |1
   (2.4-2.8) |0
   (2.8-3.2) |0
   (3.2-3.6) |0
   (3.6-4  ) |1


   ElementMaxQM(TShapeNB1) histogram (optimal mesh): (log10 plot)
   (  0-0.4) |==================================================35
   (0.4-0.8) |==========================================9
   (0.8-1.2) |4
   (1.2-1.6) |0
   (1.6-2  ) |0
   (  2-2.4) |0
   (2.4-2.8) |0
   (2.8-3.2) |0
   (3.2-3.6) |0
   (3.6-4  ) |0
```

Figure 20. Common-scale histograms for the initial and optimal meshes from Experiment 5a, using μ_1.

```
TShapeNB1 histogram (initial mesh): (log10 plot)
(  0-0.4) |==================================================23
(0.4-0.8) |==================================================15
(0.8-1.2) |6
(1.2-1.6) |1
(1.6-2  ) |1
(  2-2.4) |1
(2.4-2.8) |0
(2.8-3.2) |0
(3.2-3.6) |0
(3.6-4  ) |1


TShapeNB1 histogram (optimal mesh MAX): (log10 plot)
(  0-0.4) |==================================================19
(0.4-0.8) |==================================================29
(0.8-1.2) |0
(1.2-1.6) |0
(1.6-2  ) |0
(  2-2.4) |0
(2.4-2.8) |0
(2.8-3.2) |0
(3.2-3.6) |0
(3.6-4  ) |0

TShapeNB1 histogram (optimal mesh AVE): (log10 plot)
(  0-0.4) |==================================================35
(0.4-0.8) |==================================================9
(0.8-1.2) |4
(1.2-1.6) |0
(1.6-2  ) |0
(  2-2.4) |0
(2.4-2.8) |0
(2.8-3.2) |0
(3.2-3.6) |0
(3.6-4  ) |0
```

Figure 21. Common-scale histograms for the three meshes in Experiments 5 and 5a: (Top) Initial Mesh; (Middle) Opt with MAX; (Bottom) Opt with AVE, using μ_1.

Numerical Experiment 6 Results

The set-up for Experiment 6 is the same as Experiment 5 except that 6 uses the non-barrier Shape+Size metric μ_3 and uses a Target-matrix of the form $W = \lambda W_{eq}$. This metric is designed to improve both the shape of the triangle elements and simultaneously, their size (area). To use this metric one must calculate the scaling constant λ carefully, based on the area of the domain, otherwise very bad quality meshes can result. Recall that the metric $\mu_3(T) = 0$ if and only if $T = R$, with R an arbitrary rotation matrix. From this, it follows that the minimizing mesh obeys $A = RW$. Substituting the particular form of the Target-matrix we wish to use, we have $A = R\,\lambda\,W_{eq}$, from which we need $det(A) = \lambda^2\, det(W_{eq}) = \lambda^2$. Since λ is to be a constant (i.e., the same at all sample points) in order to create equal-sized elements, we set $det(A)$ equal to twice the area of the circular domain (π times the radius r, squared) divided by the number N of triangular elements in the mesh. From this, we determine that

$$\begin{aligned} \lambda &= \sqrt{det(A)} \\ &= \sqrt{2\frac{\pi r^2}{N}} \end{aligned}$$

is a reasonable choice for the scale factor. In this problem, $N = 48$ and $r = 5$, so we obtain $\lambda = 1.809$. Results are shown in Figure 22. Compared to the initial mesh, the optimized mesh has better-shaped elements with more nearly equal areas. The optimized mesh in Experiment 6 also has more equal-sized elements than the optimized mesh in Experiment 5 because the latter focused exclusively on shape improvement. Experiment 6 thus illustrates that, by using quality metrics from the Target-paradigm, one can simultaneously control both element shape and size within the min-max method.

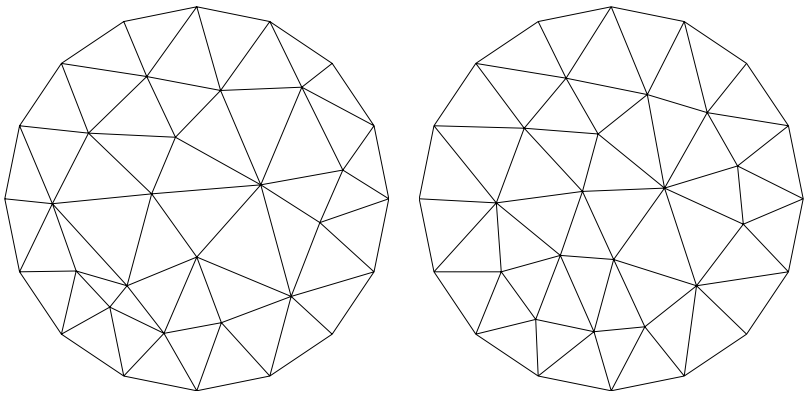

Figure 22. Mesh Patch in Experiment 6: (Left) Initial (Right) Optimized w/MAX

Element Quality:
Initial Mesh μ_3: min=0.014, ave=0.138, max=0.518
Optimized Mesh μ_3: min=0.024, ave=0.081, max=0.108

Scale Factor =1.809

Numerical Experiment 6a Results

The set-up in Experiment 6a is the same as in Experiment 6 except that we use the AVE template instead of the MAX. The results are shown in Figure 23; good quality is obtained.

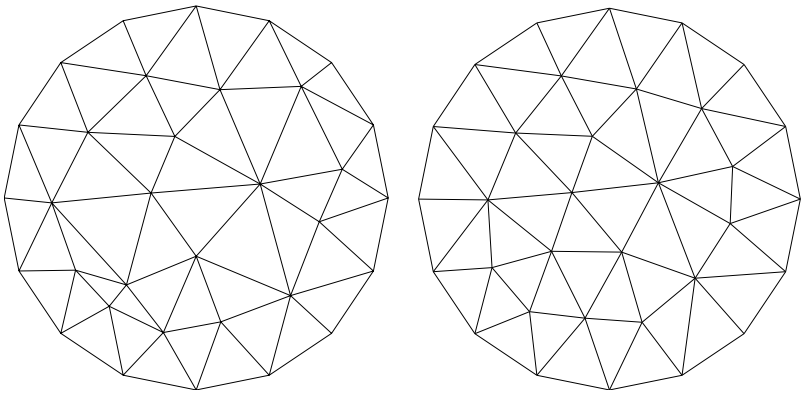

Figure 23. Mesh Patch in Experiment 6a: (Left) Initial (Right) Optimized w/AVE

Element Quality:
Initial Mesh μ_3: min=0.014, ave=0.138, max=0.518
Optimized Mesh μ_3: min=0.007, ave=0.067, max=0.161

Scale Factor =1.809

Numerical Experiment 7 Results

The set-up for Experiment 7 is the same as Experiment 6 except that the Target-matrix is scaled by a positive number that varies with position in the domain. In particular, we used $\lambda_k = \left(\frac{3}{5}r_k + \frac{1}{2}\right)^{\alpha}$, with r_k being the distance of the sample point (element centroid) from the center of the circle and α a parameter. The Target-matrices were, $W_k = \lambda_k W_{eq2}$ and, due to the choice of function in λ_k, we expect larger elements near the periphery of the domain and smaller ones near the center. The initial mesh in this experiment is the same as in Experiment 6 and can be seen in the left side of Figure 23. Three optimized meshes, for $\alpha = 1.0, 1.5, 2$, are shown in Figure 24, Note that the initial mesh quality is different in the three cases because, even though the same quality metric is used, the different values of α create three different sets of Target-matrices. This example illustrates that variable sized elements may be created by the min-max method using metrics from the Target-paradigm.

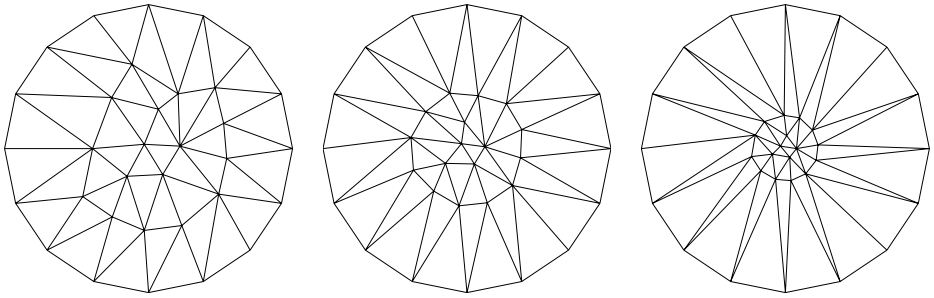

Figure 24. Optimized Meshes in Experiment 7: (Left) $\alpha = 1.0$ (Middle) $\alpha = 1.5$ (Right) $\alpha = 2.0$

Element Quality:
Initial Mesh μ_3, $\alpha = 1.0$: min=0.039, ave=0.649, max=10.1
Optimized Mesh μ_3, $\alpha = 1.0$: min=0.017, ave=0.248, max=0.329
Initial Mesh μ_3, $\alpha = 1.5$: min=0.025, ave=1.11, max=15.7
Optimized Mesh μ_3, $\alpha = 1.5$: min=0.131, ave=0.422, max=0.127
Initial Mesh μ_3, $\alpha = 2.0$: min=0.025, ave=1.62, max=24.0
Optimized Mesh μ_3, $\alpha = 2.0$: min=0.111, ave=0.563, max=0.760

Mesh min-max with 3D Tetrahedral Elements and Target-metrics

This section provides evidence, albeit limited, that the mesh min-max algorithm with Target-metrics can improve 3D (volume) mesh quality.

Numerical Experiment 11 Results

Experiment 11 optimizes a tetrahedral mesh on a portion of a tire. Table 3 shows the quality of the initial mesh and four optimized meshes. The four optimized meshes correspond to the combinations (MAX, μ_1), (MAX, μ_2), (AVE, μ_1), and (AVE, μ_2). As the table shows, the initial mesh contains some rather poor quality elements, as measured by the maximum values of μ_1 and μ_2. None-the-less, the initial mesh contains no inverted elements. Unfortunately, optimization with MAX and μ_1 creates 87 inverted elements, even though the maximum value of μ_1 is decreased. This can happen because μ_1 has no barrier. Optimization with AVE and μ_1 improved both the maximum value of μ_1 and the average value. However, even with the AVE template, the optimized mesh contains 4 inverted elements. Luckily, the initial mesh is non-inverted, so it is possible to optimize it using the barrier shape metric μ_2. In this case, the MAX template strongly reduces the maximum value of μ_2 and one obtains a non-inverted mesh. The AVE template with μ_2 reduces both the maximum and average values of the metric and the resulting mesh is non-inverted. Figures 25 and 26 show the common-scale histograms for the two meshes, MAX vs. AVE templates. As one sees, the optimal mesh produced using the MAX template has fewer outliers than the one produced using the AVE template. The histogram for the AVE template has twelve elements (out of 10920) whose quality is worse than the worst quality element in the optimal mesh produced using the MAX template. However, one wonders how important this difference is when considering that the histogram for the initial mesh (Figure 27) shows the quality of its worst element to be 106.7. From that point of view, both MAX and AVE have significantly improved the initial mesh in terms of worst quality.

Case	μ_1: min, ave, max. (# inv)	μ_2: min, ave, max, std. (# inv)
Initial	0.004, 1.93, 48.6 (0)	0.002, 0.512, 106.7, 1.44 (0)
opt MAX	0.021, 2.88, 23.0 (87)	0.007, 0.683, 3.52, 0.40 (0)
opt AVE	0.003, 1.75, 43.6 (4)	0.007, 0.414, 6.55, 0.37 (0)

Table 3. Tire Mesh Quality: min, ave, max (#inverted) out of 10920

```
TShapeB1 histogram (optimized mesh - MAX): (log10 plot)
(    0-0.292) |============================================2283
(0.292-0.584) |=============================================3457
(0.584-0.876) |=============================================3157
(0.876-1.17 ) |==============================862
( 1.17-1.46 ) |==============================789
( 1.46-1.75 ) |==============================253
( 1.75-2.04 ) |===============74
( 2.04-2.34 ) |6
( 2.34-2.63 ) |8
( 2.63-2.92 ) |===============22
( 2.92-3.21 ) |3
( 3.21-3.5  ) |0
(  3.5-3.8  ) |6
(  3.8-4.09 ) |0
( 4.09-4.38 ) |0
( 4.38-4.67 ) |0
( 4.67-4.96 ) |0
( 4.96-5.26 ) |0
( 5.26-5.55 ) |0
( 5.55-5.84 ) |0
( 5.84-6.13 ) |0
( 6.13-6.42 ) |0
( 6.42-6.72 ) |0
( 6.72-7.01 ) |0
```

Figure 25. Common-scale histograms for the two optimal mesh produced using the MAX template in Experiment 11, using μ_2.

```
TShapeB1 histogram (optimized mesh - AVE): (log10 plot)
(    0-0.292) |============================================4963
(0.292-0.584) |================================================3709
(0.584-0.876) |=============================================1385
(0.876-1.17 ) |==============================478
( 1.17-1.46 ) |==============================181
( 1.46-1.75 ) |===============88
( 1.75-2.04 ) |===============35
( 2.04-2.34 ) |===============30
( 2.34-2.63 ) |===============20
( 2.63-2.92 ) |7
( 2.92-3.21 ) |5
( 3.21-3.5  ) |3
(  3.5-3.8  ) |4
(  3.8-4.09 ) |5
( 4.09-4.38 ) |1
( 4.38-4.67 ) |1
( 4.67-4.96 ) |1
( 4.96-5.26 ) |0
( 5.26-5.55 ) |2
( 5.55-5.84 ) |0
( 5.84-6.13 ) |1
( 6.13-6.42 ) |0
( 6.42-6.72 ) |1
( 6.72-7.01 ) |0
```

Figure 26. Common-scale histograms for the optimal mesh produced using the AVE template in Experiment 11, using μ_2.

```
TShapeB1 histogram (initial mesh): (log10 plot)
(   0-4.58) |=============================================10904
(4.58-9.16) |===========11
(9.16-13.7) |3
(13.7-18.3) |0
(18.3-22.9) |0
(22.9-27.5) |0
(27.5-32.1) |0
(32.1-36.6) |0
(36.6-41.2) |0
(41.2-45.8) |0
(45.8-50.4) |0
(50.4-55  ) |0
(  55-59.5) |0
(59.5-64.1) |0
(64.1-68.7) |0
(68.7-73.3) |0
(73.3-77.9) |0
(77.9-82.4) |0
(82.4-87  ) |0
(  87-91.6) |0
(91.6-96.2) |1
(96.2-101 ) |0
( 101-105 ) |0
( 105-110 ) |1
```

Figure 27. Common-scale histograms for the three meshes in Experiment 11: (Top) Initial Mesh; (Middle) Opt with MAX; (Bottom) Opt with AVE, using μ_2.

Extending Mesh Min-max to Quadrilateral Elements

The mesh min-max method is trivially extensible to quadrilateral (and hexahedral) elements using the Target-paradigms' sample point concept. A major difference in TMOP between triangle and quadrilateral elements is that triangle quality is usually measured using a single sample point (at the element centroid), whereas quadrilateral quality is measured using four sample points within the element, one at each corner. In OptMS, min-max was formulated on triangles only. In the OptMS users manual, angles were measured at each corner of each triangular element and the maximum was taken over all of these angles. Thus, the OptMS objective function in this case was

$$\tilde{F}_\infty = \max_{i,j} \{\theta_{ij}\} \tag{16}$$

where θ_{ij} is the j^{th} included angle within the i^{th} element of the patch. The index j, of course, runs from 1, 2, 3. This objective function can also be written as

$$\begin{aligned} \tilde{F}_\infty &= \max_i \max_{j=1,2,3} \{\theta_{ij}\} && (17) \\ &= \max_i \tilde{Q}_i && (18) \end{aligned}$$

where $\tilde{Q}_i$ is the quality of the i^{th} element. In the Target-paradigm, we replace θ_{ij} with an arbitrary Target-metric μ_{ij} and j now becomes the index over the sample points within an element. In the case of quadrilateral elements with 4 sample points we have

$$\begin{aligned} F_\infty &= \max_{i,j} \{\mu_{ij}\} && (19) \\ &= \max_i \max_{j=1,2,3,4} \{\mu_{ij}\} && (20) \\ &= \max_i Q_i && (21) \end{aligned}$$

The first definition (equation 19) is essentially the non-hierarchical objective function template H_∞ defined in Section 2; this template focuses on the worst quality *sample point* in the mesh. The third definition (equation 21) is the hierarchical objective function template H_{pq}, with $p = q = \infty$. The two, of course, are equivalent. The two would not be equivalent, however, if Q_i is defined in terms of the *average* template. In the latter case, we would be using the $H_{\infty,1}$ template. Experiments 8-10a use the H_∞ template for the generalization to quadrilateral elements. The $H_{\infty,1}$ template, which focuses on improving the worst quality element in terms of the average quality of its sample points, is considered in Experiment 10b.

Numerical Experiment 8 Results

The set-up in Experiment 8 is the same as in Experiment 1 except the hexagonal patch of triangles in Experiment 1 is replaced by a unit square patch of *quadrilateral* elements containing one free-vertex. The symmetry of the domain allows the objective function to be zero at the optimum.

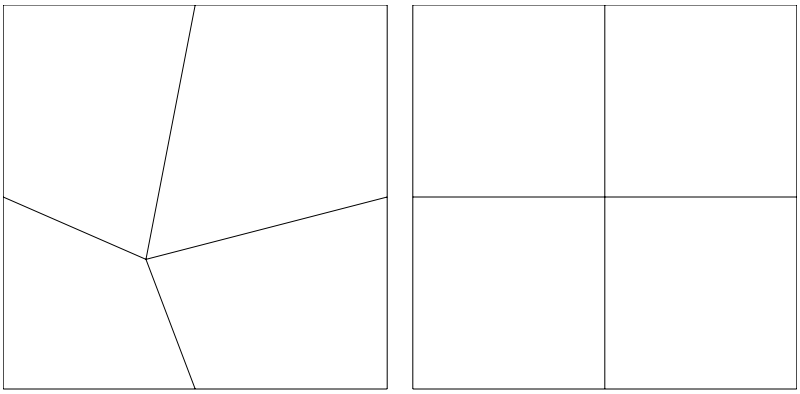

Figure 28. Mesh Patch in Experiment 8: (Left) Initial (Right) Optimized

Element Quality:
Initial Mesh μ_1: min=5.72, ave=20.4, max=31.8
Optimized Mesh μ_1: min=0, ave=0, max=0

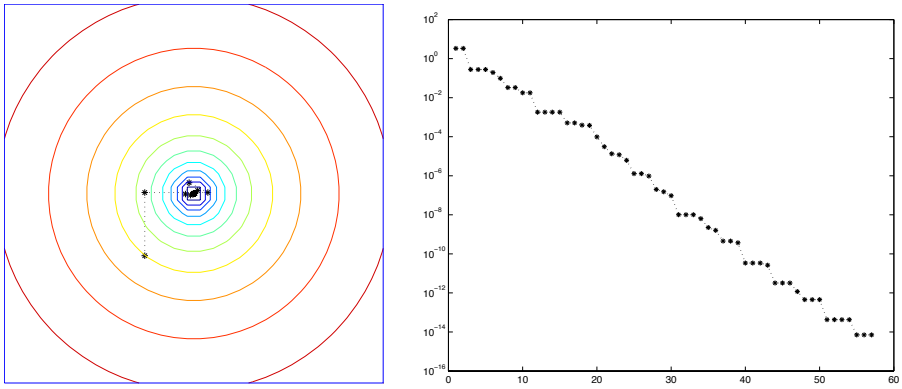

Figure 29. Experiment 8 (Left) Objective Function Contour Plot and Optimization Path (Right) Objective Function Value vs. Inner Iteration Count.
The minimum value, $\approx 10^{-14}$ (i.e., machine zero), was attained after 57 inner iterations.

Numerical Experiment 8a Results

The experimental set-up in Experiment 8a is the same as in Expereiment 8 except that a quadrilateral mesh patch with one free vertex is used on the non-convex horseshoe domain. The two lower quadrilaterals in the initial mesh patch in Figure 30 (Left) are inverted because the free vertex lies outside the domain. The value of the objective function for the initial mesh is 89.0. Level Set contours for this Experiment are shown in Fgure 31. Because we use the non-barrier shape metric μ_1, the metric domain is the whole plane. Never-the-less, the optimal point lies inside the feasible region and, because the Non-gradient solver is able to find this optimal point, the numerically optimized mesh (Right side of Figure 30) is non-inverted. The Target-matrix in this experiment specified that the ideal quadrilateral is a square. We see that, at the optimal point, none of the quadrilaterals are squares. This is not the fault of the metric or the solver, rather it is due to the fact that in this particular problem, a solution in which all four quadrilaterals are squares does not exist. This is also reflected in the non-zero value of the objective function (64.0). The convergence history in the Right part of Figure 31 shows the height $\log_{10}(OF_k - OF_{final})$, with $k = 1, 2 \ldots, k_{final}$ and OF_k being the value of the objective function at the k^{th} iteration. In this problem, 35 iterations are needed to decrease the height by four orders-of-magnitude. This experiment shows that the Maximum template applied to the non-barrier shape metric μ_1, with Target-matrix equal to the identity, along with the numerical Non-gradient solver, can find the correct optimal position for a single free-vertex non-convex patch of quadrilateral elements.

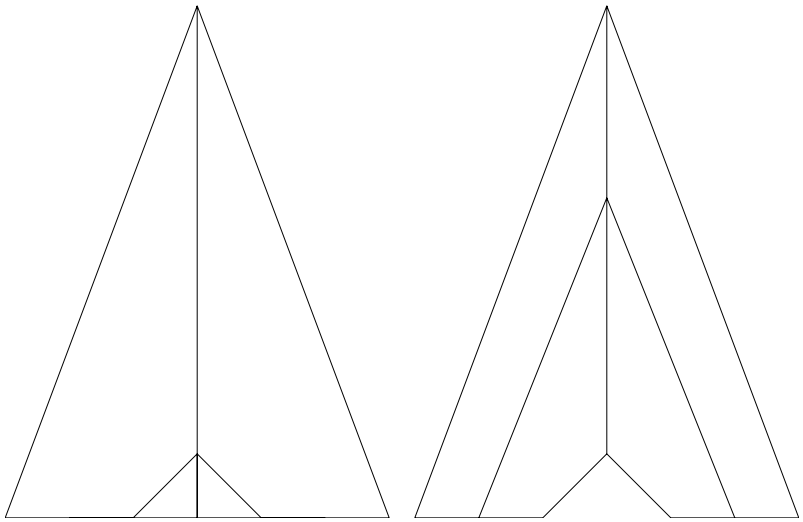

Figure 30. Mesh Patch in Experiment 8a: (Left) Initial (Right) Optimized w/MAX

Element Quality:
Initial Mesh μ_1: min=9.00, ave=49.0, max=89.0
Optimized Mesh μ_1: min=29.0, ave=46.5, max=64.0

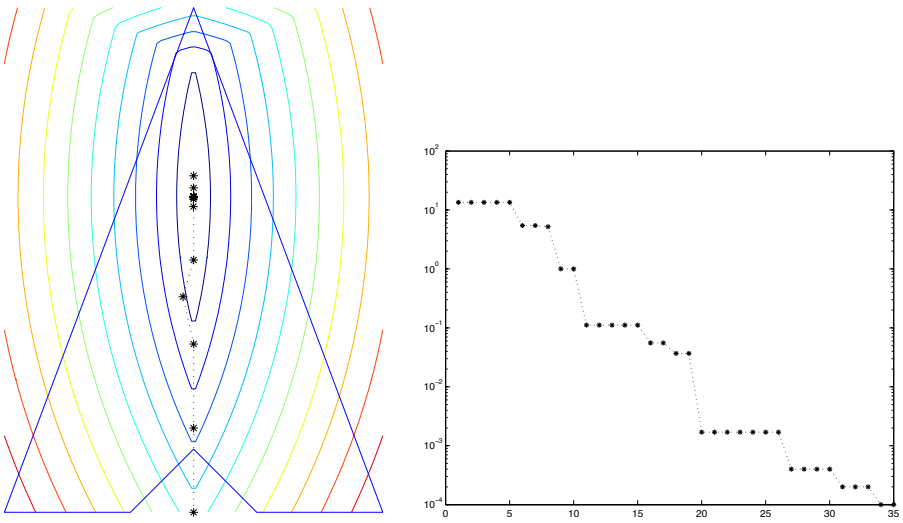

Figure 31. Experiment 8a (Left) Objective Function Contour Plot and Optimization Path (Right) Objective Function Value vs. Inner Iteration Count.
The minimum value, $\approx 10^{-4}$, was attained after 34 inner iterations.

Numerical Experiment 8b Results

The set-up in Experiment 8b is the same as in Experiment 8a except that the Hölder Mean template with $p = 1$ (AVE) is used instead of the Maximum template. The initial mesh in 8b is the same as in 8a (i.e., inverted). The level set contours in 8b are shown in Figure 33. Comparing to the contours in 8a, we see that (a) the contours in 8b are smooth and (b) the optimal point in 8b is below the optimal point in 8a, yet remains in the feasible region. The numerical optimization Non-gradient solver is able to find the correct minimum point, giving the non-inverted mesh in Figure 32. The value of the objective function decreased from 60.3 on the initial mesh to 38.6 on the optimized mesh. The mesh is *converged* in 16 outer iterations, compared to the 34 that were needed when using the Maximum template. This experiment shows that the location of the optimal point depends, among other factors, on the choice of template. In this experiment, both templates result in a non-inverted optimal mesh.

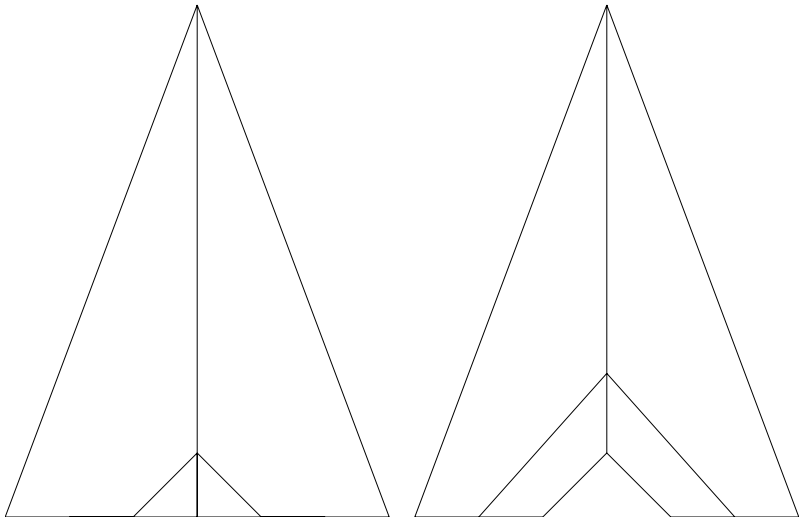

Figure 32. Mesh Patch in Experiment 8b: (Left) Initial (Right) Optimized w/AVE

Element Quality:
Initial Mesh μ_1: min=3.75, ave=28.4, max=60.3
Optimized Mesh μ_1: min=3.31, ave=20.9, max=38.6

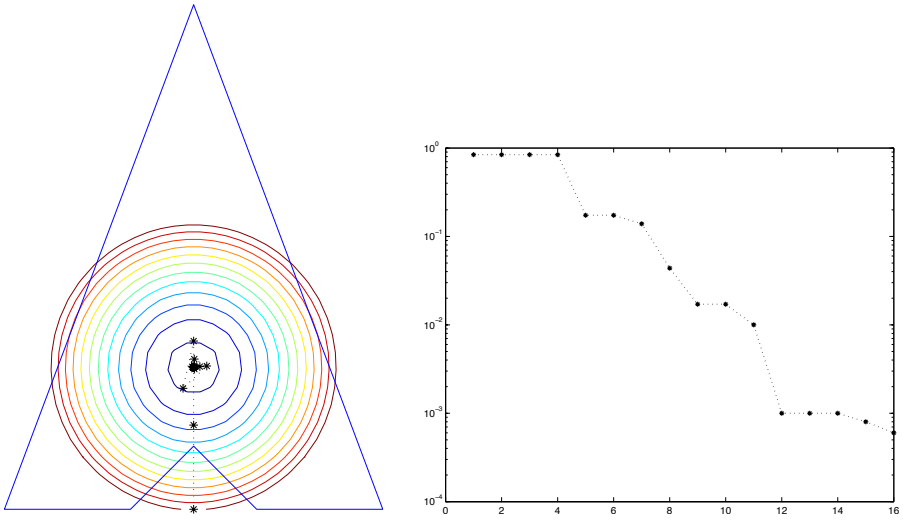

Figure 33. Experiment 8b (Left) Objective Function Contour Plot and Optimization Path (Right) Objective Function Value vs. Inner Iteration Count.
The minimum value, $\approx 6 \times 10^{-4}$, was attained after 16 inner iterations.

Numerical Experiment 9 Results

The Experiment 9 set-up is that same as Experiment 8a except that the barrier shape metric μ_2 is used. Since the domain of the barrier metric is the same as the feasible region of the horseshoe patch, the initial position of the free vertex is taken to be a point in the feasible region (see Figure 34, Left). Level Sets for this experiment are shown in Figure 35. The optimal point also lies within the feasible region. The optimized mesh, shown on the Right side of Figure 34 shows that the Non-gradient solver stops just short of the true minimum point; this can be seen by observed the the computed optimal point is just to the left of the vertical line of symmetry in the patch. Upon closer examination, it appears that the objective function is flat, or nearly flat, close to the optimal location. The right side of Figure 35 shows that the solver stagnates quickly once the flat region is reached. Never-the-less, the objective function was decreased significantly, going from 121 to 3.63.

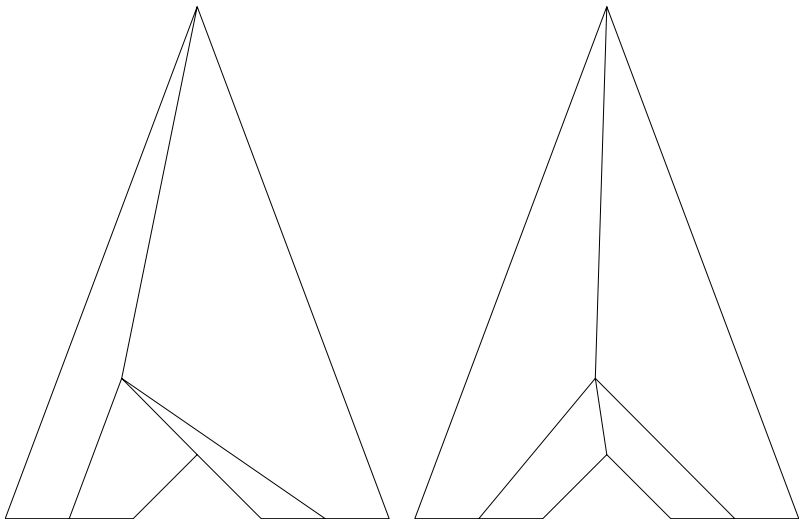

Figure 34. Mesh Patch in Experiment 9: (Left) Initial (Right) Optimized

Element Quality:
Initial Mesh μ_2: min=0.5, ave=63.7, max=121
Optimized Mesh μ_2: min=1.07, ave=2.46, max=3.63

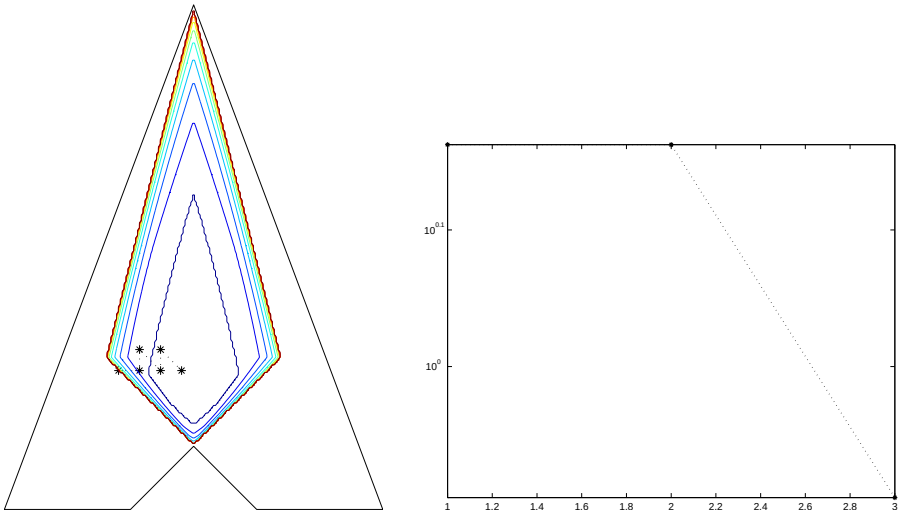

Figure 35. Experiment 9 (Left) Objective Function Contour Plot and Optimization Path (Right) Objective Function Value vs. Inner Iteration Count.
The minimum value, 10^{-1}, was attained after just 3 inner iterations.

Numerical Experiment 9a Results

The set-up in Experiment 9a is the same as in Experiment 9 except that the Hölder Mean template with $p = 1$ (AVE) is used instead of the Maximum template. The initial mesh is the same as in Experiment 9. The level sets in Experiment 9a are shown in Figure 37 and are different from those in experiment 9 due to the use of the Average template instead of the Maximum. The contours in 9a are smoother and the objective function is not as flat as they are in 9. As a result, the numerical optimization solver is able to locate the optimal point more accurately in 9a. This can also be seen in comparing Figures 36 to 34 - the computed optimum in the former lies closer to the vertical symmetry line in this problem.

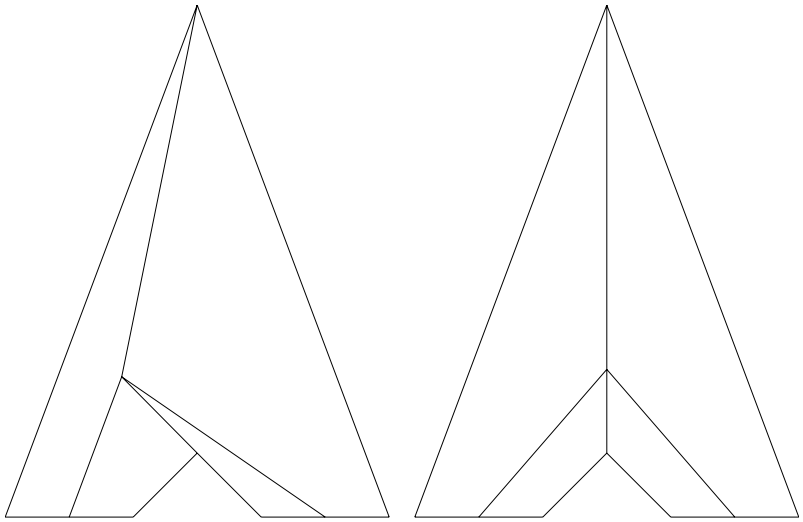

Figure 36. Mesh Patch in Experiment 9a: (Left) Initial (Right) Optimized

Element Quality:
Initial Mesh μ_2: min=0.289, ave=17.4, max=62.8
Optimized Mesh μ_2: min=0.816, ave=1.38, max=1.95

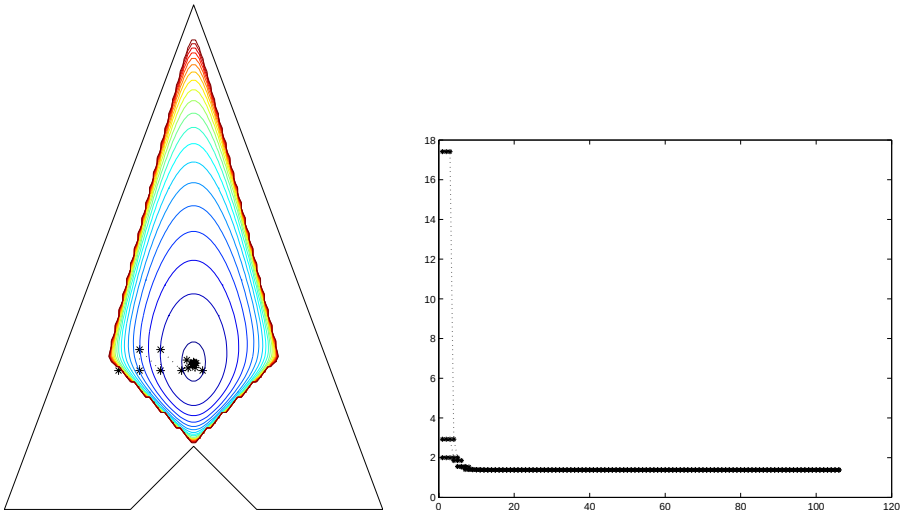

Figure 37. Experiment 9a (Left) Objective Function Contour Plot and Optimization Path (Right) Objective Function Value vs. Inner Iteration Count.
The minimum value, 10^{-1}, was attained after just 3 inner iterations.

Numerical Experiment 10 Results

The set-up in Experiment 10 is the same as Experiment 1 except we use an unstructured quadrilateral mesh with multiple vertices. Free vertices in the mesh have valence 3, 4, or 5. The initial mesh is shown in Figure 38 and six optimized meshes are shown in Figures 39, 40, and 41. The six meshes correspond to the possible combinations of MAX vs. AVE and μ_1 vs. μ_2 vs. μ_3.

As the left portion of Figure 39 shows, the combination MAX and μ_1 produced an optimized mesh with highly unsatisfactory quality. Even after 99 outer iterations, the result did not appear to be fully converged. In contrast, a good quality mesh (shown in the right portion of the same figure) was produced with the combination AVE and μ_1.

The optimized mesh on the left of Figure 40 was the result of the combination MAX and μ_2. This mesh is better than the one on the left of Figure 39, but worse than the one in the right of this figure (because it is not as smooth). The barrier metric μ_2 helps prevent the collapse of the mesh edges that we see with the non-barrier μ_1 when the MAX template is used. The mesh shown on the right of Figure 40 was AVE with μ_2. Once again the AVE template produced a better-looking mesh than MAX.

Next, we optimized the mesh using μ_3. The value selected for λ, based on the domain area and the number of elements, was 0.050, and was the same at all sample points. The result for the MAX template is shown in the left of Figure 41. The result was ugly. In contrast, the AVE template with μ_3 and $\lambda = 0.050$ (right portion of Figure 41) is again a nice-looking mesh.[7]

This set of six tests suggests that, except for meshes having a high degree of symmetry, such as those in Experiments 8, 8a, and 9, it is difficult to achieve good mesh quality using the MAX (non-hierarchical H_∞) template in the case of quadrilateral elements. To confirm this, we tried several other quadrilateral meshes, and again found poor quality, mostly in the form of non-smoothness and/or lack of convergenvce, with the MAX template. This result strongly contrasts with triangle meshes, in which the MAX template does give good quality (and where smoothness is not expected).

Next, we consider common-scale histograms for the meshes in Experiment 10, *based on the barrier metric* μ_2. We focus on the three meshes shown in Figures 38 (initial mesh), 40 (Left: result of optimization of μ_2 with MAX OF template), and 40 (Right: result of optimization of μ_2 with AVE OF template). For convenience, let us call these meshes 38, 40L, and 40R. The quality of these three meshes was measured in terms of element quality. Element quality was defined two ways: (a) by computing the *maximum* value of μ_2 over the four sample points within the element, or, (b) by computing the *average* value of μ_2 over the four sample points within the element. There are thus six combinations (and six histograms) to consider (three meshes times the two cases).

[7]We also tried $\lambda = 0.1826$, and found that the optimized mesh for MAX template and μ_3 had much better quality than with $\lambda = 0.050$.

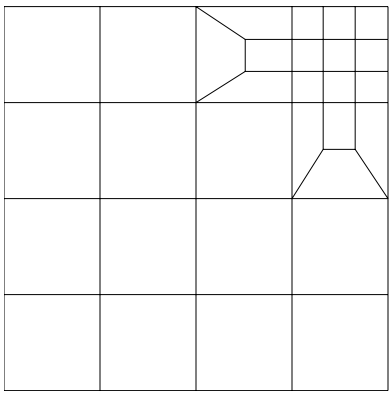

Figure 38. Initial Mesh Patch in Experiment 10.

Figure 42 shows three common-scale histograms for meshes 38, 40L, and 40R using the *maximum* value of μ_2 within an element (Case a). The quality of the initial mesh is good since 24 quadrilateral elements are squares and there are no inverted elements. On the other hand, the histogram also shows six outliers corresponding to the transition quadrilaterals. The element quality statistics for the initial mesh (in terms of the max) are: min = 0, ave = 0.1755, max = 1.073, and std = 0.3668. The histogram for mesh 40L (optimized using MAX OF template) shows that the four worst outliers in the initial mesh now have improved element quality values. The element quality statistics for mesh 40L (in terms of the max) are: min = 0.0012, ave = 0.2382, max = 0.6309, and std = 0.2499. Mesh 40L also has no inverted elements. Finally, the histogram of mesh 40R (optimized using AVE OF template) shows that there are still two outliers whose element quality is similar to the worst quality element in the initial mesh. However, mesh 40R also has as many near-perfect elements as the initial mesh. The element quality statistics for mesh 40R (in terms of the max) are: min = 0, ave = 0.1361, max = 0.7609, and std = 0.2358. Mesh 40R has no inverted elements. The quality statistics and histograms in Figure 42 inform us that

1. Mesh 38 (initial) has the largest values of maximum element quality and the largest value of standard deviation.

2. Mesh 40L (opt MAX OF template) has the largest value of average element quality and the smallest value of maximum quality (as expected).

3. Mesh 40R (opt AVE OF template) has the smallest value of average element quality (as expected), and has better max-quality and standard deviation values than the initial mesh. It also has the smallest value of standard deviation among the three meshes.

In terms of sample point quality (based on MAX(μ_2)), mesh 40L minimizes the maximum quality, while 40R also decreases it significantly compared to the initial mesh. Mesh 40R minimizes the average quality, while 40L increases the average quality compared to the initial

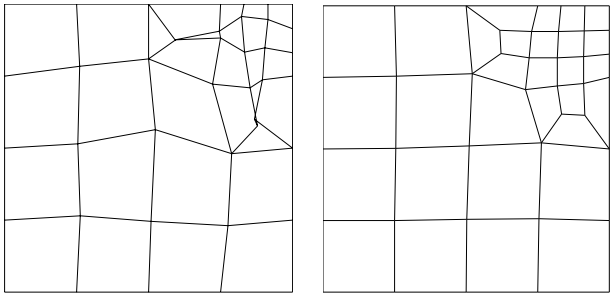

Figure 39. Optimized Mesh Patches on Square in Experiment 10: (Left) MAX with μ_1 (Right) AVE with μ_1

mesh. Inter-element smoothness is greater in 40R, compared to 40L. It is also interesting to note that mesh 40R has only two elements in the worst quality histogram *bin*, while 40L has seven elements in that same bin.

Figure 43 shows three common-scale histograms for meshes 38, 40L, and 40R using the *average* value of μ_2 within an element (Case b). The quality of the initial mesh is good since 24 quadrilateral elements are squares and there are no inverted elements. On the other hand, the histogram also shows six outliers corresponding to the transition quadrilaterals. The element quality statistics for the initial mesh (in terms of the ave) are: min = 0, ave = 0.1201, max = 0.6801, and std = 0.2383. The histogram for mesh 40L (optimized using MAX OF template) shows that the four worst outliers in the initial mesh now have improved quality values. The element quality statistics for mesh 40L (in terms of the average over the sample points within the element) are: min = 0.0006, ave = 0.1530, max = 0.5808, and std = 0.1737. Mesh 40L has no inverted elements. Finally, the histogram of mesh 40R (optimized using AVE OF template) shows that the remaining outliers in this case have average-quality values smaller than the outliers in both meshes 38 and 40L. In addition, mesh 40R has as many near-perfect elements as the initial mesh. The element quality statistics for mesh 40R (in terms of the ave) are: min = 0, ave = 0.0894, max = 0.4785, and std = 0.1630. Mesh 40R has no inverted elements. The quality statistics and histograms in Figure 43 inform us that

1. Mesh 38 (initial) has the largest value of average element quality and largest standard deviation.

2. Mesh 40L (opt MAX OF template) has the largest value of average element quality, but not the smallest value of maximum average-quality (NOT expected).

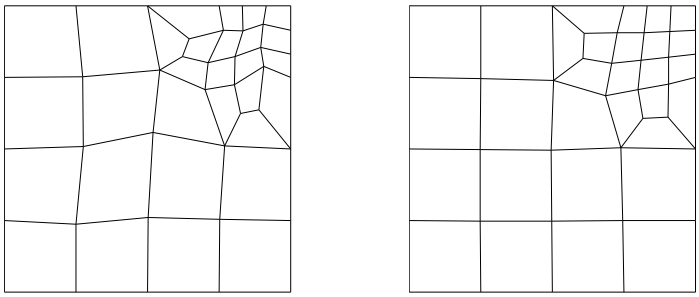

Figure 40. Optimized Mesh Patches on Square in Experiment 10: (Left) MAX with μ_2 (Right) AVE with μ_2

3. Mesh 40R (opt AVE OF template) has the smallest value of average element quality (as expected), the smallest value of maximum average-quality, and has the smallest value of the standard deviation.

In terms of element quality (based on AVE(μ_2)), mesh 40R is best. Inter-element smoothness is greater in 40R, compared to 40L.

Measuring quality in terms of the average over the sample points within the element (as opposed to measuring it in terms of the maximum) suggests that different meshes can be considered optimal, depending on whether one thinks it is more important to focus on element quality or to focus on sample point quality.

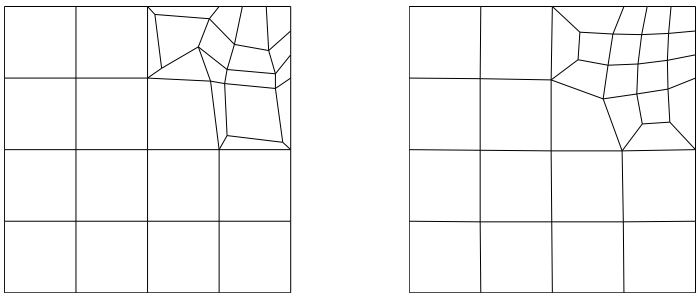

Figure 41. Optimized Mesh Patches on Square in Experiment 10: (Left) MAX with μ_3 (Right) AVE with μ_3

```
ElementMaxQM(TShapeB1) histogram (initial mesh)
(  0-0.2) |==================================================24
(0.2-0.4) |0
(0.4-0.6) |2
(0.6-0.8) |0
(0.8-1  ) |0
(  1-1.2) |4
(1.2-1.4) |0
(1.4-1.6) |0
(1.6-1.8) |0
(1.8-2  ) |0

ElementMaxQM(TShapeB1) histogram (optimal mesh - MAX OF template)
(  0-0.2) |==================================================18
(0.2-0.4) |4
(0.4-0.6) |1
(0.6-0.8) |7
(0.8-1  ) |0
(  1-1.2) |0
(1.2-1.4) |0
(1.4-1.6) |0
(1.6-1.8) |0
(1.8-2  ) |0

ElementMaxQM(TShapeB1) histogram (optimal mesh - AVE OF tempate)
(  0-0.2) |==================================================24
(0.2-0.4) |0
(0.4-0.6) |4
(0.6-0.8) |2
(0.8-1  ) |0
(  1-1.2) |0
(1.2-1.4) |0
(1.4-1.6) |0
(1.6-1.8) |0
(1.8-2  ) |0
```

Figure 42. Common-scale histograms for the three meshes in Experiment 10 with
Element Quality = MAX(μ_2) over the Sample Points.
(Top) Initial Mesh; (Middle) Opt with MAX OF template;
(Bottom) Opt with AVE OF template.

```
ElementPMeanP(TShapeB1) histogram (initial mesh):
(   0-0.07) |================================================24
(0.07-0.14) |0
(0.14-0.21) |0
(0.21-0.28) |0
(0.28-0.35) |0
(0.35-0.42) |====2
(0.42-0.49) |0
(0.49-0.56) |0
(0.56-0.63) |0
(0.63-0.7 ) |========4

ElementPMeanP(TShapeB1) histogram (optimal mesh - MAX OF template)
(   0-0.07) |==================================18
(0.07-0.14) |====2
(0.14-0.21) |====2
(0.21-0.28) |==1
(0.28-0.35) |==1
(0.35-0.42) |========4
(0.42-0.49) |0
(0.49-0.56) |====2
(0.56-0.63) |0
(0.63-0.7 ) |0

ElementPMeanP(TShapeB1) histogram (optimal mesh - AVE OF template)
(   0-0.07) |================================================24
(0.07-0.14) |0
(0.14-0.21) |====2
(0.21-0.28) |0
(0.28-0.35) |0
(0.35-0.42) |0
(0.42-0.49) |========4
(0.49-0.56) |0
(0.56-0.63) |0
(0.63-0.7 ) |0
```

Figure 43. Common-scale histograms for the three meshes in Experiment10 with Element Quality = AVE(μ_2) over the Sample Points. (Top) Initial Mesh; (Middle) Opt with MAX OF template; (Bottom) Opt with AVE OF template.

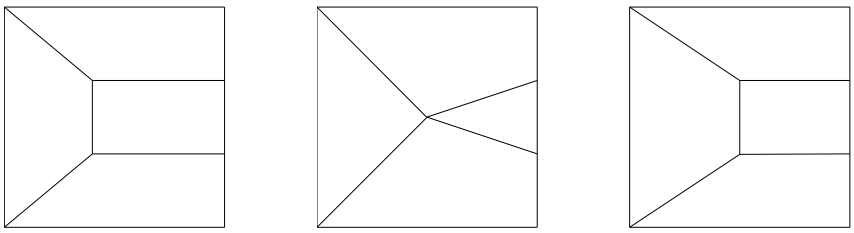

Figure 44. Mesh Plots for Experiment 10a with μ_1: (Left) Initial Mesh (Middle) Optimized w/MAX (Right) Optimized w/AVE

Numerical Experiment 10a Results

The set-up in Experiment 10a is the same as in Experiment 10 except that we use a smaller quadrilateral mesh that is similar to the larger one (see Figure 44). The purpose is to further investigate the problem discovered in Experiment 10 by using a simpler mesh. The mesh in this experiment has only two free vertices and four quadrilateral elements. The free vertices are tri-valent.

As Figure 44 shows, the optimized mesh (MAX with μ_1) has an edge whose length is zero and thus this optimization was quite detrimental to quality even though it minimizes the maximum value of the non-barrier shape metric. In contrast, the mesh optimized using AVE with μ_1 has good quality.

Figure 45 shows a pair of contour plots corresponding to the μ_1 metric, the MAX template, and the mesh in Figure 44. The sub-plot on the left shows the initial mesh and the contours for the free vertex within the patch consisting of the top, right, and leftmost quadrilaterals. The optimal location for this free vertex lies to the southeast of its current position. The sub-plot on the right shows the mesh after the upper free vertex has been moved to its optimal location within its patch. It also shows the contours for the lower free vertex within its patch, consisting of the bottom, right, and leftmost quadrilaterals. The optimal position for the bottom free vertex is thus nearly on top of the location for the upper free vertex, rather reminiscent of what was observed in Experiment 10. A similar set of plots is given in Figure 46, except for the AVE template. In this case, the optimal locations of the free mesh vertices do not coincide.

The conclusion for Experiment 10a is that the extension of mesh min-max to quadrilateral elements proposed in equations 19-21 leaves a lot to be desired.

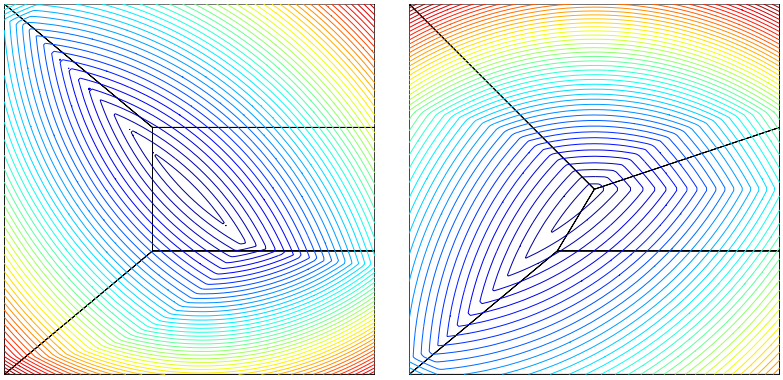

Figure 45. Contour Plots for Experiment 10a with MAX template and μ_1: (Left) Top patch (Right) Bottom patch after top vertex is moved to optimum

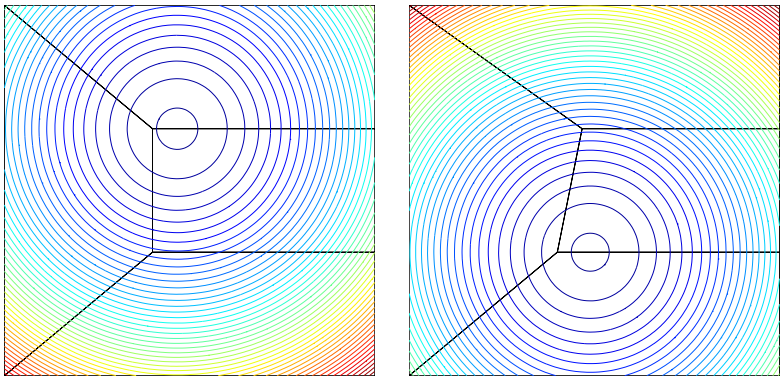

Figure 46. Contour Plots for Experiment 10a with AVE template and μ_1: (Left) Top patch (Right) Bottom patch after top vertex is moved to optimum

Alternate extensions of Min-max to Quadrilateral Mesh Elements

We saw in section 5.5 that the natural extension of min-max to quadrilateral elements, given by equations 19-21, with

$$Q_i = \max_{j=1,2,3,4} \mu_{ij} \tag{22}$$

did not produce satisfactory results. In this section we consider some alternate objective functions, based on the H_{pq} template, which are similar to the mesh min-max method which can potentially give better results without sacrificing too much of the original idea of improving worst quality.

The alternative objective functions approximate F_∞ in equation 19-21 via the H_{pq} template, giving

$$F_{p,q} = \frac{1}{N} \sum_{i=1}^{N} Q_i^p \tag{23}$$

with

$$Q_i = \left(\frac{1}{4} \sum_{j=1}^{4} \mu_{ij}^q \right)^{1/q} \tag{24}$$

To obtain an indication as to whether this alternative is viable, we explore two cases of interest: (i) $p = \infty$, $q = 1$ and (ii) $p = q = 4$. Explicitly, the two cases are

(i) Improve the worst quality *element*, as opposed to the worst quality sample point:

$$F_{\infty,1} = \max_i Q_i \tag{25}$$

with

$$Q_i = \frac{1}{4} \sum_{j=1}^{4} \mu_{ij} \tag{26}$$

(ii) Approximate F_∞ using 'large' p and q:

$$F_{4,4} = \frac{1}{N} \sum_{i=1}^{N} Q_i^4 \tag{27}$$

with

$$Q_i = \left(\frac{1}{4} \sum_{j=1}^{4} \mu_{ij}^4 \right)^{1/4} \tag{28}$$

To investigate these possibilities, we begin by creating contour plots for the upper free vertex in Experiment 10a, using the alternate objective functions. Plots for the two alternatives, along with the contour plots for the original MAX and AVE objective functions are shown in Figure 47. We see that the two alternatives are intermediate cases between MAX and AVE, and that the optimal position in the alternatives is intermediate between the optimal positions in MAX and AVE. We thus expect optimization behavior that is intermediate between the MAX and AVE results. Figure 48 shows the meshes that result from the two alternative objective functions, along with those created in Experiment 10a. Alternatives (i) and (ii) clearly produce meshes that are intermediate between those produced by the MAX and the AVE.

Experiment 10 was repeated with the two alternate objective functions as well (see results for μ_1 in Fgure 49 and for μ_2 in Figure 50). Unfortunately, none of the three meshes in the bottom row of the μ_1 figure is very satisfactory although they are (to varying degress) an improvement over MAX(MAX). The barrier metric gave somewhat better meshes with the alternative objective functions. In that case, small edge-lengths are prevented, although the meshes are still somewhat non-smooth. Basically, one should expect unstructured quadrilateral meshes to appear non-smooth if the MAX template is part of the objective function. Indeed, this is confirmed in Figure 51 where we show the results of the four objective functions with the metric μ_2 on a structured quadrilateral mesh.

In summary, results in this section lead us to conclude that the MAX(MAX) template is acceptable to use in optimizing quadrilateral meshes provided it is used in conjunction with a barrier metric (to discourage small edges) and if mesh *smoothness* is not important.

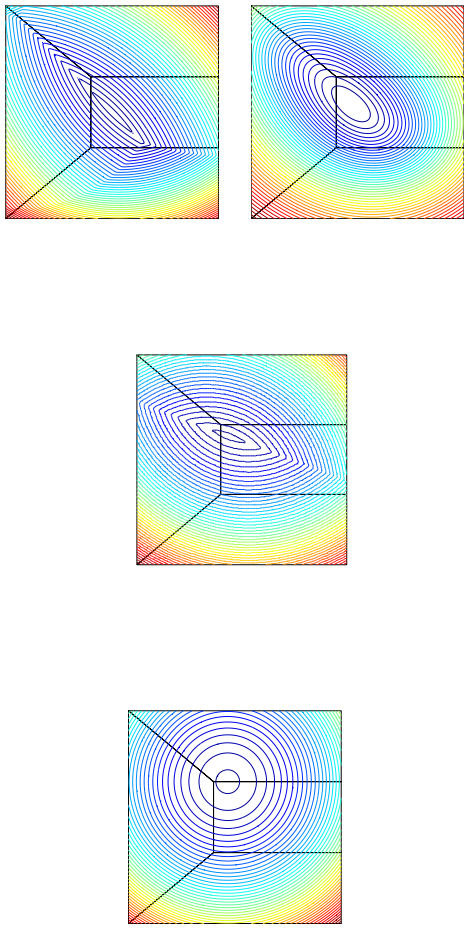

Figure 47. Contour Plots for top patch in Experiment 10a with Alternate Objective Functions (μ_1): (Top Row) Left - F_∞, i.e., MAX; Right - $F_{4,4}$; (Middle Row) $F_{\infty,1}$; (Bottom Row) Left - $F_{1,1}$, i.e., AVE

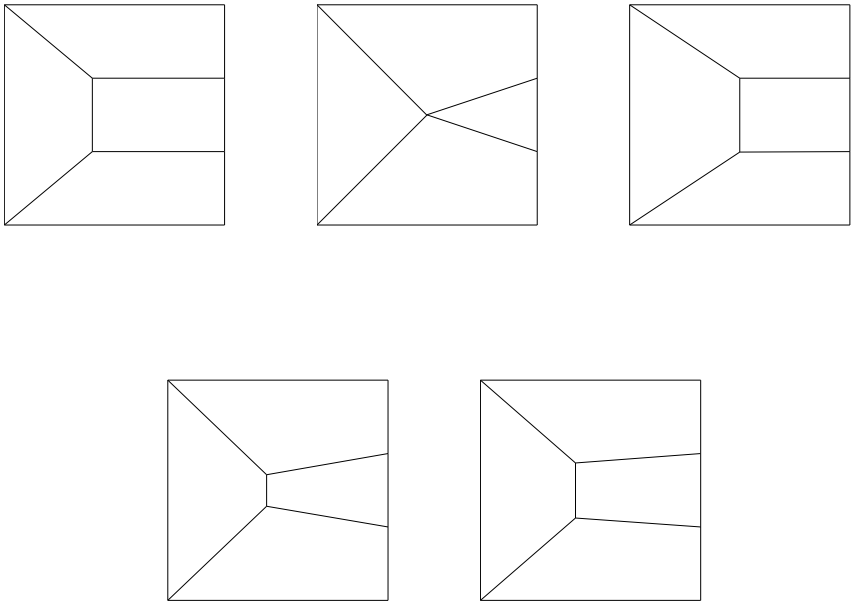

Figure 48. Mesh Plots for Experiment 10a with μ_1.
Top Row: (Left) Initial Mesh (Middle) Optimized w/MAX (Right) Optimized w/AVE
Bottom Row: (Left) Optimized w/AVE4 (Right) Optimized w/MAX(AVE)

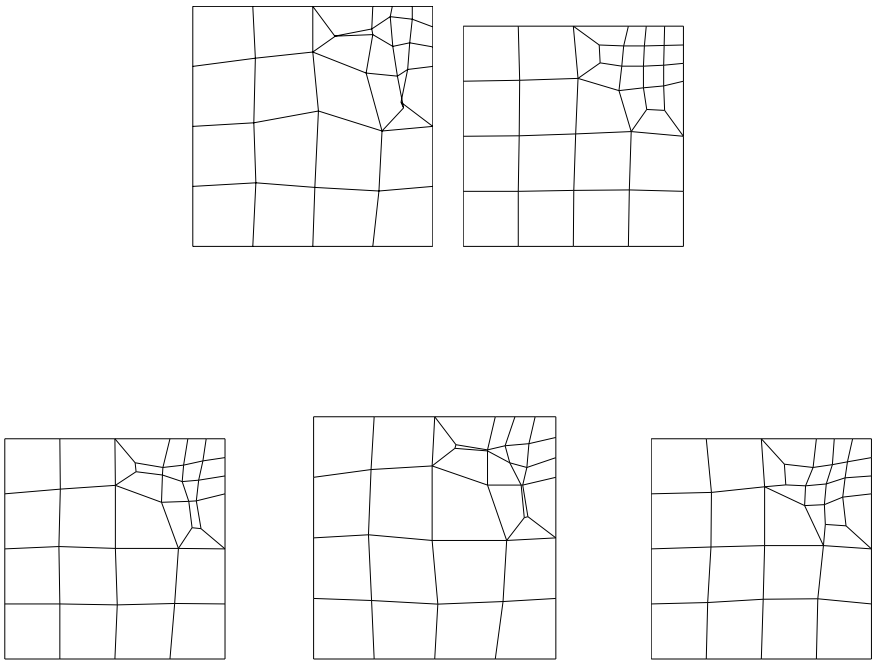

Figure 49. Optimized Mesh Patches on Square in Experiment 10 with μ_1.
Top Row: (Left) MAX(MAX) (Right) AVE(AVE)
Bottom Row: (Left) AVE2(AVE2); (Middle) AVE4(AVE4); (Right) MAX(AVE)

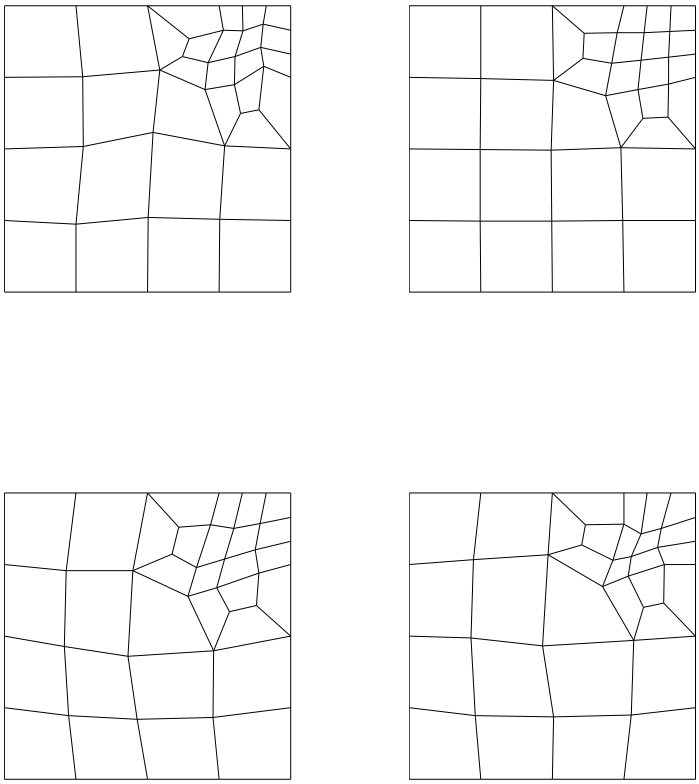

Figure 50. Optimized Mesh Patches on Square in Experiment 10 with μ_2.
Top Row: (Left) MAX(MAX); (Right) AVE(AVE)
Bottom Row: (Left) AVE2(AVE2); (Right) MAX(AVE)

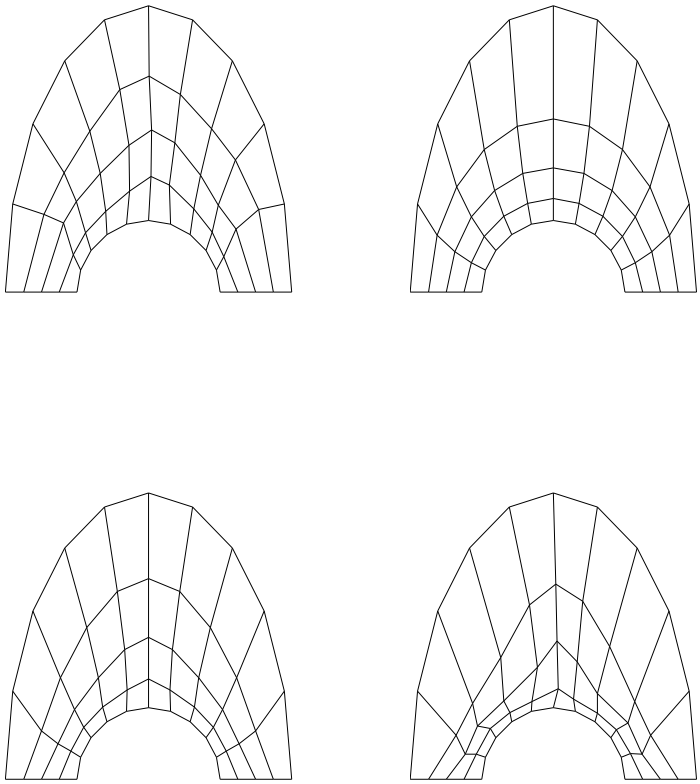

Figure 51. Optimized Mesh Patches on Horseshoe in Experiment 10 with μ_2.
Top Row: (Left) MAX(MAX); (Right) AVE(AVE)
Bottom Row: (Left) AVE2(AVE2); (Right) MAX(AVE)

Summary, Conclusions, and Future Work

A major purpose of this paper was to demonstrate both theoretically and computationally that the Target-Matrix Optimization Paradigm provides an avenue for naturally extending Diachin's min-max mesh optimization algorithm to mesh quality metrics that are based on user-supplied definitions of quality. The paradigm also readily allows extension to non-simplicial elements through the sample point construct. The result is that TMOP/min-max can potentially address a wider range of mesh optimization problems within the context of improving worst-case mesh quality.

First, let us compare the TMOP and OptMS min-max methods. The TMOP min-max method, as given in this paper, has some important features in common with the min-max mesh optimization reported in OptMS [8]. Both, of course, use the Maximum Template. In addition, both optimize quality on local patches consisting of one free vertex and both use the patch sweeping algorithm described in Section 2 to optimize the global mesh.

Thre are some important differences between OptMS and the TMOP min-max extension. The former method used simplicial element quality metrics such as (1) the maximum angle in the element, (2) the minimum angle in the element, and (3) element aspect ratio (shape). The metrics in the TMOP extension of min-max were (4) non-barrier shape, (5) barrier shape, and (6) non-barrier shape+size. The TMOP metrics differ from those in OptMS in three respects (a) they have somewhat different goals (e.g., angle vs. shape, shape vs. shape+size), (b) TMOP uses Target-matrices to permit application-defined quality, and (c) TMOP metrics can be used in barrier or non-barrier form. The use of Target-matrices is significant because this allows the user to define problem-dependent shape or size, whereas the OptMS metrics define the ideal *a priori*. The barrier form is important in that it ensures that an initially non-inverted mesh stays that way when optimized, whether using the MAX or AVE template.

Another difference between OptMS and this TMOP extension is that different numerical optimization solvers are used. OptMS uses an Active Set solver to find the solution to the min-max problem while TMOP uses the non-derivative method known as the amoeba. This difference was not important in the research described in this paper since our emphasis was on assessing the extension of min-max, not on production-izing it. Potentially, the difference in solvers will be important if the TMOP min-max algorithm is to be used in a production setting with large meshes, where efficiency is paramount.

Because the OptMS metrics were mostly non-barrier metrics, a method within the solver was needed to prevent the mesh from inverting. The OptMS Active Set solver accomplished this by adding an explicit constraint that the free vertex must remain in the feasible region. As with the barrier method, this approach requires that the initial mesh be non-inverted. However, barrier metrics have the advantage that, while the initial mesh must be non-inverted, the optimal point is guaranteed to lie within the feasible region. The Non-gradient amoeba solver used with TMOP does not directly constrain the free vertex. Another important difference between the Active Set Solver and the Non-gradient Solver is that the

former was written with the assumption that the mesh elements are simplicial, whereas in the TMOP extension, one is not limited to such.

Our findings in this study:

1. The MAX template, which improves the worst quality, can generate inverted optimal meshes if the metric has no barrier. This was observed in Experiments 1a, 10, 10a, and 11 with the μ_1 metric. The AVE template can, of course, also create inverted meshes with non-barrier metrics. We only mention this point in case someone should think that optimizing 'worst quality' guarantees a non-inverted mesh (after all, don't inverted meshes have the worst quality?)

2. In fact, the phrase 'worst quality' is ambiguous because one can optimize using a variety of quality metrics and objective function templates. For example, should worst quality be defined in terms of angles, shape, shape+size, or being tangled? Is worst quality of *elements* more important than worst quality of *sample points*? Optimizing on one definition of worst quality does not guarantee that all 'worst qualities' are optimized.

3. The location of the optimal mesh points depends not only on the choice of quality metric, but also on the template (MAX or AVE) employed. This is true even when the same quality metric is used in both templates.

4. Using Target-metrics and Target-matrices, one can perform mesh min-max in order to create meshes having variable area across the domain.

5. Both barrier and non-barrier metrics can be used within the mesh min-max method.

6. While the extension of mesh min-max to quadrilateral element meshes is straightforward, the optimized meshes are often unsatisfactory, particularly if non-barrier metrics are used. We expect similar results for hexahedral element meshes.

7. The max(max) template can be used to optimize quadrilateral meshes provided inter-element smoothness is not important. Barrier metrics help prevent MAX(MAX) optimization from creating excessively small edge-lengths in quadrilateral meshes.

8. Optimization with the MAX template can also improve average quality. Optimization with the AVE tempate can also improve the worst (MAX) quality. In fact, both methods can often significantly improve the worst quality element. A case could even be made that it is better to optimize with the AVE template because (a) the max is often still reduced sufficiently, (b) the average quality is much better, (c) it works better with non-barrier metrics, and (d) the meshes produced with AVE are smoother. This position is especially defendable in the case of non-simplicial elements.

9. The amoeba method can be used to find optimal solutions to the mesh min-max method. Efficiency of the amoeba method in relation to the mesh min-max method was not explored.

More work needs to be done in order to tell the full story concerning mesh min-max methods. This includes the following:

- More testing of the method on tetrahedral and hexahedral meshes. Does confining MIN-MAX on hexahedral meshes to barrier metrics help create acceptable mesh quality?
- More testing of the method when non-trivial Target-matrices are used. For example, except for Experiment 7, the set of Target-matrices used in the experiments did not vary from one sample point to the next.
- Determine the efficiency of the amoeba, other non-derivative solvers, and active set methods as applied to the mesh min-max problem. Production-ization of the TMOP extension to min-max.
- Explore the idea of re-ordering the local patches in the overall algorithm (section 2) according to patch quality in order to determine uniqueness of the solution.
- Explore the possibility of formulating and solving the mesh min-max as a global optimization problem.

References

[1] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. *J. Algorithms*, pages 302–322, 1999.

[2] M. Brewer, L. Diachin, P. Knupp, and D. Melander. The mesquite mesh quality improvement toolkit. In *Proceedings of the 12th International Meshing Roundtable*, pages 239–250. Sandia National Laboratories, 2003.

[3] S. A. Canann, M. B. Stephenson, and T. Blacker. Optismoothing: An optimization-driven approach to mesh smoothing. *Finite Elements in Analysis and Design*, 13:185–190, 1993.

[4] J. Escobar, E. Rodriguez, R. Montenegro, G. Montero, and J. Gonzalez-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Comput. Meth. Appl. Mech. Engrg.*, 192:2775–2787, 2003.

[5] L. Freitag and P. Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *Int. J. Numer. Methods. Engrg.*, 53:1377–1391, 2002.

[6] L. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Numer. Methods. Engrg.*, 40:3979–4002, 1997.

[7] L. Freitag and P. Plassmann. Local optimization-based simplicial mesh untangling and improvement. *Int. J. Numer. Methods. Engrg.*, 49:109–125, 2000.

[8] Lori Freitag. Users manual for Opt-MS: Local methods for simplicial mesh smoothing and untangling. Technical Report ANL/MCS-TM-239, Argonne National Laboratory, Argonne, Ill., 1999.

[9] Bryan Matthew Klingner and Jonathan Richard Shewchuk. Agressive tetrahedral mesh improvement. In *Proceedings of the 16th International Meshing Roundtable*, pages 3–23, Seattle, Washington, October 2007.

[10] P. Knupp. Introducing the target-matrix paradigm for mesh optimization via node movement. In *Proceedings of the 19th International Meshing Roundtable*, pages 67–83. Springer, 2010.

[11] P. Knupp. Introducing the target-matrix paradigm for mesh optimization via node movement. *Engineering with Computers*, 28:419–424, 2011.

[12] P. Knupp and E. van der Zee. Convexity of mesh optimization metrics using a target-matrix paradigm. Technical Report SAND2006-4975J, Sandia National Laboratories, 2006.

[13] J. Park and S. M.Shotz. Two derivatie-free optimization algorithms for mesh quality improvement. *Procedia Computer Science*, 1:387–396, 2010.

[14] V. N. Parthasarathy and S. Kodiyalam. A constrained optimization approach to finite element mesh smoothing. *Finite Elements in Analysis and Design*, 9:309–320, 1991.

[15] W.H. Press, W.T. Vetterling, S.A. Teukolsky, and B.P.Flannery. *Numerical Recipes in C++*. CUP, Cambridge, second edition, 2002.

[16] J. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Proceedings of the First Workshop on Applied Computational Geometry*, pages 124–133, Philadelphia, Pennsylvania, May 1996. ACM.

[17] P. Zavattieri, E. Dari, and G. Buscaglia. Optimization strategies in unstructured mesh generation. *Int. J. Numer. Methods. Engrg.*, 39:2055–2071, 1996.

DISTRIBUTION:

1	MS 0897	Byron Hanks, 1543
1	MS 0897	Ted Blacker, 1543
1	MS 0380	Joe Jung, 1542
		,
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories