LLNL-TR-795097

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Target Formulation and Construction in Mesh Quality Improvement

P. Knupp

October 22, 2019

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Target Formulation and Construction in Mesh Quality Improvement

Patrick Knupp

Dihedral LLC

LLNL Subcontract B615479

October 10, 2019

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

--

Contents

Abstract

In the Target-matrix Optimization Paradigm (TMOP), it's long been understood that one must create a set of target matrices before the mesh can be optimized. But there is still no general method to create correct, effective targets in response to a specific mesh quality improvement goal. The TMOP literature describes how certain sets of target matrices can be used to control the shape or size of mesh elements, but those examples address only a fraction of the problems that can occur in mesh quality improvement and were not derived from a general framework for target matrix construction. In this work, a general method of target construction is introduced based on an independent set of geometric parameters which are intrinsic to the Jacobian matrices upon which TMOP is based. The parameters enable a systematic approach to target definition and construction. The approach entails two parts. The first part defines correspondences between available raw data (stuff about the mesh and/or the physical solution) and intermediate data (e.g., a field of error estimates). Once the correspondences are established, the raw data is processed into intermediate field data existing on mesh sample points. The second part is to create a model that represents the values of the geometric target parameters as functions of the intermediate data. The model is then tested numerically to establish model constants and effectiveness. This systematic approach to target construction is illustrated in a set of examples to show how it can be applied to common problems in mesh optimization such as equidistribution of geometric properties, preservation of existing good quality, and adaptation of the mesh to the solution. The result is a systematic method of target construction for TMOP which can be applied to a wide variety of planar and volume mesh quality improvement tasks.

Chapter 1

Introduction

Mesh quality may be defined as those features of a mesh which impact simulation robustness, accuracy, and efficiency. Features such as mesh resolution, the spatial distribution of the nodes, mesh geometry, mesh smoothness, mesh topology, the type of elements in the mesh, and order of the element basis functions (if any), can all be considered to be aspects of mesh quality. In any simulation, the practitioner should be concerned with the quality of the meshes they are using. The quality of meshes can sometimes be assessed visually, through the use of mesh diagnostics, and/or values of quality metrics. Sometimes a mesh quality issue is first noticed through the examination of the numerical solution. If a quality issue is identified by any of these means one most likely will want to consider methods for improving the mesh quality. Mesh optimization via node-movement strategies (such as TMOP) is one option.

1.1 Introduction to Mesh Quality Improvement

Mesh generation is a numerical procedure used to create meshes on a given physical domain. Often these meshes lack sufficient 'quality' according to one measure or another. In *mesh quality improvement* (MQI), one presupposes the existence of an initial mesh whose quality is to be improved, through various methods. Two basic categories of mesh quality improvement methods are mesh topology changes and node movement. In general, the term 'mesh optimization' is often used inter-changeably with the term 'mesh quality improvement' and can include both topology changes and node-movement. However, for our purposes, we refer to mesh optimization as a numerical optimization method in which the initial mesh vertex coordinates are changed (thus creating node-movement) in order to minimize an objective function that is a function of the coordinates and which represents a measure of global mesh quality.

A considerable number of mesh optimization methods have been proposed over the years. One of the more famous methods is the variational method leading to the Winslow smoothing procedure [16]. In this procedure, the Euler-Lagrange equations of the variational principle consist of a homogenous, quasi-linear, second-order elliptic set of partial differential equations. The method yields smooth, non-inverted meshes on any given physical domain. The method has been extended to multi-block structured and finite element meshes.

Winslow is an example of an *unweighted* optimization method, i.e., a method which does not employ problem-specific weightings. Unweighted mesh optimization methods essentially provide a 'generic' mesh optimization method in which details of the mesh improvement problem are not taken into account. There is only one solution to the Winslow equations and thus only one optimal mesh exists for a given domain. If the Winslow mesh lacks sufficient quality with respect to a particular application, there is nothing one can do. To address this issue, many mesh optimization methods introduce weighting functions to create *weighted* methods that can potentially control the properties of the optimal mesh. In the case of Winslow, for example, weighting functions P and Q convert the set of elliptic equations to a set of Poisson equations [15]. By selecting appropriate forms and values for the weighting functions one can control additional properties of the mesh in addition to smoothness (e.g., clustering of mesh vertices towards a particular point in the domain).

Ever since the introduction of such weightings, a major issue in mesh optimization has been to find weighting functions (forms) and values which achieve the desired quality improvement. To this point, success has been rather mixed and so efforts continue to strengthen this aspect of mesh optimization. This situation was a major motivation for the creation of the Target-Matrix Optimization Paradigm (TMOP) [11]. In order to address the full range of situations in which the issue of mesh quality arises, TMOP introduces weighting functions in the form of 'target-matrices' which, unlike P and Q, have a well understood geometric meaning. While the P and Q functions appear in the equations as 'source' terms, the target-matrices in

TMOP are more akin to matrix coefficients, as in the Harmonic mapping approach. A major thrust in TMOP is to describe not only the targets but the entire process of target-matrix construction, in which one starts with a specific mesh quality improvement problem and proceeds through various steps until a full set of target-matrices has been determined, prior to numerical optimization. This report summarizes what is currently known about the targets and this procedure.

Mesh quality improvement is desired when one or more features of the existing mesh is judged inadequate. There are a surprising number of different mesh quality improvement goals and contexts in which these goals arise, as can be seen in the following non-exhaustive list. Each of these goals requires an appropriate set of target-matrices, along with proper selection of quality metrics, objective functions, and other tools in mesh optimization.

Common mesh quality improvement goals:

1. Geometric mesh improvement:
   (a) Make the initial mesh smoother,
   (b) Fix locations in the initial mesh having negative volumes or inverted elements,
   (c) Improve excessively small or large angles within the initial mesh,
   (d) Increase the length of 'short' edges in the initial mesh,
   (e) Improve the mesh quality in some transition region (e.g., between regions having different element types),
   (f) Improve inter-element orthogonality and/or volume ratios,
   (g) Improve just the worst-quality elements in the initial mesh,
   (h) Ensure that the improved mesh has quality at least as good as the initial mesh,
   (i) Remove any oscillations or wriggles in the mesh lines of the initial mesh,
   (j) Remove any element 'hour-glassing' present in the initial mesh,
   (k) Improve the mesh quality but don't let the vertices move too far from their initial position,
   (l) Adapt local element areas/volumes of a surface mesh according to surface curvature or other geometric criterion,

(m) Increase the planarity of elements in a surface mesh or the planarity of faces of 3D hexahedral elements,

(n) Maintain initial mesh quality (and features) as the domain on which the mesh resides is deformed,

(o) Improve mesh quality on or near the mesh boundary,

(p) Slide mesh vertices tangentially along an internal interface within the domain.

2. Mesh improvement by adapting the mesh to the physical solution or to discretization error:

   (a) Improve mesh quality in some physical transition region (e.g., between boundary layer and far-field),

   (b) Adapt local element area/volume according to one or more material indicator or volume fraction functions,

   (c) Adapt local element area/volume according to a posteriori error estimates or to the interpolation error,

   (d) Adapt inter-element spacing to match jump in density across material interface,

   (e) Create properly oriented, anisotropic elements according to some feature of the physical solution (e.g. a shock),

   (f) Alignment of mesh elements with a physical vector field (e.g., velocity field, electric field, magnetic field).

Various combinations of the above goals are often needed as well. To fully exploit the Target-Matrix Paradigm one must understand how to construct a set of target-matrices for each of these goals. Some of these goals require not only the right set of target-matrices but also the correct metric type, objective function template, and/or tradeoff coefficients.

1.2 Introduction to the Target-Matrix Paradigm

The basic objects and concepts used in the Target-matrix Optimization Paradigm (TMOP) are reviewed in order to set the stage for this study on target-construction.

1.2.1 Sample Points

TMOP assumes that, for each mesh element, there is a map from a logical element to the physical element. The basis of the map is specified prior to optimization, and can be low-order (i.e., first-order) or high-order (order greater than first). The logical element is determined by the element type (e.g., triangle, quadrilateral, tetrahedron, hexahedron, etc.). So, unless the mesh is a hybrid mesh, there is only one logical element for the entire mesh. In contrast, there are usually many different physical elements in the mesh on the physical domain. For each physical element, one map from the logical element to the physical element is defined. The basis of each element map is usually the same, but the mappings differ because the map is also defined in terms of the coordinates of physical nodes within each element. The nodal coordinates are different on each element. Moreover, the numbering scheme for each element is important. In particular, the correspondence between the first logical and first physical node is important in creating anisotropic elements in the correct direction. A point within a two-dimensional logical element has logical coordinates (ξ, η) while a point within a three-dimensional logical element has logical coordinates (ξ, η, ζ). A *sample point* is a fixed point within the logical element at which we wish to measure local quality. Prior to mesh optimization, a set of sample points within the logical element is selected and remains fixed throughout the optimization procedure. This set of sample points could be, for example, the set of Gaussian integration points within the element. Let the sample points be denoted by $(\xi_{i,j}, \eta_{i,j})$ for 2D elements and $(\xi_{i,j,\ell}, \eta_{i,j,\ell}, \zeta_{i,j,\ell})$ for 3D elements. The sample points can also be assigned a global index k so that (ξ_k, η_k) indicates the k-th sample point within a 2D mesh. Each sample point has a corresponding location (x_k, y_k) or (x_k, y_k, z_k) in physical space. Unlike the logical coordinates, the coordinates of the physical sample points may change during the optimization procedure when a node-movement strategy is applied.

1.2.2 Active-Matrices

Basic facts concerning active-matrices are given here, which apply to any active-matrix no matter the situation or context. At every sample point, TMOP requires both an active and a target matrix. For meshes in $\Re^d$, the active matrix is a $d \times d$ matrix and thus has d^2 elements. [1] The active-

[1] The active-matrix is 3×2 for a mesh on some surface.

matrix is denoted by A and the active-matrix at sample point k (a global index) is A_k. In general, then, the active-matrix can vary from one sample point to the next. The d^2 elements of A are $A_{i,j}$ with $i = 1, \ldots, d$ and $j = 1, \ldots, d$. The active-matrix represents the Jacobian of the map from the logical element to a physical element of the active mesh. The active matrix is a function of the nodal coordinates of the element and thus changes during the mesh optimization procedure. The determinant of A is denoted by α. An important goal of mesh optimization is to ensure that, in the optimal mesh, $\alpha > 0$ at each sample point. Occasionally we make use of the notation $A = [\mathbf{a}_1, \mathbf{a}_2]$ in which the two vectors are the column vectors of $A_{2\times2}$ or $A = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ when $d = 3$. The collection of active-matrices $\{\mathcal{A}\}$ is the set of all active-matrices in the mesh; thus $A_k \in \{\mathcal{A}\}$ for all k. We call $\{\mathcal{A}\}$ the *Active-matrix Set*. In general, A_k changes as the mesh is optimized, which is why the terminology 'active matrix' is used.

1.2.3 Target-Matrices

Basic facts concerning target-matrices are given here, which apply to any target-matrix no matter the situation or context. At every sample point, TMOP requires both an active and a target matrix. For meshes in $\Re^d$, the target matrix is a $d \times d$ matrix and thus has d^2 elements.[2] The target-matrix is denoted by W and the target matrix at sample point k (a global index) is W_k. In general, then, the target-matrix can vary from one sample point to the next. The d^2 elements of W are $W_{i,j}$ with $i = 1, \ldots, d$ and $j = 1, \ldots, d$. The target matrix represents the ideal Jacobian matrix towards which the active matrix will evolve during the mesh optimization procedure. The determinant of W is denoted by ω. A fundamental assumption we make is that, by construction, $\omega > 0$. This ensures that the target-matrix represents a location in the mesh at which the ideal Jacobian determinant is positive and that W^{-1} exists. Occasionally we make use of the notation $W = [\mathbf{w}_1, \mathbf{w}_2]$ in which the two vectors are the column vectors of $W_{2\times2}$ or $W = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3]$ when $d = 3$. There is no essential difference between targets for low-order meshes and targets for high-order meshes. Targets can provide a specific problem-dependent definition of mesh quality.

The collection of target-matrices $\{\mathcal{W}\}$ is the set of all target-matrices in the mesh; thus $W_k \in \{\mathcal{W}\}$ for all k. We call $\{\mathcal{W}\}$ the *Target-matrix Set*. The

[2] The target-matrix is 3×2 for a mesh on some surface.

target-matrix set need not be self-consistent, i.e, the set does not necessarily imply that a corresponding mesh exists. This is not a major difficulty because in mesh optimization the objective function implicitly defines a 'compromise' between inconsistent sets of targets (e.g. as in Least Squares methods). However, the more the targets can be made to correspond to a mesh which exists, the more the optimal mesh will improve quality. This is one reason why the target-construction phase is so important. The target-matrix set is defined prior to the beginning of the mesh optimization procedure. In general, the set will be different for every particular mesh and every mesh quality improvement goal. Target-matrix set 'smoothness' (however that is defined) is necessary if the optimal mesh is to be smooth.

The target-matrix set remains unchanged as mesh nodes move because each target-matrix in the set is associated with a particular sample point in logical space. On the other hand, as the mesh is optimized, the image of each sample point in physical space changes due to node-movement. There are thus two kinds of target-matrices. The first kind is the *logical target-matrix*, whose value remains unchanged as mesh nodes move. Logical targets are useful when one is concerned with mesh quality improvement of a particular element. The second kind of target-matrix is the *physical target-matrix*. In this situation the value of the target-matrix changes as the mesh is moved because its value depends on the physical coordinate of the sample point rather than on its logical coordinate. To evaluate a physical target-matrix one often must interpolate its value from values on the unchanging target-matrix set. Physical targets are useful when one is concerned with mesh quality improvement at particular physical points within the physical domain.

1.2.4 Local Quality Metrics

To measure mesh quality, TMOP uses *local quality metrics*. Local quality metrics are functions from a matrix to a scalar. The value of a local quality metric is measured at sample points of the mesh. Two common forms of local quality metrics are $\mu = \mu(T)$ and $\nu = \nu(A, W)$, with $T = AW^{-1}$ the weighted Jacobian matrix. The values of these metrics are $\mu_k = \mu(T_k)$ and $\nu_k = \nu(A_k, W_k)$, where k is the global sample point index. Local quality metrics are *well-posed* if they satisfy a certain set of requirements such as continuity, differentiability, and more. Important *categories* of local quality metrics are barrier, non-barrier, pseudo-barrier, and shifted-barrier. Barrier metrics are used to ensure that the element mappings remain non-singular when the initial mappings are non-singular. Pseudo-barrier and shifted-barrier metrics are used when the initial mappings contain singular points; their purpose is to simultaneously untangle and optimize the mesh. Sets of local quality metrics can be combined (via a norm or average) to create element quality metrics, local patch objective functions, and global objective functions. Local quality metrics also come in a variety of *metric types*; metric types in TMOP will the subject of a follow-on document.

1.2.5 Sets of Matrices

Certain sets of matrices in connection with TMOP have been defined previously. The main matrix set with which we are concerned is

- $\mathcal{M}_d$, the set of $d \times d$ matrices with real elements.

The cases $d = 2$ and $d = 3$ are important in planar and volume meshing, respectively. The following subsets of $\mathcal{M}_d$ also play an important role:

- $\mathcal{M}_d^s$, the set of singular matrices,
- $\mathcal{M}_d^{\sim s} = \mathcal{M}_d \setminus \mathcal{M}_d^s$, the set of non-singular matrices,
- $\mathcal{M}_d^p$, the set of matrices whose determinant is positive,
- $\mathcal{M}_d^d$, the set of degenerate matrices.

Definition.
A matrix $X \in M_d$ is *degenerate* if one or more of its column vectors is zero. Define $\mathcal{M}_d^{\sim d}$ to be the set of non-degenerate matrices.

Clearly, $\mathcal{M}_d^d \subset \mathcal{M}_d^s$.

1.3 Development of the Target-Matrix Paradigm

Methods for mesh quality improvement via node movement must, in some fashion, define relations between a definition of mesh quality and the mesh geometry. For example, if mesh quality is defined to be a solution-adapted mesh, then various solution-based quantities such as gradient, Hessian, flow-vectors, error estimates, etc. must be connected to the mesh geometry (i.e., to volume, shape, and orientation). The exact manner in which these relations are made varies from one MQI method to the next.

While the Target-matrix Paradigm has been studied for some time, the theory outlined in Section 1.2 is incomplete, particularly with regard to a general description of how the connection between mesh quality and mesh geometry is made for each of the mesh quality improvement goals listed in Section 1.1. This research monograph is a major step in filling this gap.

In Chapter 2, various analytic relationships between mesh geometry and the Jacobian matrix are explored in order to understand how they can be exploited in target construction. Section 2.1 deals with the case of planar meshes whose vertices lie in $\Re^2$ and $A \in \mathcal{M}_2$. Section 2.2 deals with the case of volume meshes whose vertices lie in $\Re^3$ and $A \in \mathcal{M}_3$. Sections 2.1.1 and 2.2.1 define a standard set of scalar geometric parameters that can be derived from the Jacobian matrix. Sections 2.1.2 and 2.2.2 deal with the problem of extracting these parameters from a given matrix so that local quality metrics may be evaluated. Various geometric extraction vectors and matrices in Sections 2.1.3 and 2.2.3 are defined; these are also used in the assessment or evaluation of mesh quality. Taking the opposite point of view, Sections 2.1.4 and 2.2.4 show how one defines the target matrix given the values of these geometric parameters. Next, Sections 2.1.5 and 2.2.5 give a matrix decomposition of the Jacobian matrix in terms of 'size', 'shape', and 'orientation' factors. The matrix factors motivate a definition (in Sections 2.1.6 and 2.2.6) of certain relevant matrix sets in addition to those of Section 1.2.5. In Sections 2.1.7 and 2.2.7 it is shown that if an extracted matrix belonging to one of these sets is equal to the corresponding target matrix factor, then the scalar geometric parameters in the extracted matrix are equal to the scalar geometric target parameters. This type of result is needed in the definition of quality metrics of a given type. Sections 2.1.8 and 2.2.8 provide a summary of the application of these ideas to the active and target matrices.

Prior to numerical optimization of the active mesh, one must endow the targets at every sample point with numerical values by some algorithm. In the case that no algorithm is available, one must be devised. The process by which this is done, described in Chapter 3, is called *Target Construction*. There are three phases in Target Construction, beginning with formulating a strategy which takes into account the mesh simulation and context (section 3.1.1), continuing with a determination on which geometric parameters are the most important to control (section 3.1.2), and finally, establishing correspondences between raw data and particular geometric parameters (section 3.1.3). In the second phase of Target Construction one identifies sources of raw data and devises algorithms for converting the raw data to intermediate data. Intermediate data consists of either mesh functionals (section 3.2.1) or simulation functionals (section 3.2.2). In the final stage of Target Construction one develops various models relating the functionals to values of the target geometric parameters (sections 3.3.1 and 3.3.2). In sections 3.5 and 3.6 examples of Target Construction are provided which illustrate how the construction of targets depends on the mesh quality improvement goal.

Chapter 2

Relations Between Matrices & Geometric Parameter Sets

In this chapter a standard set of *geometric parameters* is defined. The parameters are related to the elements of any 2×2 or 3×3 matrix, whether or not the matrix represents the Jacobian of some map. When applied to A, the parameters represent the local geometry of the active mesh at the sample point. When applied to W, the parameters represent the local geometry of the target mesh at the sample point. Various relationships between the TMOP matrices and their geometric parameters are explored.

From an abstract point of view, the geometric target parameters can be regarded as a vector $\mathbf{p}$ belonging to a subset $\mathcal{P}_W$ of $\Re^{d^2}$, with $d = 2$ or 3. The matrix W belongs to a subset $\mathcal{S}_W$ of $\mathcal{M}_d$. It is shown that there exists a mapping $\mathbf{F}$ from the set $\mathcal{P}_W$ to the matrix set $\mathcal{S}_W$. The mapping has an inverse and the mapping is one-to-one and onto when the sets $\mathcal{S}_W$ and $\mathcal{P}_W$ are properly defined. In addition, the active matrix A belongs to a subset $\mathcal{S}_A$ of $\mathcal{M}_d$; there exists a one-to-one and onto mapping $\tilde{\mathbf{F}}$ from $\mathcal{S}_A$ to a subset $\mathcal{P}_A$ of $\Re^{d^2}$, again provided the two sets are properly defined. In general, $\mathcal{S}_W \subset \mathcal{S}_A$. We show that, on the set $\mathcal{S}_W$, $\tilde{\mathbf{F}} = \mathbf{F}^{-1}$, and likewise for the set $\mathcal{S}_A$.

When applied to the active matrix, the mapping $\tilde{\mathbf{F}}$ is used to find the local quality of the active mesh in terms of the geometric parameters. That is, the *active geometric parameters* $\tilde{\mathbf{p}}$ are regarded as functions of the elements of the matrix A. In this case, we write $\tilde{\mathbf{p}} = \tilde{\mathbf{F}}(A)$, and say that the active geometric parameters have been *extracted* from the active Jacobian matrix. This extraction process can be applied to (1) mesh quality assessment, (2) target construction, and (3) to the evaluation of local quality metrics during the mesh optimization process.

The mapping $\mathbf{F}$ is used to analytically define the target-matrix in terms of the *target geometric parameters*, $\mathbf{p}$. That is, given $\mathbf{p}$, the elements of W are given as functions of the geometric parameters. We can write $W_{ij} = F_{ij}(\mathbf{p})$ or $W = F(\mathbf{p})$.

2.1 Planar Meshes in $\Re^2$

Planar meshes contain two-dimensional mesh elements such as triangles, quadrilaterals, or polygons. While planar meshes can exist in both $\Re^2$ and $\Re^3$, it shall be assumed in this report that a planar mesh belongs to $\Re^2$. Any planar mesh in $\Re^3$ can be transformed into a planar mesh in $\Re^2$. Each triangular or quadrilateral element in $\Re^2$ has a mapping from a two-dimensional logical space to points within the element. The boundaries of the physical elements can be curved in the high-order element case. Rather than try to characterize the quality of a planar element directly, TMOP first measures local quality at sample points within the element. The set of sample points within an element can then be combined via some averaging technique to create an element quality metric. The present section, however, is concerned with the definition and construction of targets at a given sample point of a planar mesh element.

Before elaborating on the active and target-matrices, it is useful to gather together some important geometry-related facts about square matrices with real elements. Geometry is emphasized because mesh quality improvement generally involves controlling geometric properties of a given mesh. Let $X \in \mathcal{M}_d$. Section 2.1 deals with the case $d = 2$. In terms of application to mesh optimization, it may be helpful to think of X as the active matrix A in the previous chapter.

2.1.1 Geometric Parameters for Meshes in $\Re^2$

At a given sample point in a 2D mesh element, there exists the two tangent vectors to the element mapping. From the two tangent vectors, various geometric quantities can be defined: the lengths of the two tangents, the angle between them, the two angles between the tangents and the x-axis, the aspect ratio, and the area of the parallelogram between the two tangents. Let us use the following symbols to denote these quantities: ℓ_1, ℓ_2, ϕ, θ, θ', ρ, and υ. Various relations between these quantities exist,

$$\begin{aligned} \phi &= \theta' - \theta \\ \rho &= \frac{\ell_2}{\ell_1} \\ \upsilon &= \ell_1 \ell_2 \sin\phi \end{aligned}$$

It is also useful to define a 'size' quantity ζ by

$$\zeta = \ell_1 \ell_2$$

We will call these quantities the *geometric parameters* of planar mesh quality. In general, at every sample point, the value of each parameter can be different from its value at another sample point (e.g., area will differ from one sample point to the next).

Due to the relations above, only four of the parameters are needed to create a parameter set $\mathbf{p}_2$ in which each of the parameters in $\mathbf{p}_2$ is fully independent of the others in the set and from which a Jacobian matrix can be fully determined. We elect to define $\mathbf{p}_2 \in \Re^4$ as follows

$$\mathbf{p}_2 = (\upsilon, \theta, \phi, \rho)$$

We call $\mathbf{p}_2$ the *standard* independent geometric parameter set for planar mesh quality. The non-standard parameters may be calculated from the parameters in $\mathbf{p}_2$,

The two tangent vectors at a sample point can thus be used to define the standard geometric parameter set at the sample point. We use the notation

$$\mathbf{p}_2(k) = (\upsilon_k, \theta_k, \phi_k, \rho_k)$$

to indicate the values of the standard parameters at sample point k.

2.1.2 Geometric Extraction Functions for Meshes in $\Re^2$

We have already seen in TMOP many examples of scalar functions whose argument is a matrix. For example, there are the functions $|X|$, $tr(X)$, and $det(X)$. Other important functions of matrices are themselves matrices. For example, the function $F(X) = X^{-1}$. For convenience, let X be alternatively represented not in terms of its elements X_{ij}, but in terms of column vectors, as in $X = [\mathbf{x}_1, \mathbf{x}_2]$ when $d = 2$ and $X = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ when $d = 3$. Representation of X in terms of its column vectors (instead of its elements) allows one to consider the 'geometry' of X in terms of the geometry of its column vectors. This is important because the first-order aspects of local mesh quality can be measured in terms of the geometry of the column vectors of the Jacobian matrix.

In this section scalar functions of 2×2 matrices which are related to the *geometry* of the column vectors are considered. These functions are called *Geometric Extraction Functions.* Given $X_{2\times 2}$, the functions extract information from X concerning the geometry of its column vectors. Extraction functions for length, size, area, orientation, skew, and aspect ratio are defined. In TMOP, exaction functions are used to (1) evaluate, using the Jacobian matrix, the local geometric quality of a mesh at a given sample point and (2) assist in the construction of the target matrix. In both of these uses, the matrix of interest is $X = A$, the active matrix. A *tilde* over a given geometric parameter signifies an extraction function.

The geometric properties of the column vectors are expressed as functions of the matrix X. Assume that X is non-degenerate. Two functions to extract the length of the column vectors are given by

$$\begin{aligned} \tilde{\ell}_1(X) &\equiv |\mathbf{x}_1| \\ \tilde{\ell}_2(X) &\equiv |\mathbf{x}_2| \end{aligned}$$

These two functions are defined for $X \in \mathcal{M}_2$. For $i = 1, 2$, we have $\tilde{\ell}_i(X) \geq 0$ for any $X \in \mathcal{M}_2$. If $X \in \mathcal{M}_2^{\sim d}$, then $\tilde{\ell}_i(X) > 0$.

Next, define the 'size' function by

$$\tilde{\zeta}(X) \equiv \tilde{\ell}_1(X)\, \tilde{\ell}_2(X)$$

for $X \in \mathcal{M}_2$. Clearly, $\tilde{\zeta}(X) > 0$ for all $X \in \mathcal{M}_2^{\sim d}$.

The 'volume' (or area) function is

$$\tilde{v}(X) \equiv det(X)$$

For $X \in \mathcal{M}_2$, $-\infty < \tilde{v}(X) < \infty$, but for $X \in \mathcal{M}_2^p$, $\tilde{v}(X) > 0$.

For X non-degenerate define the 'orientation' functions $\tilde{\theta}(X)$, $\tilde{\theta}'(X)$ by

$$\cos\tilde{\theta} \equiv \frac{X_{11}}{\tilde{\ell}_1(X)}$$
$$\sin\tilde{\theta} \equiv \frac{X_{21}}{\tilde{\ell}_1(X)}$$

and

$$\cos\tilde{\theta}' \equiv \frac{X_{12}}{\tilde{\ell}_2(X)}$$
$$\sin\tilde{\theta}' \equiv \frac{X_{22}}{\tilde{\ell}_2(X)}$$

The range of the *orientation function* is taken to be $-\pi < \tilde{\theta} \leq \pi$.

Define the 'skew' function $\tilde{\phi}(X)$ for X non-degenerate by

$$\cos\tilde{\phi} \equiv \frac{X_{11}X_{12} + X_{21}X_{22}}{\tilde{\zeta}(X)}$$
$$\sin\tilde{\phi} \equiv \frac{det(X)}{\tilde{\zeta}(X)}$$

One can show that $\tilde{\phi}(X) = \tilde{\theta}'(X) - \tilde{\theta}(X)$. The range of the skew function is taken to be $-\pi < \tilde{\phi}(X) \leq \pi$.

Define the 'aspect ratio' function $\tilde{\rho}(X)$ for X non-degenerate by

$$\tilde{\rho}(X) \equiv \frac{\tilde{\ell}_2(X)}{\tilde{\ell}_1(X)}$$

Because $X \in \mathcal{M}_2^{\sim d}$, $\tilde{\rho}(X) > 0$.

Scalar extraction functions $\tilde{\ell}_1(X)$, $\tilde{\ell}_2(X)$, $\tilde{\zeta}(X)$, $\tilde{v}(X)$, $\tilde{\theta}(X)$, $\tilde{\theta}'(X)$, $\tilde{\phi}(X)$, and $\tilde{\rho}(X)$ have been defined. Functions $\tilde{\ell}_i(X)$, $\tilde{\zeta}(X)$, and $\tilde{v}(X)$ exist for every $X \in \mathcal{M}_2$. On the other hand, functions $\tilde{\theta}(X)$, $\tilde{\theta}'(X)$, $\tilde{\phi}(X)$, and $\tilde{\rho}(X)$ require $X \in \mathcal{M}_2^{\sim d}$. Thus, all of the extraction functions exist on the set of non-degenerate 2×2 matrices. These functions correspond to geometric properties of the matrix column vectors: length, size, volume, orientation, skew, and aspect ratio. Because of the relationship between the orientation and skew extraction functions, the function $\tilde{\theta}'(X)$ will not be used very often.

Relating back to the mapping $\tilde{\mathbf{F}}$ mentioned at the start of this Chapter, we see that

$$\tilde{\mathbf{p}}_2(X) = \Big(\tilde{\upsilon}(X), \tilde{\theta}(X), \tilde{\phi}(X), \tilde{\rho}(X)\Big)$$

is the standard set of geometric parameters extracted from X.

Table 2.1 gives the image of three commonly used matrix sets under the various scalar extraction functions. The images are intervals $[I_k]$, $k = 1, 2, 3, 4$ on the real line. Each interval gives the range of one of the extraction functions as X varies over the indicated matrix set. In the table, U means that there exist elements of the matrix set for which the selected extraction function is undefined; in that case, the image is also undefined. Because $\cos\theta$ and $\sin\theta$ are periodic functions of θ, with period 2π, we can restrict the range of the orientation function to be any interval of length 2π. This range is given by the interval $[I_3](a)$, where the real number 'a' is to be determined later. The same situation holds for the skew parameter, but only when $X \in \mathcal{M}_2^{\sim d}$. The image of each matrix set under the extraction vector $\tilde{\mathbf{p}}_2$ is the Cartesian Product of the intervals in the appropriate column of the table.

Function	$X \in \mathcal{M}_2^p$	$X \in \mathcal{M}_2^{\sim d}$	$X \in \mathcal{M}_2$
$\tilde{\upsilon}(X)$	$[I_2]$	$[I_1]$	$[I_1]$
$\tilde{\theta}(X)$	$[I_3](a)$	$[I_3](a)$	U
$\tilde{\phi}(X)$	$[I_4]$	$[I_3](b)$	U
$\tilde{\rho}(X)$	$[I_2]$	$[I_2]$	U

Table 2.1: Maximum Range of Each Extraction Function Given $X_{2\times 2}$

Intervals: $[I_1] = (-\infty, \infty)$; $[I_2] = (0, \infty)$; $[I_3](a) = (a - \pi, a + \pi]$; $[I_4] = (0, \pi)$.

2.1.3 Geometric Extraction Matrices for Meshes in $\Re^2$

One can factor $X \in \mathcal{M}_2^{\sim d}$ in terms of *extraction matrices* as follows

$$\begin{aligned} X &= \sqrt{\tilde{\zeta}(X)}\,\tilde{R}(X)\,\tilde{Q}(X)\,\tilde{D}(X) \\ &= \sqrt{\tilde{\zeta}(X)}\,\tilde{R}(X)\,\tilde{S}(X) \\ &= \tilde{R}(X)\,\tilde{U}(X) \end{aligned}$$

with extraction matrices $\tilde{R}(X)$ (a *rotation* matrix), given by

$$\tilde{R}(X) \equiv \begin{pmatrix} \cos[\tilde{\theta}(X)] & -\sin[\tilde{\theta}(X)] \\ \sin[\tilde{\theta}(X)] & \cos[\tilde{\theta}(X)] \end{pmatrix}$$

and $\tilde{Q}(X)$ (a *skew* matrix), given by

$$\tilde{Q}(X) \equiv \begin{pmatrix} 1 & \cos[\tilde{\phi}(X)] \\ 0 & \sin[\tilde{\phi}(X)] \end{pmatrix}$$

and $\tilde{D}(X)$ (an *aspect ratio* matrix), given by

$$\tilde{D}(X) \equiv \begin{pmatrix} \frac{1}{\sqrt{\tilde{\rho}(X)}} & 0 \\ 0 & \sqrt{\tilde{\rho}(X)} \end{pmatrix}$$

Further, $\tilde{S}(X) \equiv \tilde{Q}(X)\tilde{D}(X)$, and $\tilde{U}(X) = \sqrt{\tilde{\zeta}(X)}\tilde{S}(X)$.

Extraction matrices are useful when defining various metric types.

The matrix X can be expressed in terms of its own geometric extraction functions, as seen the the proposition below. The identity is important mainly in motivating the construction of the target matrix in the next subsection.

Proposition.
Given matrix $X \in \mathcal{M}_2^{\sim d}$, the following 2D *geometric extraction identity* identity holds

$$X \equiv \sqrt{\tilde{\zeta}(X)} \begin{pmatrix} \frac{1}{\sqrt{\tilde{\rho}(X)}}\cos[\tilde{\theta}(X)] & \sqrt{\tilde{\rho}(X)}\cos[\tilde{\theta}(X)+\tilde{\phi}(X)] \\ \frac{1}{\sqrt{\tilde{\rho}(X)}}\sin[\tilde{\theta}(X)] & \sqrt{\tilde{\rho}(X)}\sin[\tilde{\theta}(X)+\tilde{\phi}(X)] \end{pmatrix} \quad (2.1)$$

This can be verified by directly substituting the function definitions into the right-hand-side.

2.1.4 A Parametric Definition of 2×2 Matrices

In section 2.1.2 the problem of extracting geometric information from the column vectors of a given 2×2 matrix was considered. In the present section, the inverse problem is considered, namely, given values of the geometric parameters in $\mathbf{p}_2$, define a matrix $Y_{2\times 2}$ such that the parameter values can be recovered using the previously defined extraction functions applied to Y. In terms of mesh optimization, one can think of Y as the Target-Matrix W.

Using the geometric extraction identity from the last section as motivation, and given size, orientation, skew, and aspect ratio parameters $\zeta > 0$, θ, ϕ, and $\rho > 0$, *define* a 2×2 matrix Y as follows

$$Y(\zeta,\theta,\phi,\rho) = \sqrt{\zeta}\begin{pmatrix} \frac{1}{\sqrt{\rho}}\cos\theta & \sqrt{\rho}\cos(\theta+\phi) \\ \frac{1}{\sqrt{\rho}}\sin\theta & \sqrt{\rho}\sin(\theta+\phi) \end{pmatrix} \tag{2.2}$$

Equation 2.2 is called the standard *parametric definition* of Y.[1] In terms of the mapping $\mathbf{F}$, $Y = \mathbf{F}(\mathbf{p}_2)$

Proposition.
Given Y constructed as above, Y is non-degenerate. If, in addition, $\sin\phi \neq 0$, then Y is non-singular.
Proof.
The length of the first column vector of Y is $\sqrt{\frac{\zeta}{\rho}}$ and the length of the second column vector of Y is $\sqrt{\rho\,\zeta}$. By assumption, $\zeta > 0$ and $\rho > 0$, thus Y is non-degenerate. Moreover, the determinant of Y is $\zeta\sin\phi$, so Y is non-singular provided $\sin\phi \neq 0$ §

Given Y defined as above, the extraction functions $\tilde{\zeta}$, $\tilde{\theta}$, $\tilde{\phi}$, and $\tilde{\rho}$ can be applied to Y because Y is non-degenerate. The following Proposition is easily verified.

Proposition.
Evaluating the four extraction functions on the Y constructed as above, we find $\tilde{\zeta}(Y) = \zeta$, $\tilde{\theta}(Y) = \theta$, $\tilde{\phi}(Y) = \phi$, and $\tilde{\rho}(Y) = \rho$.

The parametric definition is thus consistent with the results of applying the scalar extraction functions to the target matrix. In terms of the mapping,

[1] The matrix Y can be parameterized in other ways, but this particular parameterization is good for our purpose.

$\tilde{\mathbf{F}}(Y) = \mathbf{p}_2$.

2.1.5 The Parametric Matrix Factors of $Y_{2\times 2}$

The parametric representation of Y in Equation (2) has the following matrix factorization (or decomposition):

$$Y(\zeta, \theta, \phi, \rho) = \sqrt{\zeta}\, R(\theta)\, Q(\phi)\, D(\rho) \tag{2.3}$$

where

$$R(\theta) \equiv \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

$$Q(\phi) = \begin{pmatrix} 1 & \cos\phi \\ 0 & \sin\phi \end{pmatrix}$$

$$D(\rho) = \begin{pmatrix} \frac{1}{\sqrt{\rho}} & 0 \\ 0 & \sqrt{\rho} \end{pmatrix}$$

This is called the parametric factorization of $Y_{2\times 2}$. The matrix factors exist because $\rho > 0$. Notice that each factor is a function of a single geometric parameter. The matrix $R(\theta)$ is called the *orientation* matrix; R is a rotation matrix (i.e., $det(R) = 1$, $R^t R = I$). The matrix $Q(\phi)$ is called the *skew* matrix; Q is upper triangular with unit column vectors. The matrix $D(\rho)$ is called the *aspect ratio* matrix; D is diagonal, with unit determinant. Taking the determinant of both sides of the factorization relation, one finds $\upsilon = \zeta \sin\phi$, as expected.

We also define the *shape matrix* S by $S \equiv QD$ and the *shape+size* matrix U by $U = \sqrt{\zeta}\, S$. Thus

$$\begin{aligned} Y &= \sqrt{\zeta}\, R(\theta)\, Q(\phi)\, D(\rho) \\ &= \sqrt{\zeta}\, R(\theta)\, S(\phi, \rho) \\ &= R(\theta)\, U(\zeta, \phi, \rho) \end{aligned}$$

The matrix S is upper triangular with determinant equal to $\sin\phi$. The matrix U is upper triangular with determinant υ.

Proposition.
The previously defined extraction matrices applied to Y yield $\tilde{R}(Y) = R(\theta)$, $\tilde{Q}(Y) = Q(\phi)$, $\tilde{D}(Y) = D(\rho)$, $\tilde{S}(Y) = S(\phi, \rho)$ and $\tilde{U}(Y) = U(\zeta, \phi, \rho)$. The proof is straight-forward.

The parametric matrix factors defined in this section are used later in this series of reports to define local quality metrics of various types.

In terms of well-known matrix decompositions, $Y = RU$ is closely related to the well-known 'QR' factorization from linear algebra. In general, the QR-decomposition of a real matrix Y is the product of a real orthogonal matrix R and a real upper triangular matrix U. In QR, one commonly chooses the diagonal entries of U to be positive. In that case the determinant of Y depends on the determinant of R (either 1 or -1) and the determinant of U. With that convention, the factorization is unique [8]. In this report we've used an alternative convention in which the determinant of R is always 1 (i.e., the matrix is always a rotation) and the determinant of U is not necessarily positive. In the latter convention the determinant of Y depends only on the determinant of U. This convention is used because in mesh optimization it is usual to consider the local volume to be a signed quantity and, since $det(U) = v$, it is U that needs a signed determinant, not R.

It is not suggested that one use the numerical QR algorithm in TMOP to find the matrices R and U. There is no need for that because one can express both matrices (and the other matrices mentioned in this report) in analytic form. The analytic forms are used primarily to develop the relations between the geometric parameters and the Jacobian matrices, along with the extraction functions and targets. This motivates another reason we've used the alternative convention for QR in this report. The alternative is used because it simplifies the analytic representation of the matrices R and U. For example, the analytic matrix $R(\theta)$ given in the previous section can always be used in the alternative convention, whereas in the standard QR convention this analytic formula could not be used if the determinant of Y were negative. This need for switching analytic formulas depending on the sign of the determinant of Y arises in the standard QR convention, but not in the alternative. With the alternative, the switching is more easily accomplished analytically via $\sin\phi$, which appears in U and can be either positive or negative.

2.1.6 Matrix Types and Sets of 2×2 Matrices

The matrices $R(\theta)$, $Q(\phi)$, and $D(\rho)$ appearing in the geometric factorization of $Y_{2\times 2} \in \mathcal{M}_2^{\sim d}$ have certain properties of note.

Definition.
Define $\mathcal{M}_2^{rot}$ to be the set of all 2×2 rotation matrices. For example, $R(\theta) \in \mathcal{M}_2^{rot}$.

Definition.
If $Q_{2\times 2}$ is upper triangular and has column vectors all with unit length, then Q is a *skew* matrix. Define $\mathcal{M}_2^{skw}$ to be the set of all 2×2 skew matrices. For example, $Q(\phi) \in \mathcal{M}_2^{skw}$.

Definition.
If $D_{2\times 2}$ is a positive definite diagonal matrix whose determinant is 1, then D is an *aspect ratio* matrix. Define $\mathcal{M}_2^{asp}$ to be the set of all 2×2 aspect ratio matrices. For example, $D(\rho) \in \mathcal{M}_2^{asp}$.

Definition.
If $S_{2\times 2}$ is upper triangular and the product of the lengths of its column vectors is 1, then S is a *shape* matrix. Define $\mathcal{M}_2^{shp}$ to be the set of all 2×2 shape matrices. For example, $S(\phi, \rho) \in \mathcal{M}_2^{shp}$.

Definition.
If $U_{2\times 2}$ is upper triangular, then U is a *shape+size* matrix. Define $\mathcal{M}_2^{shs}$ to be the set of all 2×2 shape+size matrices. For example, $U(\zeta, \phi, \rho) \in \mathcal{M}_2^{shs}$.

The identity matrix $I_{2\times 2}$ belongs to all of the sets defined above. All pairwise intersections of the three sets $\mathcal{M}_2^{rot}$, $\mathcal{M}_2^{skw}$, and $\mathcal{M}_2^{asp}$ yields the identity matrix. Further, $\mathcal{M}_2^{skw} \subset \mathcal{M}_2^{shp} \subset \mathcal{M}_2^{shs}$.

2.1.7 Equality of Geometric Parameters for Meshes in $\Re^2$

The following Propositions justify many of the local quality metrics used in TMOP. Each metric depends on the active matrix and a target. The matrices X and Y below are placeholders for these two matrices, respectively.

Proposition.

Let $X, Y \in \mathcal{M}_2^{\sim d}$ and let Y have the factorization (2.3). If $\tilde{Q}(X) = Q(\phi)$, then $\tilde{\phi}(X) = \phi$.
Proof.
Equality of the two matrices requires

$$\begin{aligned} \cos\left[\tilde{\phi}(X)\right] &= \cos\phi \\ \sin\left[\tilde{\phi}(X)\right] &= \sin\phi \end{aligned}$$

The Proposition immediately follows. §

Similarly, if $\tilde{R}(X) = R(\theta)$, then $\tilde{\theta}(X) = \theta$. If $\tilde{D}(X) = D(\rho)$, then $\tilde{\rho}(X) = \rho$.

Proposition.
Let $X, Y \in \mathcal{M}_2^{\sim d}$ and let Y have the factorization (2.3). If $\tilde{S}(X) = S(\phi, \rho)$, then $\tilde{\phi}(X) = \phi$ and $\tilde{\rho}(X) = \rho$.
Proof.
Equality of the two matrices requires equality of their elements. This yields

$$\sqrt{\tilde{\rho}(X)} = \sqrt{\rho}$$

i.e., $\tilde{\rho}(X) = \rho$. Then we must also have $\tilde{\phi}(X) = \phi$, proving the result. §

Corollary.
If $\tilde{S}(X) = S(\phi, \rho)$, then $\tilde{Q}(X) = Q(\phi)$ and $\tilde{D}(X) = D(\rho)$.

Similar propositions can be proved for the shape+size and other matrices like the product of rotation and skew matrices.

2.1.8 Application to the 2×2 Active and Target-Matrices

The previous developments are applied to the 2×2 active and target-matrices in TMOP.

The target-matrix W can be usefully represented parametrically, as in equation (2.2), given parameters υ, θ, ϕ, and ρ. Values of these parameters must be supplied at every sample point in order to construct the set $\{\mathcal{W}\}$ prior to mesh quality assessment and improvement. The process of determining these values is called *target-parameter construction*; this process is discussed in the next Chapter. Each target-matrix is required to be in $\mathcal{M}_2^p$; therefore, we require $\upsilon > 0$. By construction, $\mathbf{p}_2 \in W_a^4 \subset \Re^4$, where W_a^4 is the Cartesian Product of the intervals in the first column of Table 2.1, i.e.,

$$\begin{aligned} W_a^4 &\equiv [I_2] \times [I_3](a) \times [I_4] \times [I_2] \\ &= (0, \infty) \times (a - \pi, a + \pi] \times (0, \pi) \times (0, \infty) \end{aligned}$$

In terms of the discussion at the beginning of this chapter, we have $\mathcal{S}_W = \mathcal{M}_2^p$ and $\mathcal{P}_W = W_a^4$, which results in the one-to-one, onto mapping $\mathbf{F}$ from $\mathcal{P}_W$ to $\mathcal{S}_W$.

Unlike the target-matrix, the active matrix is not constructed; rather, it is given. The only thing we can be sure of is that $A \in \mathcal{M}_2$. But, if A is degenerate, parameters $\tilde{\theta}(A)$ and $\tilde{\phi}(A)$ are undefined. Although the occurrence of degenerate A may be rare, robust practical algorithms must account for this possibility in some way. Thus, we restrict $\mathcal{S}_A$ to $\mathcal{M}_2^{\sim d}$. The mapping $\tilde{\mathbf{F}}$ is then well-defined and

$$\mathcal{P}_A \equiv [I_1] \times [I_3](a) \times [I_3](b) \times [I_2]$$

with a and b some selected values.

2.2 Volume Meshes

Volume meshes contain three-dimensional mesh elements such as tetrahedrons, hexahedrons, pyramids, prisms, or polyhedra. Volume meshes belong to $\Re^3$. Each 3D element in $\Re^3$ has a mapping from a three-dimensional logical space to points within the physical element. The boundary faces of the elements can be curved in the high-order element case and even in the low order case for some element types. Rather than try to characterize the quality of a volume element directly, TMOP first measures local quality at sample points within the element. Section 2.2 is concerned with target definition and construction for volume meshes.

The analysis of Section 2.1 is repeated, where needed, for the case $X \in \mathcal{M}_3$.

2.2.1 Geometric Parameters for Volume Meshes

At a given sample point in a 3D mesh element, there exists three tangent vectors to the element mapping. From the three vectors various geometric quantities can be defined: the lengths ℓ_1, ℓ_2, ℓ_3 of the vectors; the three included angles ϕ_{12}, ϕ_{13}, ϕ_{23} between pairs of the vectors; and the volume υ of the parallel-o-piped enclosed by the vectors. In addition, there are three dihedral angles χ_1, χ_2, and χ_3, giving the angles between the three planes that the pairwise combinations of the vectors make. There are also three aspect ratios ρ_1, ρ_2, ρ_3 and three pairs of spherical coordinates (θ_1, ψ_1), (θ_2, ψ_2), and (θ_3, ψ_3) defining the directions of the vectors.

Quite a few relations exist between these 19 quantities, i.e., they are not completely independent of one another. Due to these relations, only nine parameters are needed to create a parameter set $\mathbf{p}_3$ in which each of the parameters in $\mathbf{p}_3$ is fully independent of the others in the set. We elect to define $\mathbf{p}_3$ as follows

$$\mathbf{p}_3 = (\upsilon, \theta, \psi, \beta, \phi_{12}, \phi_{13}, \chi, \rho_1, \rho_2)$$

This is called the standard independent geometric parameter set for volume mesh quality. The other geometric parameters may be calculated from the parameters in $\mathbf{p}_3$. Here, $\chi = \chi_1$ (the dihedral angle between the normals to the 1,2-plane and the 1,3-plane).

The set $\mathbf{p}_3^{ori} = (\theta, \psi, \beta)$ is called the *orientation* parameter set, with θ, ψ and β to be defined later. The set $\mathbf{p}_3^{skw} = (\phi_{12}, \phi_{13}, \chi)$ is called the *skew* parameter set. The set $\mathbf{p}_3^{asp} = (\rho_1, \rho_2)$ is called the *aspect ratio* parameter set. The *shape* parameter set is $\mathbf{p}_3^{shp} = \mathbf{p}_3^{skw} \cup \mathbf{p}_3^{asp}$. The *shape+size* parameter set if $\mathbf{p}_3^{shs} = (\zeta) \cup \mathbf{p}_3^{shp}$. Define $\mathbf{p}_3^{sso} = \mathbf{p}_3^{shs} \cup \mathbf{p}_3^{ori}$ to be the *shape+size+orientation* parameter set. Finally, $\mathbf{p}_3 = [\mathbf{p}_3^{sso} \setminus (\zeta)] \cup (\upsilon)$.

The values of the nine geometric parameters at a sample point k are denoted by

$$\mathbf{p}_3(k) = (\upsilon_k, \theta_k, \psi_k, \beta_k, \phi_{12}(k), \phi_{13}(k), \chi_k, \rho_1(k), \rho_2(k))$$

The three tangent vectors form the columns of the 3×3 Jacobian matrix. Consequently, given a non-singular Jacobian matrix at a sample point, one can extract values of the parameters in $\mathbf{p}_3$. The volume extraction functions are given in Sections 2.2.2 and 2.2.3. In Section 2.2.4 a 3×3 'target' matrix is defined in terms of the parameters in $\mathbf{p}_3$. Section 2.2.5 discusses an analytic matrix decomposition representing the Jacobian matrix in terms of rotation, skew, and aspect ratio matrices.

Section 2.2.6 defines certain matrix types and sets thereof. These are used in the theory of metric types presented in this series of reports. Section 2.2.7 contains several Propositions on the consequences of certain equalities relating active parameters or matrices to target parameters or matrices. The Propositions help us later in stating the global minimums of local quality metrics. Finally, Section 2.2.8 concerns the application of the ideas in Section 2.2 to the active and target matrices or parameters.

2.2.2 Geometric Extraction Functions for Volume Meshes

This section gives geometric extraction functions for the case $X_{3\times 3} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ related to the geometry of the column vectors.

There are three length functions on $\mathcal{M}_3$, defined by

$$\tilde{\ell}_i(X) \equiv |\mathbf{x}_i|$$

for $i = 1, 2, 3$. For X non-degenerate, $\tilde{\ell}_i(X) > 0$.

The 'size' function, defined on $\mathcal{M}_3$, is

$$\tilde{\zeta}(X) \equiv \tilde{\ell}_1(X)\,\tilde{\ell}_2(X)\,\tilde{\ell}_3(X)$$

with $\tilde{\zeta}(X) > 0$ when $X \in M_3^{\sim d}$.

The 'volume' function, defined on $\mathcal{M}_3$, is

$$\tilde{v}(X) \equiv det(X)$$

If $X \in \mathcal{M}_3^p$, then $\tilde{v}(X) > 0$.

If X is non-degenerate, there exist the unit vectors

$$\hat{\mathbf{x}}_i \equiv \frac{\mathbf{x}_i}{\tilde{\ell}_i(X)}$$

The unit vectors allow the definition of three 'skew angle' extraction functions $\tilde{\phi}_{12}(X)$, $\tilde{\phi}_{13}(X)$, and $\tilde{\phi}_{23}(X)$ by

$$\begin{aligned} \cos\tilde{\phi}_{12} &\equiv \hat{\mathbf{x}}_1 \cdot \hat{\mathbf{x}}_2 \\ \cos\tilde{\phi}_{13} &\equiv \hat{\mathbf{x}}_1 \cdot \hat{\mathbf{x}}_3 \\ \cos\tilde{\phi}_{23} &\equiv \hat{\mathbf{x}}_2 \cdot \hat{\mathbf{x}}_3 \end{aligned}$$

and by

$$
\begin{aligned}
\sin\tilde{\phi}_{12} &= |\hat{\mathbf{x}}_1 \times \hat{\mathbf{x}}_2| \\
\sin\tilde{\phi}_{13} &= |\hat{\mathbf{x}}_1 \times \hat{\mathbf{x}}_3| \\
\sin\tilde{\phi}_{23} &= |\hat{\mathbf{x}}_2 \times \hat{\mathbf{x}}_3|
\end{aligned}
$$

With this definition, the skew angles obey $0 \leq \tilde{\phi}_{ij} \leq \pi$. If, in addition, X is non-singular, then $0 < \tilde{\phi}_{ij} < \pi$.

If X is *non-singular*, one can define the dihedral angle extraction function $\tilde{\chi}(X)$ given by

$$
\begin{aligned}
\cos\tilde{\chi} &\equiv \frac{\mathbf{x}_1 \times \mathbf{x}_2}{|\mathbf{x}_1 \times \mathbf{x}_2|} \cdot \frac{\mathbf{x}_1 \times \mathbf{x}_3}{|\mathbf{x}_1 \times \mathbf{x}_3|} \\
&= \frac{\cos\tilde{\phi}_{23} - \cos\tilde{\phi}_{12}\cos\tilde{\phi}_{13}}{\sin\tilde{\phi}_{12}\sin\tilde{\phi}_{13}} \\
\sin\tilde{\chi} &= \frac{det(X)}{\tilde{\zeta}\,\sin\tilde{\phi}_{12}\sin\tilde{\phi}_{13}}
\end{aligned}
$$

The dihedral angle is well-defined provided $\mathbf{x}_1 \times \mathbf{x}_2 \neq 0$ and $\mathbf{x}_1 \times \mathbf{x}_3 \neq 0$. Since X is non-singular, this is guaranteed. Also in that case, $\tilde{\zeta} > 0$, $\sin\tilde{\phi}_{12} > 0$ and $\sin\tilde{\phi}_{13} > 0$, and thus $\sin\tilde{\chi} > 0$ if and only if $det(X) > 0$.

For X non-degenerate, define three 'aspect ratio' functions $\tilde{\rho}_1(X)$, $\tilde{\rho}_2(X)$, and $\tilde{\rho}_3(X)$ by

$$
\begin{aligned}
\tilde{\rho}_1(X) &= \frac{\tilde{\ell}_1}{\sqrt{\tilde{\ell}_2\tilde{\ell}_3}} \\
\tilde{\rho}_2(X) &= \frac{\tilde{\ell}_2}{\sqrt{\tilde{\ell}_3\tilde{\ell}_1}} \\
\tilde{\rho}_3(X) &= \frac{\tilde{\ell}_3}{\sqrt{\tilde{\ell}_1\tilde{\ell}_2}}
\end{aligned}
$$

With these definitions of aspect ratio one finds

$$
\tilde{\rho}_1\,\tilde{\rho}_2\,\tilde{\rho}_3 = 1
$$

and

$$
\tilde{\ell}_i = \left(\tilde{\zeta}\,\tilde{\rho}_i^2\right)^{\frac{1}{3}}
$$

There are three scalar extraction functions which define the *orientation* of X. The first two extraction functions, $\tilde{\theta}(X)$ and $\tilde{\psi}(X)$, are introduced by writing the unit vector $\hat{\mathbf{x}}_1$ in spherical coordinates:

$$\hat{\mathbf{x}}_1 = \begin{bmatrix} \cos\tilde{\theta}\sin\tilde{\psi} \\ \sin\tilde{\theta}\sin\tilde{\psi} \\ \cos\tilde{\psi} \end{bmatrix}$$

The two scalar extraction functions are thus given by

$$\cos\tilde{\theta}(X) = \frac{X_{11}}{\sqrt{X_{11}^2 + X_{21}^2}}$$
$$\sin\tilde{\theta}(X) = \frac{X_{21}}{\sqrt{X_{11}^2 + X_{21}^2}}$$

and

$$\cos\tilde{\psi}(X) = \frac{X_{31}}{\tilde{\ell}_1(X)}$$
$$\sin\tilde{\psi}(X) = \frac{\sqrt{X_{11}^2 + X_{21}^2}}{\tilde{\ell}_1(X)}$$

Finally, we define the extraction function $\tilde{\beta}(X)$ by

$$\cos\tilde{\beta}(X) \equiv \tilde{\mathbf{r}}_2(X) \cdot \tilde{\mathbf{a}}^{\perp}(X)$$
$$\sin\tilde{\beta}(X) \equiv \tilde{\mathbf{r}}_3(X) \cdot \tilde{\mathbf{a}}^{\perp}(X)$$

with extraction vectors

$$\tilde{\mathbf{r}}_2(X) = \frac{\hat{\mathbf{x}}_2 - (\cos\tilde{\phi}_{12})\hat{\mathbf{x}}_1}{\sin\tilde{\phi}_{12}}$$
$$\tilde{\mathbf{r}}_3(X) = \frac{\hat{\mathbf{x}}_1 \times \hat{\mathbf{x}}_2}{\sin\tilde{\phi}_{12}}$$

and

$$\tilde{\mathbf{a}}^{\perp}(X) \equiv \begin{pmatrix} -\sin\tilde{\theta}(X) \\ +\cos\tilde{\theta}(X) \\ 0 \end{pmatrix}$$

Scalar extraction functions $\tilde{\ell}_i(X)$, $\tilde{\zeta}(X)$, $\tilde{\upsilon}(X)$, $\tilde{\phi}_{12}(X)$, $\tilde{\phi}_{13}(X)$, $\tilde{\phi}_{23}(X)$, $\tilde{\chi}(X)$, $\rho_i(X)$, $\tilde{\theta}(X)$, $\tilde{\psi}(X)$, and $\tilde{\beta}(X)$ have been defined for the case $d = 3$.

All of these functions are well-defined when X is non-singular. These functions correspond to geometric properties of the three column vectors of X: length, size, volume, skew, aspect ratio, and orientation.

Table 2.2 shows the range or image of each extraction function as X ranges over four commonly used matrix sets. The images are intervals $[I_k]$, $k = 1, 2, \ldots, 8$ on the real line. In the Table, 'U' means that there exist elements of the matrix set for which the selected extraction function is undefined; in that case, the image is also undefined. The image of each matrix set under the extraction vector $\tilde{\mathbf{p}}_3$ is the Cartesian Product of the intervals in the appropriate column of the table.

Parameter	$X \in \mathcal{M}_3^p$	$X \in \mathcal{M}_3^{\sim s}$	$X \in \mathcal{M}_3^{\sim d}$	$X \in \mathcal{M}_3$
$\tilde{v}(X)$	$[I_2]$	$[I_8]$	$[I_1]$	$[I_1]$
$\tilde{\phi}_{12}(X)$	$[I_4]$	$[I_4]$	$[I_7]$	U
$\tilde{\phi}_{13}(X)$	$[I_4]$	$[I_4]$	$[I_7]$	U
$\tilde{\phi}_{23}(X)$	$[I_4]$	$[I_4]$	$[I_7]$	U
$\tilde{\chi}(X)$	$[I_4]$	$[I_4]$	U	U
$\tilde{\rho}_i(X)$	$[I_2]$	$[I_2]$	$[I_2]$	U
$\tilde{\theta}(X)$	$[I_5]$	$[I_5]$	$[I_5]$	U
$\tilde{\psi}(X)$	$[I_6]$	$[I_6]$	$[I_6]$	U
$\tilde{\beta}(X)$	$[I_6]$	$[I_6]$	U	U

Table 2.2: Range of Each Geometric Parameter Given $X_{3\times 3}$.

Intervals: $[I_1] = (-\infty, \infty)$; $[I_2] = (0, \infty)$; $[I_3](a) = (a - \pi, a + \pi]$; $[I_4] = (0, \pi)$; $[I_5] = [0, 2\pi)$; $[I_6] = [0, \pi)$; $[I_7] = [0, \pi]$; $[I_8] = (-\infty, 0) \cup (0, \infty)$.

2.2.3 Geometric Extraction Matrices for Volume Meshes

Given non-singular $X_{3\times 3} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$, we define the following extraction *matrices*, whose elements are given in terms of the previously-defined scalar extraction functions.

First, let $\tilde{R}(X) = [\tilde{\mathbf{r}}_1(X), \tilde{\mathbf{r}}_2(X), \tilde{\mathbf{r}}_3(X)]$ be the rotation matrix which extracts *orientation* information from X. The column vectors of $\tilde{R}$ are the extraction vectors

$$\begin{aligned}
\tilde{\mathbf{r}}_1(X) &= \hat{\mathbf{x}}_1 \\
\tilde{\mathbf{r}}_2(X) &= \frac{\hat{\mathbf{x}}_2 - (\cos\tilde{\phi}_{12})\hat{\mathbf{x}}_1}{\sin\tilde{\phi}_{12}} \\
\tilde{\mathbf{r}}_3(X) &= \frac{\hat{\mathbf{x}}_1 \times \hat{\mathbf{x}}_2}{\sin\tilde{\phi}_{12}}
\end{aligned}$$

This matrix can be factored into two other extraction matrices such that

$$\tilde{R}(X) = \tilde{R}_2(X)\tilde{R}_1(X)$$

with

$$\tilde{R}_1(X) \equiv \left[\hat{\mathbf{x}}_1,\, \tilde{\mathbf{a}}^{\perp},\, \hat{\mathbf{x}}_1 \times \tilde{\mathbf{a}}^{\perp}\right]$$

and

$$\tilde{R}_2(X) \equiv \left[\cos\tilde{\beta}(X)\right] I_3 + \left[1 - \cos\tilde{\beta}(X)\right](\hat{\mathbf{x}}_1 \otimes \hat{\mathbf{x}}_1) - \left[\sin\tilde{\beta}(X)\right]\varepsilon(\hat{\mathbf{x}}_1)$$

$\varepsilon(\mathbf{v})$ is the matrix

$$\varepsilon(\mathbf{v}) = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix}$$

where $\mathbf{v} = (v_x, v_y, v_z)$ is any vector.

Second, let $\tilde{U}(X)$ be the upper triangular matrix which extracts *shape+size* information from X. Explicitly, in terms of the scalar extraction functions

$$\tilde{U}(X) = \begin{pmatrix} \tilde{\ell}_1(X) & \tilde{\ell}_2(X)\cos[\tilde{\phi}_{12}(X)] & \tilde{\ell}_3(X)\cos[\tilde{\phi}_{13}(X)] \\ 0 & \tilde{\ell}_2(X)\sin[\tilde{\phi}_{12}(X)] & \tilde{\ell}_3(X)\sin[\tilde{\phi}_{13}(X)]\cos[\tilde{\chi}(X)] \\ 0 & 0 & \tilde{\ell}_3(X)\sin[\tilde{\phi}_{13}(X)]\sin[\tilde{\chi}(X)] \end{pmatrix}$$

The shape+size extraction matrix can be factored as

$$\tilde{U}(X) = \left[\tilde{\zeta}(X)\right]^{\frac{1}{3}} \, \tilde{S}(X)$$

where $\tilde{S}(X)$ is the *shape* extraction matrix given by

$$\tilde{S}(X) = \begin{pmatrix} [\tilde{\rho}_1(X)]^{\frac{2}{3}} & [\tilde{\rho}_2(X)]^{\frac{2}{3}} \cos[\tilde{\phi}_{12}(X)] & [\tilde{\rho}_3(X)]^{\frac{2}{3}} \cos[\tilde{\phi}_{13}(X)] \\ 0 & [\tilde{\rho}_2(X)]^{\frac{2}{3}} \sin[\tilde{\phi}_{12}(X)] & [\tilde{\rho}_3(X)]^{\frac{2}{3}} \sin[\tilde{\phi}_{13}(X)] \cos[\tilde{\chi}(X)] \\ 0 & 0 & [\tilde{\rho}_3(X)]^{\frac{2}{3}} \sin[\tilde{\phi}_{13}(X)] \sin[\tilde{\chi}(X)] \end{pmatrix}$$

In turn, the shape extraction matrix can be factored into *skew* and *aspect ratio* extraction matrices as

$$\tilde{S}(X) = \tilde{Q}(X) \, \tilde{D}(X)$$

with skew extraction matrix

$$\tilde{Q}(X) = \begin{pmatrix} 1 & \cos[\tilde{\phi}_{12}(X)] & \cos[\tilde{\phi}_{13}(X)] \\ 0 & \sin[\tilde{\phi}_{12}(X)] & \sin[\tilde{\phi}_{13}(X)] \cos[\tilde{\chi}(X)] \\ 0 & 0 & \sin[\tilde{\phi}_{13}(X)] \sin[\tilde{\chi}(X)] \end{pmatrix}$$

and aspect ratio extraction matrix

$$\tilde{D}(X) = \begin{pmatrix} [\tilde{\rho}_1(X)]^{\frac{2}{3}} & 0 & 0 \\ 0 & [\tilde{\rho}_2(X)]^{\frac{2}{3}} & 0 \\ 0 & 0 & [\tilde{\rho}_3(X)]^{\frac{2}{3}} \end{pmatrix}$$

From these results,

$$\begin{aligned} X &= \tilde{R}(X) \, \tilde{U}(X) \\ &= \left[\tilde{\zeta}(X)\right]^{\frac{1}{3}} \, \tilde{R}(X) \, \tilde{S}(X) \\ &= \left[\tilde{\zeta}(X)\right]^{\frac{1}{3}} \, \tilde{R}(X) \, \tilde{Q}(X) \, \tilde{D}(X) \end{aligned}$$

We shall call these the 3D *geometric extraction identities.*

2.2.4 A Parametric Definition of 3×3 Matrices

There are nine geometric parameters required to define a matrix $Y_{3\times3}$ representing a target matrix. The nine parameters are taken to be those in $\mathbf{p}_3$.

First, given the parameters (θ, ψ, β) in $\mathbf{p}_3^{gri}$, define a rotation matrix $R = R(\theta, \psi, \beta) = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ to indicate the *orientation* of matrix Y. The parameters (θ, ψ) are to be the spherical coordinates of $\mathbf{r}_1$; thus

$$\mathbf{r}_1 = \begin{pmatrix} \cos\theta \sin\psi \\ \sin\theta \sin\psi \\ \cos\psi \end{pmatrix}$$

with the requirement that $0 \leq \psi \leq \pi$ and $0 \leq \theta < 2\,\pi$. To define the vectors $\mathbf{r}_2$ and $\mathbf{r}_3$, use the projection of $\mathbf{r}_1$ onto the xy-plane to create the unit vector $\mathbf{a} = (\cos\theta, \sin\theta, 0)$. Define $\mathbf{a}^\perp = \hat{\mathbf{k}} \times \mathbf{a}$, so that

$$\mathbf{a}^\perp = (-\sin\theta, \cos\theta, 0)$$

Then $\mathbf{r}_1 \cdot \mathbf{a}^\perp = 0$. Define unit vector $\mathbf{n} = \mathbf{r}_1 \times \mathbf{a}^\perp$. The matrix $R_1(\theta, \psi) = [\mathbf{r}_1, \mathbf{a}^\perp, \mathbf{n}]$ is a rotation. Explicitly, the elements of R_1 are

$$R_1(\theta, \psi) = \begin{pmatrix} \cos\theta \sin\psi & -\sin\theta & -\cos\theta\cos\psi \\ \sin\theta \sin\psi & \cos\theta & -\sin\theta\cos\psi \\ \cos\psi & 0 & \sin\psi \end{pmatrix}$$

This matrix depends only on (θ, ψ) and is thus not completely general.

To define a fully general rotation describing the orientation of Y, introduce another rotation R_2 using the parameter β, given by

$$R_2(\beta) \equiv (\cos\beta) I_3 + (1 - \cos\beta)\,(\mathbf{r}_1 \otimes \mathbf{r}_1) - (\sin\beta)\varepsilon(\mathbf{r}_1)$$

R_2 shall be used to rotate R_1 about the axis $\mathbf{r}_1$. This yields the final orientation matrix $R(\theta, \psi, \beta) = R_2(\beta)\, R_1(\theta, \psi)$. We have

$$\begin{aligned} R(\theta, \psi, \beta) &= R_2\, R_1 \\ &= R_2\, \left[\mathbf{r}_1, \mathbf{a}^\perp, \mathbf{n}\right] \\ &= \left[R_2\, \mathbf{r}_1, R_2\, \mathbf{a}^\perp, R_2\, \mathbf{n}\right] \\ &= \left[\mathbf{r}_1, (\cos\beta)\mathbf{a}^\perp - (\sin\beta)\mathbf{n}, (\sin\beta)\mathbf{a}^\perp + (\cos\beta)\mathbf{n}\right] \\ &= \begin{pmatrix} \cos\theta\sin\psi & -\sin\theta\cos\beta + \cos\theta\cos\psi\sin\beta & -\sin\theta\sin\beta - \cos\theta\cos\psi\cos\beta \\ \sin\theta\sin\psi & \cos\theta\cos\beta + \sin\theta\cos\psi\sin\beta & \cos\theta\sin\beta - \sin\theta\cos\psi\cos\beta \\ \cos\psi & -\sin\psi\sin\beta & \sin\psi\cos\beta \end{pmatrix} \end{aligned}$$

The vectors $\mathbf{r}_2$, $\mathbf{r}_3$, and $\mathbf{a}^\perp$ all belong to the plane defined by $\mathbf{r}_1$. Further, $\mathbf{r}_2 \cdot \mathbf{a}^\perp = \cos\beta$ and $\mathbf{r}_3 \cdot \mathbf{a}^\perp = \sin\beta$. The orientation matrix R is related to the matrix Y by $Y = RU$, with the latter yet to be defined.

The last six parameters, ζ, ϕ_{12}, ϕ_{13}, χ, ρ_1, and ρ_2, are called *shape+size* parameters. These are used to define a matrix $U_{3\times 3}$ representing the shape and size of the column vectors of Y. Specifically, define U by

$$U(\zeta, \phi_{12}, \phi_{13}, \chi, \rho_1, \rho_2) = \begin{pmatrix} \ell_1 & \ell_2 \cos\phi_{12} & \ell_3 \cos\phi_{13} \\ 0 & \ell_2 \sin\phi_{12} & \ell_3 \sin\phi_{13} \cos\chi \\ 0 & 0 & \ell_3 \sin\phi_{13} \sin\chi \end{pmatrix}$$

with lengths $\ell_i \equiv \left(\zeta\rho_i^2\right)^{\frac{1}{3}}$ and $\rho_1\,\rho_2\,\rho_3 = 1$. The parameters must obey $\zeta > 0$, $0 < \phi_{12} < \pi$, $0 < \phi_{13} < \pi$, $0 < \chi < \pi$, and $\rho_i > 0$, so that $det(U) > 0$.

Finally, the matrix Y representing the target-matrix is defined as

$$Y \equiv RU$$

Expressing Y in terms of its column vectors we have $Y = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3]$, with

$$\begin{aligned} \mathbf{y}_1 &= \ell_1\,\mathbf{r}_1 \\ \mathbf{y}_2 &= (\ell_2 \cos\phi_{12})\,\mathbf{r}_1 + (\ell_2 \sin\phi_{12})\,\mathbf{r}_2 \\ \mathbf{y}_3 &= (\ell_3 \cos\phi_{13})\,\mathbf{r}_1 + (\ell_3 \sin\phi_{13} \cos\chi)\,\mathbf{r}_2 + (\ell_3 \sin\phi_{13} \sin\chi)\,\mathbf{r}_3 \end{aligned}$$

These results give the standard parametric definition of $Y_{3\times 3}$ in terms of $\mathbf{p}_3$. In terms of the mapping $\mathbf{F}$ we have $Y = \mathbf{F}(\mathbf{p}_3)$.

Proposition.
Given $Y_{3\times 3}$ defined as above, Y is non-degenerate and non-singular.
Proof.
Since $Y = RU$, the determinant of Y is $det(Y) = (detR)(detU) = det(U)$. But $det(U) = \zeta \sin\phi_{12} \sin\phi_{13} \sin\chi = \upsilon$. For the required range of these parameters, υ is positive and therefore Y is non-singular. Since Y is non-singular, it is also non-degenerate. §

Proposition.
Evaluating the nine extraction functions defined in Section 2.2.2 on the Y defined above, we find that the nine geometric target parameters are recovered.
Proof.
First, we have $\tilde{\ell}_i(Y) = \mid \mathbf{y}_i \mid = \ell_i$. Then $\tilde{\zeta}(Y) = \tilde{\ell}_1(Y)\tilde{\ell}_2(Y)\tilde{\ell}_3(Y) = \ell_1\ell_2\ell_3 =$

ζ. Next,

$$\begin{aligned}
\hat{\mathbf{y}}_1 &= \mathbf{r}_1 \\
\hat{\mathbf{y}}_2 &= (\cos\phi_{12})\,\mathbf{r}_1 + (\sin\phi_{12})\,\mathbf{r}_2 \\
\hat{\mathbf{y}}_3 &= (\cos\phi_{13})\,\mathbf{r}_1 + (\sin\phi_{13}\cos\chi)\,\mathbf{r}_2 + (\sin\phi_{13}\sin\chi)\,\mathbf{r}_3
\end{aligned}$$

Therefore,

$$\begin{aligned}
\cos\tilde{\phi}_{12}(Y) &= \hat{\mathbf{y}}_1 \cdot \hat{\mathbf{y}}_2 \\
&= \cos\phi_{12} \\
\cos\tilde{\phi}_{13}(Y) &= \hat{\mathbf{y}}_1 \cdot \hat{\mathbf{y}}_3 \\
&= \cos\phi_{13} \\
\cos\tilde{\chi}(Y) &= \frac{\mathbf{y}_1 \times \mathbf{y}_2}{|\mathbf{y}_1 \times \mathbf{y}_2|} \cdot \frac{\mathbf{y}_1 \times \mathbf{y}_3}{|\mathbf{y}_1 \times \mathbf{y}_3|} \\
&= \mathbf{r}_3 \cdot [(\cos\chi)\,\mathbf{r}_3 - (\sin\chi)\mathbf{r}_2] \\
&= \cos\chi
\end{aligned}$$

and

$$\begin{aligned}
\sin\tilde{\phi}_{12}(Y) &= |\hat{\mathbf{y}}_1 \times \hat{\mathbf{y}}_2| \\
&= \sin\phi_{12} \\
\sin\tilde{\phi}_{13}(Y) &= |\hat{\mathbf{y}}_1 \times \hat{\mathbf{y}}_3| \\
&= \sin\phi_{13} \\
\sin\tilde{\chi}(Y) &= \frac{det(Y)}{\tilde{\zeta}(Y)\left[\sin\tilde{\phi}_{12}(Y)\right]\left[\sin\tilde{\phi}_{13}(Y)\right]} \\
&= \sin\chi
\end{aligned}$$

The three aspect ratio extraction functions applied to Y give

$$\begin{aligned}
\tilde{\rho}_i(Y) &\equiv \frac{\tilde{\ell}_i(Y)}{\sqrt{\tilde{\ell}_{i+1}(Y)\tilde{\ell}_{i+2}(Y)}} \\
&= \frac{\ell_i}{\sqrt{\ell_{i+1}\ell_{i+2}}} \\
&= \frac{(\zeta\rho_i^2)^{\frac{1}{3}}}{\sqrt{(\zeta\rho_{i+1}^2)^{\frac{1}{3}}(\zeta\rho_{i+2}^2)^{\frac{1}{3}}}} \\
&= \frac{\rho_i^{\frac{2}{3}}}{\rho_{i+1}^{\frac{1}{3}}\rho_{i+2}^{\frac{1}{3}}} \\
&= \rho_i
\end{aligned}$$

The orientation of Y is recovered as follows:

$$\begin{aligned}
\cos\tilde{\theta}(Y) &= \frac{Y_{11}}{\sqrt{Y_{11}^2 + Y_{21}^2}} \\
&= \frac{\cos\theta\sin\psi}{\sqrt{(\cos\theta\sin\psi)^2 + (\sin\theta\sin\psi)^2}} \\
&= \cos\theta
\end{aligned}$$

Similarly, $\sin\tilde{\theta}(Y) = \sin\theta$, $\cos\tilde{\psi}(Y) = \cos\psi$, and $\sin\tilde{\psi}(Y) = \sin\psi$.

Lastly,

$$\begin{aligned}
\cos\tilde{\beta}(Y) &= \tilde{\mathbf{r}}_2(Y)\cdot\tilde{\mathbf{a}}^{\perp}(Y) \\
&= \frac{\hat{\mathbf{y}}_2 - \left[\cos\tilde{\phi}_{12}(Y)\right]\hat{\mathbf{y}}_1}{\sin\tilde{\phi}_{12}(Y)}\cdot\tilde{\mathbf{a}}^{\perp}(Y) \\
&= \frac{\hat{\mathbf{y}}_2 - [\cos\phi_{12}(Y)]\,\hat{\mathbf{y}}_1}{\sin\phi_{12}(Y)}\cdot\tilde{\mathbf{a}}^{\perp}(Y) \\
&= \mathbf{r}_2\cdot\tilde{\mathbf{a}}^{\perp}(Y) \\
&= \mathbf{r}_2\cdot\mathbf{a}^{\perp} \\
&= \cos\beta
\end{aligned}$$

and

$$\begin{aligned}
\sin\tilde{\beta}(Y) &= \tilde{\mathbf{r}}_3(Y)\cdot\tilde{\mathbf{a}}^{\perp}(Y) \\
&= \frac{\hat{\mathbf{y}}_1\times\hat{\mathbf{y}}_2}{\sin\tilde{\phi}_{12}(Y)}\cdot\tilde{\mathbf{a}}^{\perp}(Y) \\
&= \frac{\hat{\mathbf{y}}_1\times\hat{\mathbf{y}}_2}{\sin\phi_{12}}\cdot\mathbf{a}^{\perp} \\
&= \mathbf{r}_3\cdot\mathbf{a}^{\perp} \\
&= \sin\beta
\end{aligned}$$

Thus, all nine parameters are recovered by the extraction functions when applied to Y. §

The parametric definition is consistent with the results of applying the scalar extraction functions to the target matrix. In terms of the mapping, $\tilde{\mathbf{F}}(Y) = \mathbf{p}_3$.

2.2.5 The Parametric Matrix Factors of $Y_{3\times3}$

The parametric representation of Y in the Section 2.2.4 has the following factorization

$$Y(\mathbf{p}_3^{sso}) = \zeta^{\frac{1}{3}}\, R(\mathbf{p}_3^{ori})\, Q(\mathbf{p}_3^{skw})\, D(\mathbf{p}_3^{asp}) \tag{2.4}$$

where $R(\mathbf{p}_3^{ori}) = R(\theta, \psi, \beta)$ and

$$R(\theta, \psi, \beta) \quad = \quad R_2(\beta)\, R_1(\theta, \psi)$$

and $Q(\mathbf{p}_3^{skw}) = Q(\phi_{12}, \phi_{13}, \chi)$ with

$$Q\,(\phi_{12}, \phi_{13}, \chi) = \begin{pmatrix} 1 & \cos\phi_{12} & \cos\phi_{13} \\ 0 & \sin\phi_{12} & \sin\phi_{13}\cos\chi \\ 0 & 0 & \sin\phi_{13}\sin\chi \end{pmatrix}$$

and $D(\mathbf{p}_3^{asp}) = D(\rho_1, \rho_2, \rho_3)$

$$D(\rho_1, \rho_2, \rho_3) = \begin{pmatrix} \rho_1^{\frac{2}{3}} & 0 & 0 \\ 0 & \rho_2^{\frac{2}{3}} & 0 \\ 0 & 0 & \rho_3^{\frac{2}{3}} \end{pmatrix}$$

The result of taking the determinant of both sides of the factorization relation is $\upsilon = det(Y)$, with

$$\upsilon = \zeta\, \sin\phi_{12}\sin\phi_{13}\sin\chi$$

Note that (1) the three orientation matrices R are each rotations, (2) Q is upper triangular with unit column vectors, and (3) D is diagonal and positive definite. Define the *shape* matrix $S(\mathbf{p}_3^{shp}) = Q(\mathbf{p}_3^{skw})D(\mathbf{p}_3^{asp})$ and the *shape+size* matrix $U(\mathbf{p}_3^{shs}) = \zeta^{\frac{1}{3}} S(\mathbf{p}_3^{shp})$, giving

$$Y \quad = \quad R(\theta, \phi, \beta) U(\zeta, \phi_{12}, \phi_{13}, \chi, \rho_1, \rho_2)$$

Proposition.
From the last proposition in Section 2.2.4, one can show that the extraction matrices applied to Y as constructed in this section obey $\tilde{R}(Y) = R(\theta, \psi, \beta)$, $\tilde{Q}(Y) = Q(\phi_{12}, \phi_{23}, \chi)$, $\tilde{D}(Y) = D(\rho_1, \rho_2, \rho_3)$, etc.

2.2.6 Matrix Types and Sets of 3×3 Matrices

The definitions of the matrix sets $\mathcal{M}_2^{rot}$, $\mathcal{M}_2^{skw}$, etc. given in Section 2.1.6 are easily extended to three dimensions. Thus, define the sets $\mathcal{M}_3^{rot}$, $\mathcal{M}_3^{skw}$, $\mathcal{M}_3^{asp}$, $\mathcal{M}_3^{shp}$, and $\mathcal{M}_3^{shs}$. Then $R(\theta, \psi, \beta) \in \mathcal{M}_3^{rot}$ is a rotation matrix, $Q(\phi_{12}, \phi_{13}, \chi) \in \mathcal{M}_3^{skw}$ is a skew matrix, etc.

2.2.7 Equality of Geometric Parameters for Volume Meshes

The following Propositions justify many of the local quality volume metrics used in TMOP. Each metric depends on the active matrix and a target. The matrices X and Y below are placeholders for these two matrices, respectively.

Proposition.
Let $X, Y \in \mathcal{M}_3^{\sim s}$ and let Y have the factorization in Section 2.2.4. If $\tilde{Q}(X) = Q(\phi_{12}, \phi_{13}, \chi)$, then $\tilde{\phi}_{12}(X) = \phi_{12}$, $\tilde{\phi}_{13}(X) = \phi_{13}$, and $\tilde{\chi}(X) = \chi$.
Proof.
The proof equates each element of $\tilde{Q}(X)$ to the corresponding element of $Q(\phi_{12}, \phi_{13}, \chi)$. The result follows quickly. §

Similarly, if $\tilde{R}(X) = R(\theta, \psi, \beta)$, then $\tilde{\theta}(X) = \theta$, $\tilde{\psi}(X) = \psi$, and $\tilde{\beta}(X) = \beta$.

Proposition.
Let $X, Y \in \mathcal{M}_3^{\sim s}$ and let Y have the factorization in Section 2.2.4. If $\tilde{S}(X) = S(\phi_{12}, \phi_{13}, \chi, \rho_1, \rho_2, \rho_3)$, then $\tilde{\phi}_{12}(X) = \phi_{12}$, $\tilde{\phi}_{13}(X) = \phi_{13}$, $\tilde{\chi}(X) = \chi$, $\tilde{\rho}_i(X) = \rho_i$.

Corollary. If $\tilde{S}(X) = S(\phi_{12}, \phi_{13}, \chi, \rho_1, \rho_2, \rho_3)$, then $\tilde{Q}(X) = Q(\phi_{12}, \phi_{13}, \chi)$ and $\tilde{D}(X) = D(\rho_1, \rho_2, \rho_3)$.

Similar propositions can be given for the shape+size and other matrices such the product of the rotation and skew matrices.

2.2.8 Application to the 3×3 Active and Target-Matrices

The previous developments are applied to the 3×3 active and target-matrices in TMOP.

The target-matrix W can be usefully represented parametrically, as in Section 2.2.4, given the parameters in $\mathbf{p}_3$. Values of these parameters must be supplied at every sample point in order to construct the set $\{\mathcal{W}\}$ prior to mesh quality assessment and improvement. The process of determining these values is called *target-parameter construction.* Each target-matrix is required to be in $\mathcal{M}_3^p$; therefore, $\upsilon > 0$ is required. By definition, $\mathbf{p}_3 \in W^9 \subset \Re^9$, where W^9 is the Cartesian product of the nine intervals in the first column of Table 2.2. In terms of the mapping discussed at the beginning of this chapter, one has $\mathcal{S}_W = \mathcal{M}_3^p$ and $\mathcal{P}_W = W^9$, which results in the one-to-one, onto mapping $\mathbf{F}$.

Unlike the target-matrix, the active matrix is not constructed; rather, it is given. The only thing we can be sure of is that $A \in \mathcal{M}_3$. But, if A is degenerate, parameters $\tilde{\chi}(A)$ and $\tilde{\beta}(A)$ are undefined. Although the occurrence of degenerate A may be rare, robust practical algorithms must account for this possibility in some way. Thus, $\mathcal{S}_A$ is restricted to $\mathcal{M}_3^{\sim s}$. The mapping $\tilde{\mathbf{F}}$ is then well-defined, with $\mathcal{P}_A$ being the Cartesian product of the nine intervals in the second column of Table 2.2.

Chapter 3

Target Construction

To review briefly, in the previous chapter a standard set of geometric parameters $\mathbf{p}$ was identified for both planar and volume meshes. These parameters can be grouped into (i) volume/size, (ii) orientation, (iii) skew, and (iv) aspect ratio parameters. Values of these geometric parameters at every sample point can be obtained by applying 'extraction functions' to the active mesh Jacobian matrix. Such parameters are called *active* mesh parameters. Further, given 'target' values of the geometric parameters at the sample points, one has *target* mesh parameters and can obtain target-matrices $W(\mathbf{p})$, $R(\mathbf{p}^{ori})$, $U(\mathbf{p}^{shs})$, if needed. TMOP requires the existence, at every sample point in the mesh, of values of the active mesh parameters and corresponding values of the target mesh parameters. The values of these parameters and/or their corresponding matrices are used to evaluate the local mesh quality metrics and, in numerical mesh optimization, the objective function. The values of the target mesh parameters must be determined before the numerical mesh optimization procedure can begin.

A critical issue in TMOP is to understand, in a general way, how to assign values to the target mesh parameters. In the ideal situation, one can use either a published method to assign the values or an existing algorithm in a mesh optimization code. If neither is available, one can consider developing their own method for assigning the target parameter values. This chapter describes the process by which a new method for assigning the target parameter values can be devised.[1] We call this process *Target Construction*

[1] It is expected that, over time, more and more methods for assigning target parameter values will be developed so that one will not have to develop a new method each time a mesh is to be optimized.

(or Target-Parameter Construction).

Definition.
Target Construction is a process by which one develops a numerical algorithm which takes raw data and converts it into values of the target parameters at every mesh sample point.[2] The process takes into account the mesh quality improvement goal, facts about the mesh, and the simulation. The resulting numerical algorithm is called a *Target Construction Method*.

One can, of course, develop a Target Construction Method without following the process to be described. However, a general description of the Target Construction process should help those who wish to create an effective method appropriate to the given mesh quality improvement goal and the problem context.

At the present time there exists only a few proven Target Construction Methods (e.g. shape improvement and shape improvement with size equidistribution). In the long term, one or more Target Construction Methods for each of the mesh quality improvement goals listed in Section 1.1 are envisioned. In this vision, the practitioner with a mesh quality issue would be able to consult the list of quality improvement goals to find an appropriate target construction method which they could use directly or adapt to their specific problem. In the meantime, barring the existence of an appropriate Target Construction Method, one must devise their own method. The purpose of this chapter is to assist and encourage the development of new methods.

Three sequential phases within Target Construction are proposed:

1. Target Construction Strategy
2. Intermediate Data Algorithms
3. Target Parameter Model Development

In Target Construction Strategy one develops an approach to the problem of obtaining values of the target parameters. In this phase one assesses the quality of the initial mesh to be improved, formulates a mesh quality improvement goal, determines the relevant target parameters and how they will be treated, and examines the raw data. At the end of this stage, one should have a plan or strategy as to how the mesh quality improvement

[2] We shall discuss what we mean by *raw data* in Section 3.1.3.

goal will be addressed and how the values for the target parameters will be obtained. In the Intermediate Data Algorithms phase one determines the algorithms by which raw data will be converted into intermediate data at the mesh sample points. In the last phase, one develops a model or algorithm by which the intermediate data will be converted into values of the target parameters. These stages are described in Sections 3.1, 3.2, and 3.3.

3.1 Target Construction Strategy

In the first phase of Target Construction one develops an overall Target Construction Strategy to determine the mesh quality improvement goal and to sketch an approach to the problem of assigning appropriate values of the target parameters. It is assumed that at this point the practitioner has assessed the quality of their meshes, found them lacking, and determined a mesh quality improvement goal. He has determined that there is no appropriate Target Construction Method that already exists and has decided to develop a new method.

Strategy development itself entails a number of sub-phases which are described in 3.1.1 - 3.1.3. Section 3.1.1 considers the mesh and simulation context, i.e., facts about the mesh and the simulation that pertain to Target Construction. Section 3.1.2 discusses the Parameter Control Decision, in which it is determined what subset of the target geometric parameters should be controlled in order to achieve one's mesh quality improvement goal. Finally, Section 3.1.3 discusses the correspondences between particular raw, intermediate, and parameter data that need to be established.

3.1.1 Mesh and Simulation Context

As background information for the Target Construction process it is helpful to determine the mesh and simulation context. This information can bear on the approach selected at various stages in the Target Construction process.

The mesh context concerns basic facts about the initial mesh to be optimized. Because node-movement methods are to be used, the context is the same for the optimal mesh. A mesh consists of a collection of vertices, edges,

Mesh	Examples
Embedding	$\Re^2$, planar in $\Re^3$, surface in $\Re^3$, volume
Type	structured, unstructured conformal, non-conformal hybrid, non-hybrid static, moving high order/low order
Elements	triangle, quadrilateral, polygonal tetrahedral, hexahedral, prisms, pyramids, polyhedral

Table 3.1: Mesh Context Characterization

faces, and elements connected to one another. Table 3.1 shows basic information needed to characterize the mesh context. In addition, one might ask for the following information:

1. What is the typical number of mesh nodes or elements?
2. What are the mesh sample points?
3. Are there special regions in the domain which need to be treated differently from the rest?
4. Are there mesh nodes having a special valence?
5. Will the mesh nodes on the mesh boundary be allowed to move?
6. Does the mesh possess special fixed nodes in its interior?
7. Does the mesh have one or more internal interfaces to which nodes are constrained?
8. Does the optimal mesh need to be symmetric in some way?
9. Will negative Jacobian determinants at some initial mesh sample points occur?
10. In terms of mesh quality, what is good about the initial mesh and what is not so good?

The simulation context concerns basic facts about the simulation code and the particular simulation. Table 3.2 shows basic information needed to characterize the simulation context. In addition, one might ask for the following information:

Simulation	Examples
Physics	Thermal, Fluid, Electromagnetic
PDE	Elliptic, Parabolic, Navier-Stokes, Maxwell
Boundary Conds.	Dirichlet, Neuman, No-slip
Disc. Method	Finite Difference/Volume/Elements, Spectral
Dependent Variables	density, pressure, velocity, temperature
Functionals	Flux, Discretization Error, Lift

Table 3.2: Simulation Context Characterization

1. How does the problem domain change with time?
2. What PDE coefficients are involved and what is the spatial variation in their values?
3. How do the values of the boundary conditions vary spatially?
4. Is there a time-step restriction? If so, how is it defined?
5. What is roughly the size of the physical domain?
6. Is the mesh to be adapted to the physical solution and, if so, how?

Obtaining answers to these questions before engaging in Target Construction is a good idea because it can clarify the optimization goal, identify possible sources of raw data, and assist in target model development.

3.1.2 Target Parameter Control Decision

As noted previously, there are four groups of target parameters: volume, orientation, shape, and aspect ratio. In the volume mesh case, there are several target parameters within each of these groups. In this phase of developing a Target Construction Strategy, one decides which target parameter groups will be controlled and which will not. Parameters within a particular parameter group are *controlled* if values of the target parameters in the group are or will be assigned (at all the sample points) in Target Construction.[3] If, on the other hand, values are not assigned for a particular parameter

[3] If one wants to control a particular parameter within a parameter group, one will usually want to control all the other parameters in the group as well. There may be the rare occasions where one wants to control only some of the parameters within a group.

group, then the group is not controlled.

It would seem at first thought that one would always want to control all four parameter groups in order to achieve the highest mesh quality. The four groups are not equally important, however. Volume is perhaps the most important parameter, closely followed by skew. Aspect ratio is only important if the physics of the problem is anisotropic. Control over orientation is probably the least often required. Overall, the relative important of the groups depends on the particular simulation at hand and on the mesh optimization goal. In addition, assigning values at every mesh sample point to a parameter or parameter group can be challenging, especially when the physics does not dictate that the parameter needs to be controlled. Thus, it is fortunate that there exist mesh quality metrics which allow one to abstain from controlling a particular parameter group if desired. One cannot abstain on too many parameters at once, however, so a balance between controlling and abstaining is needed.

The decision to control or not control the parameter groups can be represented by a sequence of four letters, one for each of the groups, i.e., volume, orientation, skew, and aspect ratio. Each group can only be assigned the letters C or A, standing for 'control' and 'abstain' (i.e., not control). There are thus $2^4 = 16$ possible combinations of these letters, yielding 16 possible decisions one can make. If one only wants to control volume, for example, the Control Decision is represented as $CAAA$. The Control Decision determines (a) which target parameter groups need to be assigned values in Target Construction and (b) the metric type. Metric type refers to local quality metrics which control only the parameters one has decided to control; in the example $CAAA$ one controls only the volume parameter and thus the required metric type is a Volume Metric. Volume metrics are functions of the active and target parameters corresponding to v. There are thus 16 metric types. Some of the 16 metric types are not recommended for use in mesh optimization. About 5 metric types are very effective in improving mesh quality. This topic is discussed in detail in a subsequent report.

So, how does one make the Control Decision? Basically, the decision is made in light of the identified mesh quality improvement goal, the mesh and simulation context, and the available raw data. If, for example, the mesh quality improvement goal is to improve the shape of the mesh elements, then one needs to control the skew and aspect ratio parameters. Such a goal tacitly implies that the volume and orientation parameters are not particularly important (except for keeping the volume positive) and thus one can abstain on those parameters. The Control Decision for Shape Improvement is thus $AACC$. In general, one might wish to abstain on a parameter group because (i) it is unimportant with respect to the mesh quality improvement goal or (ii) it is important but there is no raw data that can be converted into appropriate values of the particular parameter. By abstaining on a particular parameter group, the optimization procedure should be able to produce an optimal mesh more in tune with the controlled parameters (and the quality improvement goal) than otherwise. Although a Control Decision always results in one of the 16 combinations of A and C, the decision should be restricted so that the corresponding metric type is one of the 5 effective types mentioned above.[4]

Once the Control Decision has been made one need consider only the controlled parameter groups in the rest of the Target Construction process.

3.1.3 Establishing Data Correspondences

As noted earlier, Target Construction results in an algorithm (or series of algorithms) which takes raw data and converts it into values of the target parameters. As an intermediate stage, one frequently converts the raw data into intermediate data and subsequently converts the intermediate data into target parameter data.

Definition.
Raw Data primarily consists of geometric data and simulation data that is available prior to Target Construction. See Table 3.3 for examples.

Definition.
Intermediate Data primarily consists of data which is computed from the raw data and can be considered as 'functionals' of the raw data. See Table 3.4 for examples.

Definition.
Parameter Data consists of values of the controlled target parameters over

[4] It is recognized that, in more complex situations, the mesh quality improvement goal may require one to make multiple Control Decisions, for example, one for each sub-region of the physical domain.

the set of sample points and which is computed from Intermediate Data.

In this section some ideas are presented concerning potential sources of raw data which may be accessible to the optimization code. Raw data generally comes from two kinds of sources: (1) geometric sources and (2) simulation sources. Geometric data sources consist of items that are available independently of any physical information and independently of the simulation which is to be performed using the mesh. This includes information and data concerning the physical and computational domain, the mesh type, the mesh topology, element type, the coordinates of the initial mesh, and the coordinates of any reference meshes. Simulation data sources consist of items that are available from physical information and the simulation itself. Simulation data can be used as raw data for target-matrix construction at the beginning of the simulation or as the simulation proceeds, while geometric data is generally available prior to the simulation. An exception to the latter might be the case in which the domain deforms with time.

3.1.3.1 Geometric Raw and Intermediate Data

There are three potential sources of geometric raw data. First, there is the physical domain on which the computation is to take place. The user has high-level knowledge of the domain concerning its dimension (2D/3D), topology, symmetries, sub-regions on which important physics takes place, and whether it is deforming or static. Although some of this information may be difficult to quantify, it may bear upon the algorithms used in constructing the target-matrices. In addition, the physical domain is represented discretely in the computer by nodes, curves, surfaces, volumes, and regions. This raw data can potentially be processed to find intermediate quantities such as a domain bounding box and centroid, a 'diameter', a rough measure of aspect ratio and/or orientation (with respect to the coordinate system), and finally, a set of domain sub-region indicators which indicate either geometric sub-regions or physical sub-regions.

The second potential source of geometric raw data is the initial mesh, i.e., the mesh whose quality is to be improved via node-movement (or other) techniques. The initial mesh contains raw data concerning the mesh type (structured/unstructured, planar/surface/volume, low/high order, conformal/non-conformal, moving/static, etc.), mesh connectivity, element type, the vertex coordinates, and whether or not a vertex is on the domain boundary. From these quantities one can calculate the area or volume of the computational domain and thus an average local or cell area/volume. From the boundary discretization one can determine a boundary parameterization or arclength. From element type one can determine ideal element shape (skew and aspect ratio). One can also compute the initial mesh Jacobian matrix (and determinant) at every sample point. One can calculate the initial mesh quality using various metrics. One can calculate mesh statistics (e.g., average, minimum, maximum, variance) on such quantities as length or area/volume. One can compute a local 'sizing' function [14]. The initial mesh can thus be used to calculate a wealth of intermediate data valuable in constructing target-matrices.

The third potential source of geometric raw data is a reference mesh, i.e., a mesh whose connectivity can be put into a one-to-one correspondence with the initial mesh. Reference meshes can be obtained from: (1) the initial or computational mesh at an earlier time in the simulation (applies to moving mesh or deforming domain problems) or (2) a mesh generated 'from scratch' on a similar domain with the same mesh topology (applies mainly to structured meshes). As with the initial mesh, the vertex coordinates of the reference mesh can be used to calculate the *reference* Jacobian-matrix and to calculate reference mesh statistics on length or area/volume.

3.1.3.2 Simulation Raw and Intermediate Data

Raw data that is created by the physical simulation consists primarily of the numerical solution to the governing equations. The numerical solution is the values of the dependent variables in the equations at some discrete set of points, perhaps along with interpolated values (as in FEM). Often the numerical solution includes velocities or fluxes, too. The numerical solution gives rise to scalar and vector fields as intermediate data.

Other fields can be obtained from the coefficients of the governing partial differential equations. As an example, in Darcy flow the permeability tensor appears as a PDE coefficient and determines principle flow directions. An anisotropic permeability tensor may call for an anisotropic optimal mesh. Material indicators are an example of a scalar field that can help determine target-matrix input parameters. Each type of physics simulation will have its own set of raw data which can perhaps help determine the input parameters in the target-matrix set.

Source	Raw Data	Intermediate Data
Physical Domain	Geometric Description: (CAD data, etc.) (Domain Topology) (Domain Symmetries) (Dimension)	Bounding Box & Centroid Domain 'Diameter' Aspect Ratio Orientation Sub-region Indicators
Initial Mesh	Mesh Type Mesh connectivity Element type(s) Vertex coordinates Boundary Flags	Domain area or volume Boundary parameterization Ideal element shape Mesh Jacobian Local quality metrics Mesh statistics Sizing functions
Reference Mesh	Vertex coordinates	Reference Jacobian Mesh Statistics

Table 3.3: Geometric Raw and Intermediate Data

The raw data present in the simulation can be further processed to create intermediate data. From symmetric-matrix data one can obtain real eigenvectors and eigenvalues which, in turn are useful for determining directions and aspect ratios. From vector fields one obtains directions and lengths. Raw solution data can be used to recover the solution Hessian matrix and to create spatially dependent a posteriori error estimates. The literature describes many different methods for creating some types of intermediate data (e.g., Hessian recovery methods [6], a posteriori error estimates [17], and interpolation error [7], [4]).

3.1.3.3 Correspondences Between Raw, Intermediate, and Parameter Data

The raw and intermediate data mentioned in Tables 3.3 and 3.4 only identifies what kinds of data can potentially be of use in Target Construction. In practice one needs to identify the specific raw data and the specific intermediate data which will be used in Target Construction. Specifying raw or intermediate data includes determining the form of the data and the locations of the data on the mesh or physical domain. Specific raw data may

Source	Raw Data	Intermediate Data
Scalar Fields	The Solution Material Indicator Function Streamlines	Gradient, Hessian Error Estimates
Vector Fields	Solution Gradient or Flux Velocity Electric and Magnetic Fields	Directions, Lengths
Matrix/Tensor Fields	Permeability Matrix Stress Tensor	Eigenvectors, Eigenvalues

Table 3.4: Simulation Raw and Intermediate Data

depend on the particular simulation code that is used.

Beyond identification of the specific raw and intermediate data that will be used, one needs to determine a *Correspondence Chain* between the raw and target data.

Definition.
A *Correspondence Chain* is a mapping which identifies, for a controlled target parameter, (a) the specific intermediate data which will be used to compute the controlled parameter data and (b) the specific raw data which will be used to compute the specific intermediate data.

One correspondence chain is needed for each target parameter that is to be controlled. As an example, one might define a correspondence chain between a particular simulation density field (the raw data), a particular error estimator (the intermediate data), and the volume parameter. In this example, the local volume in the optimal mesh is to be adapted to the simulation densify field.

Note that, if multiple correspondence chains are needed, one is free to choose the raw data sources for each chain independently of what the raw data is in a different chain. For example, the skew angles can be constructed from the raw data 'element type' while the area/volume parameter can be constructed from raw data consisting of the initial mesh vertex coordinates. At the same time the direction parameter group could be constructed from raw data corresponding to a simulation-produced vector field. As long as the raw data is accessible, one can mix and match the raw data sources when constructing the full set of correspondence chains. This flexibility is quite important in being able to find appropriate values for the target parameters.

3.1.3.4 The PIE Decision

When creating correspondence chains which identify specific data, it is helpful to keep in mind what can be referred to as the PIE-decision. For each of the parameter groups one has basically four options in defining how the group will be constructed. The four options are: (P) preserve, (I) improve, (E) equidistribute, or (A) abstain. In the *preserve* option, one seeks to retain the quality present in the initial mesh, with respect to the given input parameter. Thus, the specific raw data in the correspondence chain will be the initial mesh if one chooses the preserve option. In the *improve* option, the parameter group is constructed so-as to improve upon the quality existing in the initial mesh because the existing initial mesh quality is inadequate (with respect to the given parameter). In the *equidistribute* option, the parameter group is constructed by assigning a value which is constant over all sample points. In this way, quantities such as area, volume, or size can be equidistributed over the mesh. Strictly speaking, one would only do this in an attempt to improve the quality of the mesh with respect to the given parameter group, so equidistribution can be viewed as a special case of the improve option. Finally, the *abstain* option is used when there is no good raw data source for creating acceptable values for a given parameter group. This option was discussed in section 3.1.2.

As an example, consider the area/volume parameter group. In r-adaptivity, one often seeks to adapt (*improve*) mesh quality by varying the area/volume of mesh elements according to some scalar quantity such as an a posteriori error estimate. In mesh generation one often *equidistributes* cell area/volume in the absence of any specific knowledge about the particular physical simulation which the mesh will facilitate. In some mesh quality improvement problems, the area/volume of mesh elements has already been adapted to the solution; in that case, one wants the area/volume construction procedure to *preserve* the existing area/volume, i.e., not obliterate it during mesh optimization. Finally, it is often the case in mesh generation in the absence of simulation information to be indifferent or neutral to element area/volume, as in shape optimization. In this case, one abstains from constructing the area/volume parameter. One also abstains when there is no relevant raw data for the construction of the parameter group.

The 'preserve' option, if selected, usually calls for extracting values of geometric parameters from the reference mesh, so the latter becomes the specific raw data in the correspondence chain. The 'equidistribute' option most often calls for constants available or computable from mesh and simulation data. The skew parameter is very often determined by the ideal element data; this would fall under the equidistribute option. 'Improve' is the most difficult option to effect since it often requires model development. These options are illustrated in section 3.3.

In the PIE decision, one assigns to each correspondence chain (representing one target-parameter) either P, I, or E. Including the control/abstain option there are, in theory, $4^4 = 256$ possible PIE-A combinations that can be made, ranging from P-P-P-P (preserve everything) to A-A-A-A (abstain from all). In practice, the number of useful combinations is probably much smaller. Never-the-less, the large number of combinations provides a great deal of flexibility.

Finally, note that most often if a particular set of PIE-A decisions is made, it usually applies to every sample point of the mesh. For example, if one chooses to preserve the area parameter, one wants to preserve it at every mesh sample point. However, in some cases it may be desire-able to preserve area on one set of sample points and to equidistribute it on the remaining sample points. The flexibility to do this should be available in any general optimization code.

Having identified the mesh and simulation context, made the Parameter Control Decision, and established the necessary Correspondence Chains, one now has a complete Target Construction Strategy. Note that a correspondence chain only defines the mapping between the various levels of data; it does not tell one how the data is to be converted. The latter question is addressed in section 3.2.

3.2 Raw to Intermediate Data Conversion Algorithms

The purpose of the algorithms mentioned in this section is to convert raw mesh or simulation data into values of one or more mesh or solution functionals over the mesh. The correspondence chains determine exactly which raw data and which intermediate data will be the input and output of each algorithm needed in this phase. The intermediate data must ultimately be defined at the mesh sample points. If not, then the interpolation of the intermediate data to the sample points must be done in the target parameter calculation phase. The algorithms in this section provide examples as to how the raw data is converted.

3.2.1 Algorithms for Mesh Functionals

Examples of raw mesh data includes mesh connectivity, vertex or nodal coordinates, and element types. Examples of intermediate mesh data includes ideal element coordinates, the Jacobian matrix of a reference mesh, and mesh statistics. Algorithms for calculating the latter are described next.

3.2.1.1 Ideal Element Coordinates from Element Type

This sub-section covers a rather trivial but important example. The most commonly used mesh elements are the triangle, quadrilateral, tetrahedron, hexahedron, triangular prism, and the pyramid. An *ideal* element is a straight-sided isotropic element for which all angles within a given face are equal. The ideal forms of the first four element types are the equilateral triangle, the square, the equilateral tetrahedron, and the cube. Ideal elements define the desired *shape* of an element but not its size or orientation. The raw data in this example is the element type. If the mesh is a hybrid mesh then each element will likely have a flag associated with it that tells the element type; this flag is the raw data. If the mesh consists of only one element type, then one has (in principle) just one flag per mesh. Given the element type, one can define (as intermediate data), the vertex coordinates of the ideal element (see Tables 3.5 and 3.6).[5] Ideal elements need not have unit edge lengths. However, in terms of defining vertex coordinates for the ideal element, assuming unit edge length is convenient. Thus, the vertex coordinates in the table have been defined for ideal elements with unit edge lengths. Note that there is no conversion algorithm involved in this example, only a look-up table. One can, of course, provide their own definition of the coordinates of an ideal element. From the coordinates of

[5]These coordinates are not unique, but uniqueness is not needed for the present purpose.

an ideal element one can calculate the corresponding target skew and aspect ratio parameters, but this activity does not conceptually belong to the calculation of intermediate data (we discuss this latter activity in section 3.3.1.1).

Ideal Element	Vertex	Coordinates
Equilateral Triangle	0	(0,0)
	1	(1,0)
	2	$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$
Unit Square	0	(0,0)
	1	(1,0)
	2	(1,1)
	3	(0,1)

Table 3.5: Two-dimensional Ideal Elements

Ideal Element	Vertex	Coordinates
Equilateral Tetrahedron	0	(0,0,0)
	1	(1,0,0)
	2	$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right)$
	3	$\left(\frac{1}{2}, \frac{1}{2\sqrt{3}}, \sqrt{\frac{2}{3}}\right)$
Unit Cube	0	(0,0,0)
	1	(1,0,0)
	2	(1,1,0)
	3	(0,1,0)
	4	(0,0,1)
	5	(1,0,1)
	6	(1,1,1)
	7	(0,1,1)
Triangular Prism	0	(0,0,0)
	1	(1,0,0)
	2	$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right)$
	3	(0,0,1)
	4	(1,0,1)
	5	$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 1\right)$
Pyramid	0	(0,0,0)
	1	(1,0,0)
	2	(1,1,0)
	3	(0,1,0)
	4	$\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{2}}\right)$

Table 3.6: Three-dimensional Ideal Elements

3.2.1.2 Reference Mesh Jacobian from a Reference Mesh

A *reference mesh* is a mesh that exists and has the same topology (connectivity) as the mesh that is to be optimized. An important example of a reference mesh is the *initial mesh*; this is the mesh whose quality is to be improved via optimization. Another example of a reference mesh occurs in deforming domain problems: in that context, the reference mesh can be the active mesh a some earlier time in the deformation process. As a third example of a reference mesh, one may be able to use a moving mesh at an earlier time as the reference mesh. In many optimization problems the initial mesh has quality that is, in part, adequate or compatible with the optimization goal. When that is the case, we can use the local Jacobian matrices of the reference mesh to calculate all or part of the target-matrix. Suppose the element mapping has the form

$$\mathbf{x}(\xi_1, \xi_2, \xi_3) = \sum_{n=1}^{N} \mathbf{p}_n \, \phi_n(\xi_1, \xi_2, \xi_3)$$

with N the number of element nodes (or control points), $\mathbf{p}_n$ the coordinates of the n-th node in the reference mesh, and ϕ_n the n-th basis function. Then the element reference Jacobian matrix at the point (ξ_1, ξ_2, ξ_3) has elements

$$\begin{aligned} A^{ref}_{i,j}(\xi_1, \xi_2, \xi_3) &\equiv \frac{\partial x_i}{\partial \xi_j} \\ &= \sum_{n=1}^{N} p_{n,i} \frac{\partial \phi_n}{\partial \xi_j} \end{aligned}$$

The element reference Jacobian matrix at the sample points can be found by evaluation of the above at the logical sample points.

In this example of raw to intermediate data coversion, the reference mesh nodal coordinates are the raw data and the reference mesh Jacobian matrix (which can be considered to be a mesh functional) is the intermediate data. Once the reference mesh Jacobian is calculated one can use it to compute values of the target parameters at the sample points. This latter step converts intermediate data into parameter data and is discussed in section 3.3.1.

3.2.2 Algorithms for Simulation Functionals

Raw data related to the simulation is commonly available in the form of scalar and vector fields for the dependent variables in the PDE, along with PDE coefficients related to material properties. This data can be used to compute intermediate data (i.e.,solution/simulation functionals) such as gradients, flux, Hessians, error indicators, interpolation error, and a posteriori error estimates. Methods for doing so are better known to the simulation community than to the meshing community. Here we only provide some references to methods for converting simulation raw data into intermediate simulation data. For gradient recovery methods see [5], [9], [12], [17]. For an introduction to Hessian recovery methods see [6], [10], [13]. For interpolation error see [2], [3], and [7].

3.3 Target Parameter Model Development

The final phase of target construction is concerned with the conversion of intermediate data into target parameter data. Because the final phase often entails devising an ad-hoc model of target parameter behavior as a function of intermediate data, complete with arbitrary constants whose values must be determined, it is referred to as Target Parameter Model Development. As an example, one might wish to devise a model which relates local a posteriori error estimates to the target volume parameter υ. Clearly, in that case, one wants a function which decreases monotonically as the error increases. Beyond that, it is often unclear what type of function or model is most useful, so numerical experiments may be warranted. In such cases, Model Development is clearly an art, not a science.

Each correspondence chain identified in the first phase pairs some specific intermediate data with specific geometric target parameters, so multiple models may be necessary, one for each chain. On the other hand, there are cases in which models do not need to be developed because the relationship between the geometric parameter and the intermediate data is unambiguous. This is especially true when the intermediate data consists of mesh functionals.

3.3.1 Converting Mesh Functionals into Parameter Data

We give three examples of converting intermediate data in the form of mesh functionals into values of target parameters.

3.3.1.1 Skew and Aspect Ratio Parameters from Ideal Element Coordinates

Ideal elements can be used to find target parameter values for the skew and aspect ratio groups, but not the volume or orientation groups. The intermediate data in this case consists of the coordinates of the ideal element (see Tables 3.5 and 3.6). For the triangle element, let (u, v) be a point in the ideal triangle. For $0 \le \xi \le 1$ and $0 \le \eta \le 1 - \xi$, the points in the triangle are

$$\begin{aligned} u(\xi, \eta) &= \xi\, u_1 + \eta\, u_2 + (1 - \xi - \eta)\, u_0 \\ v(\xi, \eta) &= \xi\, v_1 + \eta\, v_2 + (1 - \xi - \eta)\, v_0 \end{aligned}$$

with (u_k, v_k) the coordinates at vertex $k = 0, 1, 2$. In that case the target-matrix corresponding to the ideal element is

$$\begin{aligned} W &= \begin{pmatrix} u_\xi & u_\eta \\ v_\xi & v_\eta \end{pmatrix} \\ &= \begin{pmatrix} u_1 - u_0 & u_2 - u_0 \\ v_1 - v_0 & v_2 - v_0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix} \end{aligned}$$

The skew angle is found via the dot product $\mathbf{w}_1 \cdot \mathbf{w}_2$, giving

$$\begin{aligned} \cos\phi &= (1,0) \cdot \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right) \\ &= \frac{1}{2} \end{aligned}$$

Then $\sin\phi = \frac{\sqrt{3}}{2}$ and $\phi = \frac{\pi}{3}$. Since all the lengths in the equilateral triangle are equal, the aspect ratio parameter is $\rho = 1$. Proceeding similarly for the unit square element, one finds $\cos\phi = 0$, $\sin\phi = 1$, and $\phi = \frac{\pi}{2}$. Note that the Jacobian matrix of these ideal elements is a constant over the element, thus so are the skew and aspect ratio parameters.

Table 3.7 gives the skew angles for the ideal elements in $\Re^3$, as computed from their vertex coordinates. For the triangular prism element it is assumed that the mapping is such that the $\zeta = 0$ face corresponds to points on the lower triangle and $\zeta = 1$ to points on the upper triangle. For the pyramid element it is assumed that the points on the base of the pyramid correspond to $\zeta = 0$ and the apex point is at $\zeta = 1$. For the 3D element types define $\cos\phi_{12} = \mathbf{u}_\xi \cdot \mathbf{u}_\eta$, $\cos\phi_{13} = \mathbf{u}_\xi \cdot \mathbf{u}_\zeta$, and $\cos\phi_{23} = \mathbf{u}_\eta \cdot \mathbf{u}_\zeta$. The Jacobian matrix of the ideal elements is constant over all the ideal elements except the pyramid, however the skew and aspect ratios are constant over all the element types including the pyramid. The aspect ratios of the ideal 3D elements are all 1.0.

In this ideal element example, the raw data is element type, the intermediate data are the vertex coordinates of the ideal element, while the final data is the face and dihedral angles ϕ_{12}, ϕ_{13}, and χ_1. The raw data is defined on the element, the intermediate data on the element vertices, and the final data

Element	ϕ_{12}	ϕ_{13}	ϕ_{23}	χ
Tetrahedron	$\frac{\pi}{3}$	$\frac{\pi}{3}$	$\frac{\pi}{3}$	$\cos^{-1}(\frac{1}{3})$
Hexahedron	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$
Tri-Prism	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{3}$	$\frac{\pi}{3}$
Pyramid	$\frac{\pi}{2}$	$\frac{\pi}{3}$	$\frac{\pi}{3}$	$\cos^{-1}(\frac{1}{\sqrt{3}})$

Table 3.7: Skew Angles for Ideal 3D Elements

at the sample points (because the angles are constant over the ideal element).

Of course, in this example, the computation of the ideal skew and aspect ratio parameters does not change from mesh to mesh or from problem to problem. One can simply use the results given in this section as the target values of the skew and aspect ratio parameters, going directly from the element type indicator to the target values.

3.3.1.2 Target-Parameter Values from a Reference Mesh Jacobian

Another example of using mesh functionals to find target parameter values is that of using a reference mesh (see section 3.2.1). The intermediate data in this example is the set of Jacobian matrices that can be computed at the sample points of the reference mesh. Values of the target parameters can be found by applying the matrix extraction functions defined in sections 2.1.3 and 2.2.3 to the Jacobian matrix. Or, one can apply the extraction functions defined in sections 2.1.2 and 2.2.2.

As an example, in many optimization problems the initial mesh has quality that is, in part, adequate or compatible with the optimization goal. When that is the case, one can use the local Jacobian matrices of the reference mesh to calculate all or part of the target-matrix. Let A_{ref} be the Jacobian matrix of the reference mesh at some given sample point (or just A when the context is clear). Then, if one wants to create values at every sample point for the area or volume parameter υ from the reference mesh, they simply let $\upsilon = \tilde{\upsilon}(A_{ref})$. Using these values as target parameter values implies that one finds the corresponding reference mesh values acceptable, or that one wants to preserve those values in the optimal mesh to be created.

Note that the target-parameters calculated from the reference mesh may not always satisfy the requirements of a valid target-matrix input parameter. For example, if the reference mesh has poor quality, one might encounter sample point locations for which $\ell_i = \tilde{\ell}_i(A_{ref}) = 0$ or for which $\upsilon = \tilde{\upsilon}(A_{ref}) \leq 0$. If the reference mesh contains such points, the reference mesh cannot be used, at least not at the sample points where these issues occur. Thus the reference mesh should have good quality, at least with respect to the target-parameters of interest.

In the reference mesh example of target-matrix input parameter construction, the reference mesh vertex coordinates are the raw data, the reference Jacobian matrix is the intermediate data, and the target-parameter data extracted from the matrix is the final parameter data.

3.3.1.3 Reference Mesh Statistics

As a third exmple of converting intermediate mesh data into target parameter values, consider again the Jacobian matrix computed on the reference (or initial) mesh. For any given target parameter p, one can form the set $\{p_k\}$ of values over the set of reference mesh sample points by using the extraction functions. Then, given values $\{p_k\}$, one can calculate parameter statistics over the sample points: the average,

$$\bar{p} = \frac{1}{K} \sum_{k=1}^{K} p_k$$

the minimum,

$$p_{min} = \min_k \{p_k\}$$

the maximum,

$$p_{max} = \max_k \{p_k\}$$

the standard deviation,

$$\sigma(p) = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (p_k - \bar{p})^2}$$

and other statistics.

Mesh statistics can be useful in Target Parameter Model Development. An important example occurs in the area or volume equi-distribution problem where the goal is to create equal area or volume elements throughout the mesh. In that case, the local area parameter υ should be constant, so one can set $\upsilon = \bar{\upsilon}_{ref}$ at all sample points. In this example, the vertex coordinates of the reference mesh are the raw data and the reference mesh Jacobian and the parameter values calculated therefrom, are the intermediate data. The final parameter data is based on a model which relates a mesh statistic (or, more generally, some function of a mesh statistic) to a set of target parameter values.

3.3.2 Converting Simulation Functionals into Parameter Data

This topic is illustrated with an example in which the model is relatively simple. Suppose one wants to adapt a 2D quadrilateral mesh to the principle axes of the permeability tensor K (a symmetric, positive definite matrix) in a subsurface Darcy flow simulation. The first column of the active Jacobian matrix is to be aligned with the principle flow direction to improve simulation accuracy. The mesh spacing in that direction should be small compared to the normal direction. The raw data is the permeability tensor K. The intermediate data consists of the unit eigenpairs of K; let those be referred to as $(\lambda, \mathbf{e})$ and $(\lambda^\perp, \mathbf{e}^\perp)$, with $\mathbf{e} = (\cos u, \sin u)$ and $\mathbf{e}^\perp = (-\sin u, \cos u)$ and u the angle between the x-axis and the direction of $\mathbf{e}$. Define the ratio $r \equiv \left(\frac{1}{\lambda}\right) \lambda^\perp$. If $r \leq 1$ align the first column of the active matrix with $\mathbf{e}$, so the orientation, skew, and aspect ratio target parameters can be chosen to be $\theta = u$, $\phi = \frac{\pi}{2}$, and $\rho = \frac{1}{r}$. On the other hand, if $r > 1$, align the first column of the active matrix with $\mathbf{e}^\perp$, so the orientation, skew, and aspect ratio target parameters can be chosen to be $\theta = u + \frac{\pi}{2}$, $\phi = \frac{\pi}{2}$, and $\rho = r$.

Three of the four target parameters were defined in the Darcy flow example above. If the permeability tensor is not a constant, then the orientation and aspect ratio parameters will vary from one sample point to the next. The remaining target parameter, corresponding to the volume υ, can also be defined using the tensor. It is reasonable to expect that the local mesh volume would be a function of the determinant of K, i.e., $\upsilon = \upsilon(\lambda \lambda^\perp)$. Here is where model development is needed. It seems clear that the volume should be a decreasing function of the determinant because a relatively large determinant means greater flows and thus requires smaller element areas.

Many functions, of course, satisfy this condition, for example

$$\upsilon = \upsilon_o \exp^{-a\,\lambda\,\lambda^{\perp}}$$

and

$$\upsilon = \upsilon_o \left\{ 1 - \frac{2}{\pi} \arctan\left(a\,\lambda\,\lambda^{\perp} \right) \right\}$$

with constants υ_o, a to be determined. Both functions obey $\upsilon = \upsilon_o$ when the determinant is zero and monotonically decrease to zero as the determinant approaches infinity. Two issues arise in this kind of model development (1) what is a good functional form and (2) what values should be given to the constants? From experience, the answer to the first question seems to be that the particular functional form is often not critical, as long as it is monotonically decreasing. However, in practice, trying several functional forms is advisable. Answering the second question generally requires numerical experimentation using the simulation of interest. In this example a PDE coefficient matrix and its eigenvalues/vectors were used as the intermediate data. Other sources of intermediate data that might be useful in a simulation were given in Table 3.4.

In this Darcy flow example, there are four correspondence chains, one for each of the four geometric parameters associated with meshes in $\Re^2$. All four chains began with the Permeability Tensor K over the domain as the raw data. The first chain connects K to the intermediate data $det(K)$ and then to the volume parameter υ. Since in this chain there was no use of a reference mesh nor of any constant data, the chain is an 'improve' chain with respect to the PIE decision. The second chain connects K to the intermediate data consisting of the eigenvectors $(\mathbf{e}, \mathbf{e}^{\perp})$ of K. The second chain then connects the intermediate data to values of the orientation parameter θ; this chain is also an 'improve' chain. The third chain is built upon the fact that the eigenvectors of K are orthogonal, suggesting that the skew angle should be that of the ideal quadrilateral element, i.e.,$\frac{\pi}{2}$. In this case one can say that the raw data is K, the intermediate data is the ideal angle in a quadrilateral element, and the final parameter data is the skew angle. The chain uses the 'equidistribute' option since the target skew angle is constant over the mesh. The last chain is build upon the idea that the ratio of the eigenvalues of K suggest an appropriate value for the aspect ratio parameter. In this case one can say that the raw data is K, the intermediate data consists of the eigenvalues of K and the final parameter data is the aspect ratio parameter. The chain uses the 'improve' option since the target aspect ratio is non-constant and does not come from a reference mesh. In summary, the four chains correspond to a PIE-A decision of the form I-I-E-I.

3.3.3 Summary of the Model Development Phase

In the model development phase one defines the way in which the intermediate data in each correspondence chain will be converted into values of a particular target parameter. There are many ways in which this can be done and one is limited only by one's ingenuity. The values should all be defined over the set of mesh sample points. Once the parameter values have all been determined, one can further compute various target matrices if needed by the local quality metric.

3.4 Summary of the Target-Parameter Construction Process

The three phases in Target Construction are (1) Strategy Development, (2) Intermediate Data Algorithms, and (3) Parameter Data Model Development. Although each stage has been carefully described in some detail, target construction is not envisioned as a formal process. Rather, the description is intended only to guide those seeking to define values of the required target parameters that are appropriate to a particular mesh quality improvement goal. Although other mesh optimization methods have used target parameters of various sorts, this is the first time that methods for assigning values to these parameters has been systematically described. There are no guarantees that the Target Construction process will result in an adequate optimal mesh or that the mesh quality improvement goal will be reached. However, at least there is now a rational approach to the problem of assigning values of target parameters that can be systematically employed in the future. As noted earlier, in the long term this could result in a library of target-construction algorithms available to the general community.

3.5 Target Construction Examples for Planar Meshes

In this section examples are given of target-parameter construction for two-dimensional meshes in order to illustrate the wide variety of mesh improvement problems which can be addressed in TMOP. In each example, the process by which one determines which target-parameters are to be controlled is described, along with their values, over the full set of mesh sample points.

Mesh quality improvement begins with an *initial* mesh and ends with an *optimized* mesh. To improve the initial mesh one must construct the target-parameter values. As noted in Section 3.1, the values are ultimately determined by raw and intermediate data. If the data used to construct the parameter values is strictly geometric in nature (i.e., does not originate from a physical simulation), then the data will be defined at logical locations associated with the mesh. As the mesh moves within the domain during the optimization procedure, the data moves along with it. On the other hand, if some of the data used to construct the parameter values is obtained from the simulation, then data will most likely be defined at physical locations associated with the domain. As the mesh moves within the domain during the optimization procedure the simulation data does not move with it. In this case, the physical parameter data must be interpolated (or advected) to logical locations in the mesh. However, the interpolation or advection of data occurs during the optimization step and therefore has nothing to do with the target-parameter construction process, which occurs prior to optimization. So, target-parameter construction does not depend on whether or not one uses geometric or physical data in the target (one can mix and match the different types as needed). That said, the final values of the target-parameters to be constructed must be located at logical locations within the mesh called sample points, because that is where the active and target-matrices reside. In the examples below it is assumed that interpolation of the data to sample points has taken place at some stage of the raw-intermediate-final data processing, but this topic is not discussed further in the examples.

Any mesh quality improvement problem in which simulation data is used in the construction of the target-parameters is called a solution-adaptive problem and the target is called a *solution-adaptive target*. If no simulation data is used in the construction of the target parameters, then the target is called a *geometric target*. In many instances one uses a mixture of simulation and geometric data; these cases are also considered solution-adaptive targets. Section 3.5.1 concerns examples of geometric target construction while Section 3.5.2 gives examples of solution-adaptive target construction. The distinction between these two types of target is only important in the optimization procedure. In terms of target construction, the distinction is unimportant.

In the examples that follow reference is made to certain metric *types*. One outcome of the PIE-A decision is to define the type of metric which should be used in the optimization procedure. In a follow-on document it will be shown that there are 15 metric types that can potentially arise. Of these only 4-6 metric types are actually viable in mesh optimization due to the existence of multiple optimal meshes. The PIE-A decision is thus constrained with those limitations in mind. In the examples of 2D Target Construction below some of the viable metric types are referred to even though they are not defined in this document.

3.5.1 Target-Parameter Construction of Geometric Targets for Planar Meshes

3.5.1.1 Equi-distribute Local Shape

The goal in this example is to create a set of target-matrices which can be used to create an optimized mesh in which the average element shape will be closer to the ideal element shape than the average element shape in the initial mesh. Ideal element shape is defined a priori. The raw data in this problem is *element type*. For codes that use multiple element types, there is usually some indicator function to inform the code of the type of each specific element. For codes that use only one element type, this is of course trivial. The skew angles which define the ideal element type are considered to be intermediate data. To determine the target, this raw and intermediate data is converted into the final values of the target-parameters within the target matrices. To control shape, we need to control the skew and aspect ratio parameter groups and abstain from the size and orientation parameter groups. Thus, the PIE-A decision for volume-orientation-skew-aspect is abstain, abstain, equidistribute, equidistribute (AAEE). One could also view the latter two decisions as falling into the 'improve' category, but since only (constant) a priori data is used, it seems more appropriate to consider this as a decision to equidistribute.

Since only skew and aspect ratio (i.e., shape) are to be controlled, metric type *shape* must be used. The full target-matrix W is the product of the size parameter times the orientation matrix times the shape matrix. If we use a shape metric that explicitly depends on W, we can use default values for the size parameter and for the orientation factor. If the metric is invariant to size and orientation, then the actual default values selected will not affect the optimal mesh. Only the actual values in the shape matrix will matter. As will be seen later, one can also use shape metrics which depend only on the shape matrix S. In that case, one need only construct the incomplete target S_W, not W. For any shape metric, it is the construction of S_W that is critical.

If every element in the mesh has the same type, and one desires that they all have the ideal shape, then the shape-matrix must be the same at every mesh sample point. Thus, if the shape matrix at sample point k is S_k, then S_k is a constant matrix, independent of k. If the mesh has more than one element type (a hybrid mesh), then there will be one shape matrix for each of the element types. The shape-matrix will then be constant over all sample points belonging the same element type. Mesh topology is immaterial to the construction of the target when the ideal mesh element is isotropic (as in the case of triangles, quadrilaterals, tetrahedral, and hexahedra).

For completeness, the shape-matrix (i.e., the incomplete target-matrix) is given here for each ideal element type. For triangular elements, $\phi = \frac{\pi}{3}$ and $\rho = 1$. Then

$$
\begin{aligned}
S_k^{tri} &= \begin{pmatrix} 1 & \cos\phi \\ 0 & \sin\phi \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{\rho}} & 0 \\ 0 & \sqrt{\rho} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix}
\end{aligned}
$$

for every k.

For quadrilateral elements, $\phi = \frac{\pi}{2}$ and $\rho = 1$. Then

$$\begin{aligned} S_k^{quad} &= \begin{pmatrix} 1 & \cos\phi \\ 0 & \sin\phi \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{\rho}} & 0 \\ 0 & \sqrt{\rho} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

for every k.

3.5.1.2 Equi-distribute Local Shape and Size

The goal in this example is to define a set of target-matrices which will encourage the optimized mesh to have elements whose shape and size is nearly constant. The previous example showed how targets can be constructed to encourage ideally-shaped (i.e., constant shape) elements. To additionally incorporate size into the target-matrices, note first that the parameters for size (ζ) and volume (υ) are related. In two-dimensions we have $\upsilon = \zeta \sin\phi$ and in three-dimensions, $\upsilon = \zeta \sin\phi_{12} \sin\phi_{13} \sin\chi$. So, by controlling both skew and size, one is also controlling volume. The PIE-A decision for this example is equidistribute, abstain, equidistribute, equidistribute (for volume/size, orientation, skew, and aspect ratio). The raw data in this example consists of (a) element type and (b) the initial mesh and mapping. The method by which intermediate shape data is created from element type. Intermediate area or volume data is created by approximating the volume of the physical domain by summing up the volumes of the mesh elements. The element volumes, in turn, are found by summing over the local volumes at each sample point within the element. The initial mesh can be used for this purpose. Since υ is a local quantity defined at each sample point, the total domain area must be divided by the number of elements to get a value of υ. Finally, to find ζ, divide the volume by the sine of the skew angles. Since it is intended to control the size and shape parameters, the corresponding incomplete target-matrix is $U(\zeta, \phi, \rho) = \sqrt{\zeta}\, S(\phi, \rho)$ in two dimensions and $U(\zeta, \phi_{12}, \phi_{13}, \chi, \rho_1, \rho_2, \rho_3) = \zeta^{\frac{1}{3}}\, S$ in three dimensions, where S is the ideal shape matrix for the given element type. The shape+size factor U is thus a constant matrix, being the same at every sample point. In order to use these targets, a shape+size metric must be used. If the shape+size metric depends explicitly on U, nothing more need be done. If the shape+size metric depends explicitly on W, then one must choose a default value (such as the identity) for the orientation factor R. Then $W = RU$ and the metric can be evaluated.

3.5.1.3 Equi-distribute Local Skew and Size

In the previous example, the goal was to control size, skew, and aspect ratio (i.e, volume and shape). In some cases, one can obtain a better mesh by ignoring aspect ratio and controlling only size and skew. Values for the size and skew parameters in this example are determined exactly as in the previous example. The result is a constant matrix. The PIE-A decision is E-A-E-A. The incomplete target-matrix is $\sqrt{\zeta}\, Q$ in two-dimensions and $\zeta^{\frac{1}{3}}\, Q$ in three dimensions. This set of targets requires one to use a 'volume+skew' metric.

3.5.1.4 Preserving Good Meshes on Deforming Domains

In this example the physical domain changes with time during a simulation due to various forces exerted on the domain. It is assumed that the boundary of the domain is known at the beginning of each time step. It is also assumed that at the previous time-step the mesh has acceptable quality and includes non-constant features such as anisotropy, biasing, clustering of vertices, and perhaps other features. The goal is to create target-matrices such that the optimized mesh (to be used at the next time-step) resembles the mesh at the previous time-step and thus preserves the non-constant features. There are several variations on this theme, depending on the physical problem and the particular features one wishes to preserve. In TMOP this is again described by the PIE-A decision.

In the simplest situation, one wishes to preserve all of the features of the mesh at the previous time-step (i.e., the initial mesh). That is, the goal is to preserve shape, size, and orientation of the previous mesh, corresponding to the PIE-A decision P-P-P-P. In that case, the full target-matrix W is needed. The preservation option requires the use of the initial mesh as the reference mesh, with Jacobian A_{ini}. Values of the reference Jacobian are found from the mesh at the previous time-step. Thus, one constructs the set of target-matrices at each sample point by setting $W = A_{ini}$. Because all of

W is used in this situation, one must use metric type shape+size+orientation (VOS). The raw data in this example can be considered to be the mesh at the previous time-step, the intermediate data is $\{A_{ini}\}$, and the final data is $W = A_{ini}$.

A variation on the previous example would be to change the PIE-A decision to P-P-E-P, in which the ideal skew angles replace the angles from the reference mesh.

Another situation which occurs in practice is the need to preserve everything except volume. This occurs, for example, when the size of the domain is rapidly increasing or decreasing with time. In this case, one can either abstain on volume (giving A-P-P-P) or one can re-calculate the local volume at each sample point as time passes. In the former, one would calculate the matrix $R\,S$ from the reference mesh and use that as the incomplete target-matrix, along with a shape+orientation (OS) metric. Volume would not be controlled and should change automatically as needed, with this selection. In the latter possibility, the PIE-A decision is I-P-P-P, in which data must be found for the size factor ζ (or, equivalently the volume factor υ). One way to do this would be to compute the total volume of the domain (V), both at the previous and at the current time steps $(n, n+1)$. This requires the use of the mesh at time-step n and the mesh at time-step $n+1$. The latter mesh conforms to the updated domain boundary and is considered the initial mesh in the optimization procedure for time step $n+1$. Then, at each sample point, let

$$\upsilon^{n+1} = \frac{V^{n+1}}{V^n}\,\upsilon^n$$

Since, in two dimensions, $\upsilon = \zeta\,\sin\phi$, one has $\upsilon^n = \zeta^n \sin\phi^n$ and $\upsilon^{n+1} = \zeta^{n+1}\sin\phi^n$. In the latter one can use ϕ^n in place of ϕ^{n+1} because the intention is to preserve the skew angle. Thus

$$\zeta^{n+1} = \frac{V^{n+1}}{V^n}\,\zeta^n$$

In the A-P-P-P case, one uses an orientation+shape metric, with incomplete target-matrix $R(W)\,S(W) = R(A)\,S(A)$. In the I-P-P-P case, one uses a VOSA (shape+size+orientation) metric, with $W^n = \zeta^{n+1}\,R(A_{ref})\,S(A_{ref})$.

3.5.1.5 Removing Small Edges

Suppose that in two dimensions, at some sample point, the edge which defines ℓ_2 in the initial mesh is unacceptably small. Let $\ell_{min} > 0$ be the smallest acceptable edge length in the mesh and assume $\ell_2 < \ell_{min}$. The goal is to construct the target-parameters such that in the optimized mesh the small edge length ℓ_2 at the same sample point has been increased. Let the new edge length be $\ell_2 + \Delta\ell_2$. At the same time, let's preserve the length ℓ_1 at the same sample point, i.e., there should be no change in ℓ_1 as a result of the mesh optimization. In fact, let $f \geq 1$ be a user-input and let us suppose that we desire $\ell_2 + \Delta\ell_2 = f\,\ell_{min}$ after optimization.

Recall that in two-dimensions, $\zeta = \ell_1\,\ell_2$ and $\rho = \frac{\ell_2}{\ell_1}$. Thus,

$$\begin{aligned} \rho\,\zeta &= \ell_2^2 \\ \frac{\zeta}{\rho} &= \ell_1^2 \end{aligned}$$

Then, since a goal is to preserve ℓ_1,

$$\begin{aligned} \Delta\left(\frac{\zeta}{\rho}\right) &= 0 \\ \frac{\rho\,\Delta\zeta - \zeta\,\Delta\rho}{\rho^2} &= 0 \end{aligned}$$

and thus $\rho\,\Delta\zeta = \zeta\,\Delta\rho$ or

$$\frac{\Delta\zeta}{\Delta\rho} = \frac{\zeta}{\rho}$$

This means that, for some $\beta \neq 0$, one needs $\Delta\zeta = \beta\,\zeta$ and $\Delta\rho = \beta\,\rho$.

Furthermore,

$$\begin{aligned} \Delta\,(\rho\,\zeta) &= \Delta\left(\ell_2^2\right) \\ \rho\,\Delta\zeta + \zeta\,\Delta\rho &= 2\,\ell_2\,\Delta\ell_2 \\ \rho\,(\beta\,\zeta) + \zeta\,(\beta\,\rho) &= 2\,\ell_2\,\Delta\ell_2 \\ \beta\,\rho\,\zeta &= \ell_2\,\Delta\ell_2 \\ \beta &= \frac{\Delta\ell_2}{\ell_2} \end{aligned}$$

So, if $\beta > 0$, then $\Delta\ell_2 > 0$ and the length of the small edge will increase under optimization. Finally, in terms of f and ℓ_{min},

$$\beta = \frac{f\ell_{min}}{\ell_2} - 1$$

with $\beta > 0$.

Therefore, in this target construction method, one first writes the target-matrix in the parametric form in equation (2.2). For the given sample point at which the small edge occurred, set $\theta = \theta_{init}$ and $\phi = \phi_{init}$ from the initial mesh. Then replace ζ in the target with $\zeta + \Delta\zeta = (1+\beta)\,\zeta_{init}$ and replace ρ in the target with $\rho + \Delta\rho = (1+\beta)\,\rho_{init}$ (where β is given in the last formula above, with ℓ_2 the value of ℓ_2 on the initial mesh). A metric of type VOSA preserves orientation and skew, while improving size and aspect ratio with the revised target-matrix. The PIE-A decision is thus I-P-P-I.

In practice, one can find every sample point in the initial mesh at which $\ell_2 < \ell_{min}$ and construct the target accordingly.

Of course, it may happen that at some sample points $\ell_1 < \ell_{min}$. In that case, let's preserve ℓ_2 and increase ℓ_1. That is, set

$$\Delta\,(\rho\,\zeta) \quad = \quad 0$$

and

$$\Delta\left(\frac{\zeta}{\rho}\right) \quad = \quad \Delta\left(\ell_1^2\right)$$

Then,

$$\frac{\Delta\zeta}{\Delta\rho} = -\frac{\zeta}{\rho}$$

and

$$\rho\,\Delta\zeta - \zeta\,\Delta\rho \quad = \quad 2\rho^2\,\ell_1\Delta\ell_1$$

From the first relation, for $\beta \neq 0$, $\Delta\zeta = \beta\,\zeta$ and $\Delta\rho = -\beta\,\rho$ is a solution. (The alternative solution, $\Delta\zeta = -\beta\,\zeta$ and $\Delta\rho = \beta\,\rho$ will be discussed momentarily). From the second relation,

$$\rho\,\Delta\zeta - \zeta\,\Delta\rho \quad = \quad 2\,\rho^2\,\ell_1\Delta\ell_1$$

$$
\begin{aligned}
\rho\,(\beta\,\zeta) - \zeta\,(-\beta\,\rho) &= 2\,\rho^2 \ell_1\,(f\,\ell_{min} - \ell_1) \\
2\,\beta\,\rho\,\zeta &= 2\,\rho^2 \ell_1\,(f\,\ell_{min} - \ell_1) \\
\beta\,\frac{\zeta}{\rho} &= \ell_1\,(f\,\ell_{min} - \ell_1) \\
\beta\,\ell_1^2 &= \ell_1\,(f\,\ell_{min} - \ell_1) \\
\beta &= \frac{f\,\ell_{min} - \ell_1}{\ell_1}
\end{aligned}
$$

Here, $\beta > 0$, but while ζ will be increased, ρ will be decreased. In that case, ℓ_1 is increased but ℓ_2 is preserved. The alternative solution gives

$$\beta = -\frac{f\,\ell_{min} - \ell_1}{\ell_1}$$

Here, $\beta < 0$. Then, from the alternative solution one sees that $\Delta\zeta > 0$ and $\Delta\rho < 0$. Therefore, in the alternative solution, ζ will be increased and ρ will be decreased. Thus, the alternative solution gives the same result as the original solution, i.e. ℓ_1 is increased but ℓ_2 is preserved.

Finally, this scheme does not guarantee that in the optimal mesh all edge lengths will be greater than ℓ_{min}. It only encourages it. If the scheme fails to give the needed improvement one could consider (a) decreasing the value of f (while keeping in greater than 1) or (b) choosing I-A-P-I in the PIE-A decision to provide more flexibility to the mesh optimization. That would require a size+shape (VS) metric.

3.5.2 Target-Parameter Construction of Solution-Adaptive Targets for Planar Meshes

In these examples, one or more of the target-parameter values is defined in physical space, while others are defined in logical space. Thus, some parameters will be interpolated (or advected) while other will not.

3.5.2.1 Adapting Local Volume to a Set of Error Estimates

Suppose that, as raw data, one has values of the dependent variable in a simulation at the sample points of the initial mesh (the one just prior to optimization). Suppose, in addition, that the raw data is processed to give intermediate data consisting of values $\{e_j\}$ of some scalar error estimator at the sample points. From the intermediate data, compute as final data, the values of the local area parameters $\{\upsilon_j\}$ at the sample points. To do this, a function $\upsilon = \upsilon(e)$ is used to convert the local errors into local areas. With such a function, one can find $\upsilon_j = \upsilon(e_j)$. Finally, convert the local volumes to local sizes $\{\zeta_j\}$ by the usual formulas in 2D or 3D, if needed.

Assume, for the moment, that one has such a function. Next, the PIE decision must be made. It has already been decided to adapt the local volume to the error, so the decision must be of the form I-X-Y-Z, where X, Y, and Z are to be determined. Most likely, one will want to abstain on the local orientation, so choose X=A, giving I-A-Y-Z. It is also reasonable to assume that in most cases, the skew decision will be Y = E, so that angles in the adapted mesh will be close to ideal.[6] So far, then, one has I-A-E-Z. For the aspect ratio parameter it might be best to choose Z=P (i.e., to preserve existing aspect ratios), especially if the initial mesh contains high-aspect ratio elements. So, the final PIE-A decision is I-A-E-P. This choice requires a shape+size metric.

With the I-A-E-P choice, one need only interpolate the values of the local volume during the optimization procedure. The other parameters are computed in the usual fashion on the intial mesh and do not need to be updated during optimization.

The remaining question is how to create an adequate function $\upsilon(e)$. To begin, define a non-dimensional error parameter, given by

$$E = \frac{e - e_{min}}{e_{max} - e_{min}}$$

with $e_{min} = \min_j\{e_j\}$ and $e_{max} = \max_j\{e_j\}$. Then $0 \leq E \leq 1$ and

$$E_j = \frac{e_j - e_{min}}{e_{max} - e_{min}}$$

Let $\upsilon(e) = \tilde{\upsilon}(E)$. There are many functions $\tilde{\upsilon}(E)$ which may suffice for adapting the mesh to the error, so let us first mention some basic requirements on $\tilde{\upsilon}$. The first requirement is that $\tilde{\upsilon}(E) > 0$ for $0 \leq E \leq 1$. This guarantees that the targets have positive local volume. The second requirement is that $\tilde{\upsilon}$ should be a strictly decreasing function of E because larger

[6] In some rare instances, one might wish to choose Y = I, if one has some a priori knowledge about the skew angles.

errors require smaller local volumes in the adapted mesh. A third requirement is that

$$\sum_{j=1}^{J} \tilde{\upsilon}_j = \sum_{j=1}^{J} (\upsilon_A)_j$$

where υ_A is the local volume in the initial mesh.

To proceed further, define the change in the volume at point j to be

$$(\Delta\tilde{\upsilon})_j = \tilde{\upsilon}_j - (\upsilon_A)_j$$

Substituting this relation into the third requirement on $\tilde{\upsilon}$, the requirement on $\Delta\tilde{\upsilon}$ is

$$\sum_{j=1}^{J} (\Delta\tilde{\upsilon})_j = 0$$

From the second requirement on $\tilde{\upsilon}$, one sees that $\Delta\tilde{\upsilon}$ must be a decreasing function of E because for E near 1, the change in volume must be negative, while for E near 0, the change in volume must be positive. In addition, let's require that $\Delta\tilde{\upsilon}(E) = 0$ at $E = \bar{E}$, where the latter is the average of the $\{E_j\}$. Finally, from the first requirement on $\tilde{\upsilon}$,

$$(\Delta\tilde{\upsilon})_j > -(\upsilon_A)_j$$

for every j. Because the function $\Delta\tilde{\upsilon}$ is decreasing, its minimum value occurs at $E = 1$. Thus $(\Delta\tilde{\upsilon})_j \geq \Delta\tilde{\upsilon}(1)$. Therefore, it is sufficient to require

$$\begin{aligned} \Delta\tilde{\upsilon}\,(1) &> \max_j\{-(\upsilon_A)_j\} \\ &= -\min_j\{(\upsilon_A)_j\} \end{aligned}$$

The simplest function to satisfy these requirements would be a linear function in E. Therefore, assume that $\Delta\tilde{\upsilon}$ has the form

$$\Delta\tilde{\upsilon} = a\,E + b$$

Since it is required that $\Delta\tilde{\upsilon}(\bar{E}) = 0$, $b = -a\,\bar{E}$ and thus

$$\Delta\tilde{\upsilon} = a\,\left(E - \bar{E}\right)$$

Since this is supposed to be a decreasing function one needs $a < 0$. Further

$$\Delta\tilde{\upsilon}(1) = a\,\left(1 - \bar{E}\right)$$

Assuming $(\upsilon_A)_j > 0$ for all j, the first requirement on $\Delta\tilde{\upsilon}$

$$a\ (1 - \bar{E}) > -\min_j\{(\upsilon_A)_j\}$$

is satisfied when

$$a > -\frac{\min_j\{(\upsilon_A)_j\}}{1 - \bar{E}}$$

Therefore, the range of a must be constrained to

$$-\frac{\min_j\{(\upsilon_A)_j\}}{1 - \bar{E}} < a < 0$$

Finally, note that with this model,

$$\begin{aligned}
\sum_{j=1}^{J} (\Delta\tilde{\upsilon})_j &= \sum_{j=1}^{J} a\ (E_j - \bar{E}) \\
&= a \sum_{j=1}^{J} (E_j - \bar{E}) \\
&= a \left(\sum_{j=1}^{J} E_j - \sum_{j=1}^{J} \bar{E} \right) \\
&= a\ (J\,\bar{E} - J\,\bar{E}) \\
&= 0
\end{aligned}$$

Thus the constraint on the sum of the delta's is automatically satisfied for any a.

The final value of the constructed volume parameter is

$$\tilde{\upsilon}_j = (\upsilon_A)_j + a\ (E_j - \bar{E})$$

with 'a' a user input parameter in the specified range.

Of course, many other functional forms for $\Delta\tilde{\upsilon}$ satisfying the requirements may be more suitable for adapting to the local error. For example, if $\Delta E = E - \bar{E}$, functions of the form $a\,\Delta E + b\,(\Delta E)^3$ or $\tan(\Delta E)$ may work better because they are less sensitive to small ΔE but change rapidly with larger ΔE. To determine that, however, requires an actual simulation with error values available.

3.5.2.2 Adapting to a Material Indicator Function

In the ICF problem described in earlier work, the domain was a quarter circle which was meshed by a multi-block structured quadrilateral mesh consisting of three blocks. In the outer part of the mesh, one had azimuthal symmetry. The goal in the problem was to create values for the target parameters (the target-matrix in particular) so that the mesh would be adapted to one or more material layers at distances r_1 and r_2 from the center of the circular domain.The suggested target-matrix for one layer was

$$W = \begin{pmatrix} 1 & 0 \\ 0 & \rho \end{pmatrix}$$

with

$$\rho = \begin{cases} f & \text{if } a < \sqrt{x^2 + y^2} < b \\ 1 & \text{else} \end{cases}$$

with user-parameters $0 < f \leq 1$ and $\sqrt{2} < a < b < 3$. Notice that the target-matrix here is a function of position in physical space and $\rho = \rho(r)$. With this target-matrix, the optimized mesh is adapted to the material layers by creating high aspect-ratio cells in the ring at distance $r = \sqrt{x^2 + y^2}$ from the circle center. The shape metric μ_2 was used in optimizing the objective function.

In terms of the construction method presented in this document, the Parameter Control Decision is to abstain or not abstain on each of the four geometric parameters in 2D. Since a shape metric was used, it is clear that the decision must have been A - A - C - C. In PIE, the decision was A - A - E - I. The skew angle in the target is the ideal skew angle for a quadrilateral element ($\frac{\pi}{2}$) and the aspect ratio parameter defined as above. Thus, the correct shape matrix in this problem is

$$S = \begin{pmatrix} 1 & \cos\phi \\ 0 & \sin\phi \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{\rho}} & 0 \\ 0 & \sqrt{\rho} \end{pmatrix}$$

With the ideal skew angle, the skew matrix Q becomes the identity matrix. Thus $S = D$. The aspect ratio matrix D can be written

$$D = \frac{1}{\sqrt{\rho}} \begin{pmatrix} 1 & 0 \\ 0 & \rho \end{pmatrix}$$

Since shape metrics are invariant to scaling, using the matrix W above is equivalent to using S above. As seen previously, using S in place of T in μ_2 gives the same result.

Now consider the more general two-dimensional, two-material problem in which one is given a discrete material indicator function ι_j with $0 \le \iota_j \le 1$ at physical points (x_j, y_j) in the initial mesh. The goal is to adapt the local mesh volume to the two materials it contains. Clearly one needs to control volume and skew. Without additional information, it makes sense to abstain on the orientation parameter. In the ICF problem just described aspect ratio was controlled instead of volume. This was because thin mesh layers at distance r were desired. In the more general problem here, no such assumption is made, and thus it is better to control volume instead of aspect ratio. It seems that, in the absence of additional information, there are three potential approaches to aspect ratio: (a) set aspect ratio to 1, (b) abstain on aspect ratio, or (c) preserve aspect ratio. The PIE decision is I - A - E - E, I - A - E - A, or I - A - E - P, respectively. The resulting metric types are VS (volume+shape), VQ (volume+skew), and VS, respectively. The choice between the three approaches depends on whether one believes that the simulation will produce high-aspect ratio elements or not; if it might, then option (a) should be avoided. Additionally, if one believes that the aspect ratio in the optimized mesh should not be too different from the aspect ratios in the initial mesh, then option (c) is a viable approach. If one expects the aspect ratios to change a lot between the initial and optimal meshes, then option (b) is the best choice.

Finding the values for the skew and aspect ratios is easy once one of the three options above is selected. Next, concentrate on finding a model for the volume parameter, given the material indicator. Suppose that $\iota = 1$ means that locally the only material present is material 1 and $\iota = 0$ means that locally only material present is material 2. Further, assume that one wants the local volume of material 1 to decrease monotonically with ι (and therefore the local volume of material 2 (which is $1 - \iota$) should increase monotonically with ι). Suppose that $\upsilon_{ini}^{(1)}$ is the local volume of material 1 in the initial mesh and $\upsilon_{opt}^{(1)}$ is the local volume of material 1 in the optimal mesh (at the same physical location). Assume that $\upsilon_{init}^{(1)} > 0$ at all locations. Write $\upsilon_{opt}^{(1)} = \upsilon_{ini}^{(1)} + \Delta\upsilon$. It seems reasonable to require that $\Delta\upsilon < 0$ if and only if $\iota > \frac{1}{2}$. The simplest model for $\Delta\upsilon$ is

$$\Delta\upsilon = m\left(\frac{1}{2} - \iota\right)$$

with the requirement $0 < m < 2\,\upsilon_{ini}^{(1)}$ to ensure that $\upsilon_{opt}^{(1)} > 0$ at all locations. The value of m would be a user input parameter.

With this model the target volume parameter is fully defined. The maximum range of $\upsilon_{opt}^{(1)}$ is $0 < \upsilon_{ini}^{(1)} - \frac{m}{2} < \upsilon_{opt}^{(1)} < \upsilon_{ini}^{(1)} + \frac{m}{2} < 2\,\upsilon_{ini}^{(1)}$. Other models for $\Delta\upsilon$ can, of course, be developed.

3.5.2.3 Alignment of Mesh Lines to a Smooth Vector Field

The term 'alignment' in meshing has been defined in the past by saying that two vectors in $\Re^3$ are *aligned* if their cross product is zero [1]. The equivalent in two dimensions is to say that the area enclosed by the two vectors is zero. With this definition, if $\mathbf{u}$ and $\mathbf{v}$ are aligned, then $\mathbf{u} = \lambda\mathbf{v}$, with λ a real number. In TMOP, however, an alternative definition of alignment is used based on target terminology. Recall that θ is the angle between the x-axis and the first column vector of any non-degenerate matrix $X_{2\times 2}$. In TMOP, two vectors in $\Re^2$ are aligned if their orientation angle with respect to the x-axis is the same. In $\Re^3$ two vectors are aligned if they have the same spherical coordinate angles. This definition is used, not because it is better, but because it is more consistent with the target construction approach. With the TMOP definition, if $\mathbf{u}$ and $\mathbf{v}$ are aligned, then $\mathbf{u} = \lambda\mathbf{v}$, with $\lambda > 0$. Further, in two-dimensions, $\theta_u = \theta_v$.

In this problem suppose, for the sake of clarity, that one has an initial quadrilateral mesh with globally structured mesh topology. In addition let there be, as raw or intermediate data, a discrete vector field $\{\mathbf{v}_j\}$ defined at points (x_j, y_j) in the physical domain (j is a global index) and possibly at some discrete time t_n. The vector field is processed into a final discrete vector field by interpolating it to the sample points of the mesh (if not already located there). It is assumed that the underlying continuum vector field is 'smooth', i.e., there are no abrupt changes in direction between nearby sample points. Further assume that the vector field is non-self intersecting. The feasibility of aligning mesh lines with a vector field depends heavily on the orientation of the vector field with respect to the domain/mesh boundary. If the boundary vertices are allowed to move, better alignment can be achieved, but this is not always possible.

The goal of the target parameter construction is to specify values of the parameter θ so that the first column of the active matrix (i.e., $\mathbf{a}_1$) is properly oriented with respect to the vector field at every point j. In addition, reasonably good geometric mesh quality is desired. That means in particular, the ideal skew angles are best for the skew parameter data. At this point in the Parameter Control Decision the situation is ? - C - C - ? It is not obvious what to do about the other two target-parameters (volume and aspect ratio) because several choices seem feasible. So, for the time being, let us abstain on aspect ratio and, to assist feasibility, preserve the local size in the 'initial' mesh (i.e., the mesh at time t_n). Then the PCD decision looks like C - C - C - A. Leaving aspect ratio undefined provides some flexibility to the optimizer so that perhaps the alignment can be more effective. Given this decision, the metric type must be VOQ (volume+orientation+skew).

The PIE decision is now easy to complete. The result is P-I-E-A (i.e., preserve volume, improve orientation, and equi-distribute the skew angle). From this, one has directly that $\upsilon = \upsilon(A_{init})$ and $\phi = \phi_{ideal}$. Since $\phi_{ideal} = \frac{\pi}{2}$ for a quadrilateral element (and thus $\sin \phi_{ideal} = 1$), the size parameters is $\zeta = \upsilon$, so ζ is also preserved.

Recall that in two dimensions $\theta(A)$ is the angle between the x-axis and the mesh vector $\mathbf{a}_1$ and $\theta(W)$ is the angle between the x-axis and the target column vector $\mathbf{w}_1$. The challenge is to define values for the orientation parameter $\theta(W)$ at every sample point. The most straight-forward thing to do is to set $\theta(W) = \theta_{\mathbf{v}}$, where the latter is the orientation of the vector $\mathbf{v}$ with respect to the x-axis, at the given sample point. Doing so should make $\theta(A) \sim \theta_{\mathbf{v}}$ in the optimal mesh provided the mesh topology allows it. However, if the goal is to create alignment on a wide class of smooth vector fields, one cannot just try to align $\mathbf{a}_1$ with $\mathbf{v}$ at every sample point. Doing so could produce a very poor quality mesh with some vector fields. It may be better, for example, to align $-\mathbf{a}_1$ with $\mathbf{v}$ at some sample points and to align to $\mathbf{a}_1$ at others. It may also be better to try to align $\pm\mathbf{a}_1^{\perp}$ with $\mathbf{v}$ at other sample points.[7] The decision as to which of these vectors should be aligned with $\mathbf{v}$ has to be made at every sample point. Since there are so many sample points, an automatic way of making the decision is needed. One way

[7] If $\mathbf{w}$ is a non-zero vector in $\Re^2$, then we use the notation $\mathbf{w}^{\perp}$ to indicate the vector which is perpendicular to $\mathbf{w}$ and obeys $det\left[\mathbf{w}, \mathbf{w}^{\perp}\right] > 0$.

to do this goes as follows. Given $\mathbf{a}_1$ from the *initial* mesh and the non-zero vector $\mathbf{v}$ at the same location, compute the corresponding unit vectors $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{v}}$. Then evaluate the two quantities $c \equiv \hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_1$ and $s \equiv det\,(\hat{\mathbf{v}}, \hat{\mathbf{a}}_1)$. Then the alignment decision is made as follows:

1. If $c \geq \frac{1}{\sqrt{2}}$, then align $\mathbf{a}_1$ with $\mathbf{v}$ by setting $\cos\theta_{\mathbf{w}} = \frac{v_x}{|\mathbf{v}|}$ and $\sin\theta_{\mathbf{w}} = \frac{v_y}{|\mathbf{v}|}$,
2. If $c \leq -\frac{1}{\sqrt{2}}$, then align $-\mathbf{a}_1$ with $\mathbf{v}$ by setting $\cos\theta_{\mathbf{w}} = -\frac{v_x}{|\mathbf{v}|}$ and $\sin\theta_{\mathbf{w}} = -\frac{v_y}{|\mathbf{v}|}$,
3. If $s > \frac{1}{\sqrt{2}}$, then align $\mathbf{a}_1^{\perp}$ with $\mathbf{v}$ by setting $\cos\theta_{\mathbf{w}} = \frac{v_y}{|\mathbf{v}|}$ and $\sin\theta_{\mathbf{w}} = -\frac{v_x}{|\mathbf{v}|}$,
4. If $s < -\frac{1}{\sqrt{2}}$, then align $-\mathbf{a}_1^{\perp}$ with $\mathbf{v}$ by setting $\cos\theta_{\mathbf{w}} = -\frac{v_y}{|\mathbf{v}|}$ and $\sin\theta_{\mathbf{w}} = \frac{v_x}{|\mathbf{v}|}$.

That completes the definition of the three target parameters ζ, θ, ϕ for this alignment problem. One might wonder why we do not try to align the second column vector of A (i.e., $\mathbf{a}_2$) with $\mathbf{v}$ in this scheme. The main reason is that the orientation of $\mathbf{a}_2$ is not directly used in target construction (because its orientation is $\theta + \phi$). Additionally, $\phi = \phi_{ideal}$ has already been specified so that, by controlling θ and ϕ, one is already controlling the orientation of $\mathbf{a}_2$. Notice also that in this scheme, the target parameters are fully defined prior to optimization. This is why $\mathbf{a}_1$ from the initial mesh is used to make the alignment decision in the construction of the orientation parameter. Experience has shown that trying to update target parameters during the optimization procedure can lead to convergence and non-uniqueness issues. Finally, note once again that, because the vector field is associated with a position in physical space, the target parameters must be updated via interpolation or by an advection scheme during the optimization procedure.

There are other contexts aside from a structured quadrilateral mesh in which the alignment of the mesh with a given vector field can be considered. Among them are unstructured quadrilateral meshes, triangle meshes, hybrid meshes, hexahedral, and even tetrahedral meshes (both high order and low order). Except for the triangle mesh case, I have not worked out target construction algorithms for these contexts but believe that it can be done.

3.5.3 The 'Delta' Method of Target Parameter Construction

Prior to optimization the preparation process in TMOP is to construct the target-parameters and to select an appropriate local quality metric. For simplicity, let us concentrate on the two-dimensional mesh case - most of what is said here extends directly to the three-dimensional case as well. In the two-dimensional case, values must be supplied for the four parameters ζ, θ, ϕ, and ρ, at every mesh sample point (unless some have been abstained on). This brings us to the Parameter Control Decision in which one decides whether to control or abstain on each of the four parameters. Once the PCD decision is completed, one has automatically determined the *type* of metric which must be used. Once metric type is decided, one can select a concrete metric of the determined type. How does one decide whether to abstain on a given parameter? Two considerations are relevant. The first consideration is whether or not one *needs* to control the given target-parameter in order to improve the mesh. If there is a need to control the parameter then one cannot abstain on the parameter; if there is no need to control it, then one may be able to abstain. The second consideration is whether or not one has raw and/or intermediate data pertaining to the factor one needs to control. If not, then one is forced to abstain.

The PIE decision has already been described. If one completes this stage, the next step is to define the methods by which the various 'controlled' parameters will be assigned values. Previously, this process has been described as a process which results in a function which takes the raw data and converts it to data for the parameter at each sample point. While this remains valid, there is another way to think about this process. Suppose, instead, that one starts with the *initial* mesh, whose set of Jacobians is $\{A_i\}$. One can evaluate each of the non-abstained parameters on the sample points of the initial mesh, giving $\zeta_i = \zeta(A_{ini})$, $\theta_i = \theta(A_{ini})$, and so on. Define the *initial target-parameter set* as $\{\zeta_i, \theta_i, \phi_i, \rho_i\}$ (or, if one has an actual target-matrix $W_i = W(\zeta_i, \theta_i, \phi_i, \rho_i)$. Using this data in the optimization will, of course, result in an optimal mesh identical to the initial mesh. However, if any of this data is changed, one will have new or partly new target data and a different 'optimal' mesh will result. The change in the data can be described in two ways. As a example, let ζ_i be replaced by ζ_o (where o stands for optimal). Then the challenge is to define ζ_o. On the other hand, let $\zeta_o = \zeta_i + \Delta\zeta$. Then the challenge is to define $\Delta\zeta$ (at every sample point). Thus, the final target-matrix (if there is one) can be expressed as either

$W = W(\zeta_o, \theta_o, \phi_o, \rho_o)$ or as $W = W(\zeta_i + \Delta\zeta, \theta_i + \Delta\theta, \phi_i + \Delta\phi, \rho_i + \Delta\rho)$. In theory, it is possible that for some parameters and some problems it will be easier to define ζ_o directly and in others it will be easier to define $\Delta\zeta$. In practice, however, most of the time it will be easier to define the values ζ_o, υ_o, and perhaps ρ_o in terms of $\Delta\zeta$, $\Delta\upsilon$, and $\Delta\rho$. On the other hand, it is probably easier to specify θ_o and ϕ_o directly, rather than in terms of $\Delta\theta$ and $\Delta\phi$.

3.6 Target Construction Examples for Volume Meshes

Only two examples of Target Construction for Volume Meshes are given in this section because many of the examples of Target Construction for planar meshes given in Section 3.5 straight-forwardly extend to the volume mesh case. The two examples presented are Equi-distribution of Ideal Volume Element Shape and Small Edge Removal, both being examples of geometric target construction. More elaborate target constructions in 3D are best explained within the context of a particular application.

3.6.1 Equi-distribute Local Shape

Examples of optimizing a planar mesh given the element type were discussed in Section 3.5.1.1. The remarks in that section apply equally to the volume mesh case, except for the entries in the shape matrices, which depend on the 3D element type.

For the 3D elements, the ideal shape metrics have the form

$$S_k^{3d} = \begin{pmatrix} 1 & \cos\phi_{12} & \cos\phi_{13} \\ 0 & \sin\phi_{12} & \sin\phi_{13}\cos\chi \\ 0 & 0 & \sin\phi_{13}\sin\chi \end{pmatrix} \begin{pmatrix} \rho_1^{\frac{2}{3}} & 0 & 0 \\ 0 & \rho_2^{\frac{2}{3}} & 0 \\ 0 & 0 & \rho_3^{\frac{2}{3}} \end{pmatrix}$$

Using Table 3.7, the ideal tetrahedron has $\phi_{12} = \frac{\pi}{3}$, $\phi_{13} = \frac{\pi}{3}$, and $\chi = \cos^{-1}\left(\frac{1}{3}\right)$. This gives

$$\begin{aligned} S_k^{tet} &= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2\sqrt{3}} \\ 0 & 0 & \sqrt{\frac{2}{3}} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2\sqrt{3}} \\ 0 & 0 & \sqrt{\frac{2}{3}} \end{pmatrix} \end{aligned}$$

For the ideal hexahedron, $\phi_{12} = \frac{\pi}{2}$, $\phi_{13} = \frac{\pi}{2}$, and $\chi_1 = \frac{\pi}{2}$. This yields

$$S_k^{hex} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

For the ideal triangular prism, $\phi_{12} = \frac{\pi}{2}$, $\phi_{13} = \frac{\pi}{2}$, and $\chi_1 = \frac{\pi}{3}$. This yields

$$S_k^{prism} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{\sqrt{3}}{2} \end{pmatrix}$$

To use this shape-matrix correctly, one must be sure that the dihedral angle is the angle between two square faces of the element. To do that, one must take into account the node numbering within each specific element.

For the ideal pyramid element, $\phi_{12} = \frac{\pi}{2}$, $\phi_{13} = \frac{\pi}{3}$, and $\chi_1 = \frac{\pi}{3}$. This yields

$$S_k^{pyr} = \begin{pmatrix} 1 & 0 & \frac{1}{2} \\ 0 & 1 & \frac{1}{2\sqrt{3}} \\ 0 & 0 & \sqrt{\frac{2}{3}} \end{pmatrix}$$

To use this shape-matrix correctly, one must be sure that the dihedral angle is the angle between a square face and a triangular face of the element. To do that, one must take into account the node numbering within each specific element.

3.6.2 Removing Small Edges in Volume Meshes

Suppose that, at some sample point of a volume mesh, the edge which defines ℓ_3 in the initial mesh is unacceptably small. Let $\ell_{min} > 0$ be smallest acceptable length in the mesh and assume $\ell_3 < \ell_{min}$. The goal is to construct the target-parameters such that in the optimized mesh the small edge length ℓ_3 at the same sample point has been increased. Let the increased edge length be $\ell_3 + \Delta\ell_3$. At the same time, let us preserve the lengths ℓ_1 and ℓ_2, i.e., there should be no change in the other two lengths as a result of the mesh optimization. In fact, let $f \geq 1$ be a user-input and let us suppose that $\ell_3 + \Delta\ell_3 = f\ell_{min}$ after optimization.

Recall that for volume elements, $\zeta = \ell_1\ell_2\ell_3$ and $\ell_i = \left(\zeta\,\rho_i^2\right)^{\frac{1}{3}}$. For $i = 1, 2$, preserve the lengths by requiring

$$\begin{aligned} \Delta\ell_1 &= 0 \\ \Delta\ell_2 &= 0 \end{aligned}$$

Considering the first equation,

$$\begin{aligned} 0 &= \Delta \ell_1 \\ &= \Delta \left(\zeta \rho_1^2\right)^{\frac{1}{3}} \end{aligned}$$

and thus

$$\begin{aligned} 0 &= \Delta \left(\zeta \rho_1^2\right) \\ &= 2\zeta \rho_1 \Delta \rho_1 + \rho_1^2 \Delta \zeta \\ &= 2\zeta \Delta \rho_1 + \rho_1 \Delta \zeta \end{aligned}$$

Similarly,

$$0 = 2\zeta \Delta \rho_2 + \rho_2 \Delta \zeta$$

Eliminate $\Delta\zeta$ between these two relations to obtain

$$\rho_2 \Delta \rho_1 - \rho_1 \Delta \rho_2 = 0$$

From this,

$$\begin{aligned} \Delta \rho_1 &= \beta\, \rho_1 \\ \Delta \rho_2 &= \beta\, \rho_2 \end{aligned}$$

for some $\beta \neq 0$. Substituting these into the original pair above gives

$$\Delta \zeta = -2\beta\zeta$$

Next, since $\rho_1 \rho_2 \rho_3 = 1$,

$$\begin{aligned} \Delta \left(\rho_1 \rho_2 \rho_3\right) &= 0 \\ \rho_1 \rho_2 \Delta \rho_3 + \rho_2 \rho_3 \Delta \rho_1 + \rho_3 \rho_1 \Delta \rho_2 &= 0 \\ \rho_1 \rho_2 \Delta \rho_3 + 2\beta(\rho_1 \rho_2 \rho_3) &= 0 \end{aligned}$$

and thus

$$\Delta \rho_3 = -\frac{2\beta}{\rho_1 \rho_2}$$

However, since $\rho_3 = \frac{\ell_3}{\sqrt{\ell_1 \ell_2}}$,

$$\Delta \rho_3 = \frac{\Delta \ell_3}{\sqrt{\ell_1 \ell_2}}$$

Equating these two expressions yields (after some work)

$$\beta = -\frac{1}{2}\frac{\Delta\ell_3}{\ell_3}$$

Thus, in order that ℓ_3 be increased after optimization, $\beta < 0$ is required. Recalling that $\ell_3 + \Delta\ell_3 = f\ell_{min}$,

$$\beta = \frac{\ell_3 - f\ell_{min}}{2\ell_3}$$

and, since $\ell_3 < \ell_{min} \leq f\ell_{min}$, we see that $\beta < 0$, as required.

Finally, the actual target construction method for increasing the small edge ℓ_3 is to first write the target-matrix in the parametric form given in Section 2.2.4. For the given sample point at which the small edge occurred, set the orientation and skew angle values to their values on the initial mesh. Then replace ρ_1 with $\rho_1 + \Delta\rho_1 = (1+\beta)(\rho_1)_{init}$ (with β given in the last formula above, and with ℓ_3 the value of ℓ_3 on the initial mesh). Also replace ρ_2 with $(1+\beta)(\rho_2)_{init}$ and ζ with $\zeta + \Delta\zeta = (1-2\beta)(\zeta)_{init}$.

The cases in which ℓ_1 is a small edge or in which ℓ_2 is a small edge are not worked out here, but doing so should be straight forward.

In practice, one can find every sample point in the initial mesh at which $\ell_3 < \ell_{min}$ and construct targets at each of these sample points accordingly.

Chapter 4

Summary and Conclusion

The Target-matrix Paradigm is a methodology for mesh quality improvement that uses optimization to find optimal locations of mesh vertices and nodes. The TMOP methodology is general in that it can be applied to a wide variety of mesh quality improvement problems, rather being limited to just one situation. An important feature of TMOP is the use of Target-matrices (or Target-parameters) to define the local geometric properties of the desired optimal mesh.

Chapter 2 investigated the relationship between $d \times d$ matrices (with $d = 2, 3$) and a standard set of geometric parameters representing local volume, orientation, skew, and aspect ratio. With certain restrictions, the relationship between the matrices and the parameters can be represented as a one-to-one, onto map. Given the matrix, one can find the geometric parameters; this process is called parameter extraction. Given the geometric parameters, one can find the corresponding matrix. In the context of TMOP, the matrices represent Jacobian matrices of the element mapping at the mesh sample points. Thus, one can say that the geometric parameters represent the first-order geometric quality of the mesh (qualities such as curvature cannot be described by the Jacobian matrix). Given the active mesh, i.e., the mesh that is being optimized, one can find its active Jacobian matrices. In turn, one can extract values of the geometric parameters from the active matrices. These values are useful in the assessment of mesh quality and in the evaluation of local quality metrics in the optimization procedure. On the other hand, given the values of the target geometric parameters, one can find the Target-matrix. While the idea of matrices having geometric content is not new, the contribution in this chapter is to identify a standard set of geometric parameters based on the Jacobian matrix which can be used in Target Construction.

In TMOP, every local quality metric is defined in terms of a set of active and corresponding target parameters. The active parameter values are extracted from the active mesh while the target parameter values are defined prior to the mesh optimization procedure. Chapter 3 describes the process in TMOP by which new Target Construction algorithms can be devised. This process is broken into three phases: that of developing a construction strategy, developing algorithms for converting raw to intermediate data, and developing models for converting intermediate data into final target parameter data. In developing the strategy one begins with a mesh quality improvement goal that is appropriate for the application. Next, one considers the mesh and simulation context. In the Target Parameter Control Decision phase, one decides whether to control or abstain on each geometric parameter. The fact that one can abstain on a parameter means, in general, that (1) values of the corresponding target parameters need not be assigned and (2) that there may be increased improvement in the parameters that are controlled. This flexibility gives greater freedom in the design of mesh optimization algorithms. Finally, correspondence chains are defined that associate specific raw, intermediate, and target data to one another so that the flow between data types is well-defined. In the next phase of Target Construction, one determines how the raw data in each correspondence chain will be converted into intermediate data (i.e., into mesh or solution functionals). The determination consists of identifying particular existing algorithms which are suitable for the conversion or, in some cases, devising an entirely new algorithm. Examples of such algorithms includes a posteriori error estimation, Hessian or Gradient recovery methods, or other techniques used in mesh adaptivity. In the last phase, one determines how the intermediate data in each correspondence chain will be converted into values of the target parameters at every mesh sample point. Determination may consist of a straight-forward direct conversion of the data (as in the case of skew parameters defined by the ideal element) or may require the development of an ad-hoc model that consists of an analytic formula or function relating, for example, the determinant of a matrix to the local mesh volume parameter. Model development mainly consists of determining the best functional form, as well as selecting (via numerical experiment) the best values of any constants contained within the model. With the completion of the Target Construction process one has a complete description of the data and algorithms which will be used to assign values to the Target-parameters in order to address the mesh quality improvement goal. This description, along with corresponding software, yields a Target Construction Method.

In the ideal situation, Target Construction is not needed because there already exists an appropriate Target Construction Method corresponding to the mesh quality improvement goal, mesh context, and simulation context. If this is not the case, then one may engage in the Target Construction process. It is hoped that, over the long term, a library of Target Construction algorithms will be developed and made available so that the need to engage in Target Construction becomes gradually less. Various examples of Target Construction for both planar and volume meshes were presented in Chapter 3.

Much of the existing mesh optimization literature focuses on the definition or selection of quality metrics and/or the numerical optimization procedure. Many of the proposed quality metrics do not make use of targets. For methods which do include targets, the most glaring gap is the lack of discussion on how to construct a set of targets that are suitable for the particular application at hand. The description of Target Construction in this report is thus considered to fill a major gap in the mesh optimization literature.

Of course, not all of the mesh quality improvement problems listed in Section 1.1 can be solved simply by proper construction of the target-matrix. On the one hand, target-construction is an essential step in mesh quality improvement; with this document there is finally a clear exposition of this topic. On the other hand, we still lack firm solutions to a number of the listed mesh quality improvement problems, especially in view of the large number of contexts in which they may occur. In addition to Target Construction, the solution to these problems may include (a) choosing or defining a quality metric having the right metric type and which can produce unique optimal meshes, (b) choosing the right objective function template, (c) constructing proper trade-off coefficients, and (d) employing other techniques such as the use of a different metric in different parts of the domain or adding other terms to the basic objective function.

The consequence of the Target Parameter Control Decision described in this report is that there are fifteen theoretical types of local quality metrics. The idea of metric types and its consequences will be explored in our next report. The two reports together will constitute the most up to data description of the Target-matrix Paradigm.

Appendix A

Appendices

A.1 Key to Notation Used in this Document

The following tables provide a map from a mathematical symbol used globally in this document to a brief explanation of what the symbol represents. Symbols used only locally in the document are not included in these tables.

Symbol	Represents
$\mathbf{a}^{\perp}$	auxiliary vector in the definition of R and β
$\tilde{\mathbf{a}}^{\perp}$	auxiliary extraction vector used in $\tilde{R}$ and $\tilde{\beta}$
$\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$	the column vectors of the active matrix A
d	as subscript, the number of rows and columns of a matrix as superscript, a degenerate matrix
ℓ_i	the i-th length parameter
$\tilde{\ell}_i$	extraction function for the length of the i-th column vector
$\mathbf{p}_2$	$(\upsilon, \theta, \phi, \rho)$
$\mathbf{p}_2(k)$	$\mathbf{p}_2$ evaluated at the k-th sample point
$\mathbf{p}_3$	$(\upsilon, \theta, \psi, \beta, \phi_{12}, \phi_{13}, \chi, \rho_1, \rho_2)$
$\mathbf{p}_3(k)$	$\mathbf{p}_3$ evaluated at the k-th sample point
$\mathbf{p}_3^{asp}$	(ρ_1, ρ_2)
$\mathbf{p}_3^{ori}$	(θ, ψ, β)
$\mathbf{p}_3^{skw}$	$(\phi_{12}, \phi_{13}, \chi)$
$\mathbf{p}_3^{shp}$	$\mathbf{p}_3^{skw} \cup \mathbf{p}_3^{asp}$
$\mathbf{p}_3^{shs}$	$(\zeta) \cup \mathbf{p}_3^{shp}$
$\mathbf{p}_3^{sso}$	$\mathbf{p}_3^{shs} \cup \mathbf{p}_3^{ori}$
$\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$	the column vectors of $R(\mathbf{p}_3^{ori})$
$\tilde{\mathbf{r}}_1$, $\tilde{\mathbf{r}}_2$, $\tilde{\mathbf{r}}_3$	the column vectors of the extraction matrix $\tilde{R}(X)$
$\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$	the column vectors of the target matrix W
$\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$	the column vectors of the 3×3 matrix X
$\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, $\hat{\mathbf{x}}_3$	the unit column vectors of the matrix X

Table A.1: Key to Notation (Lower Case Roman Letters)

Symbol	Represents
A	the active Jacobian matrix
$A_{i,j}$	the i,j-th element of the active matrix
A_{ini}	the initial active Jacobian matrix
A_k	the active Jacobian matrix at sample point k
A_{ref}	Jacobian matrix at a sample point in a reference mesh
D, $D(\rho)$, $D(\mathbf{p}_3^{asp})$	the parameterized aspect ratio matrix
$\tilde{D}$, $\tilde{D}(X)$	the aspect ratio matrix extraction function
$\mathbf{F}$	mapping from $\mathcal{P}_W$ to $\mathcal{S}_W$
$\tilde{\mathbf{F}}$	mapping from $\mathcal{S}_A$ to $\mathcal{P}_A$
I_d	the $d \times d$ matrix identity
$[I_n]$	Particular intervals on the real line, for $n = 1, 2, 3, 4$
$[I_3](a)$	Interval $[I_d]$ as a function of the real parameter a
Q, $Q(\phi)$, $\tilde{Q}(\mathbf{p}_3^{skw})$	the parameterized skew matrix
$\tilde{Q}$, $\tilde{Q}(X)$	the skew matrix extraction function
R, $R(\theta)$, $R(\mathbf{p}_3^{ori})$	the parameterized orientation matrix
$R_1(\theta, \psi)$	first matrix factor of $R(\mathbf{p}_3^{ori})$
$R_2(\beta)$	second matrix factor of $R(\mathbf{p}_3^{ori})$
$\tilde{R}$, $\tilde{R}(X)$	the orientation matrix extraction function
$\tilde{R}_1(X)$, $\tilde{R}_2(X)$	matrix extraction functions for the factors of $\tilde{R}(X)$
S, $S(\phi, \rho)$, $S(\mathbf{p}_3^{shp})$	the parameterized shape matrix
$\tilde{S}$, $\tilde{S}(X)$	the shape matrix extraction function
T	the weighted Jacobian matrix AW^{-1},
$T_{i,j}$	the i,j-th element of the weighted Jacobian matrix
T_k	the weighted Jacobian matrix at sample point k
U, $U(\zeta, \phi, \rho)$, $U(\mathbf{p}_3^{shs})$	the parameterized shape+size matrix
$\tilde{U}$, $\tilde{U}(X)$	the shape+size matrix extraction function
W	the Target matrix
$W_{i,j}$	the i,j-th element of the target matrix
W_k	the target matrix at sample point k
W_a^4, W^9	Cartesian Products of the intervals $[I_n]$
X, Y	arbitrary matrices in $\mathcal{M}_d$

Table A.2: Key to Notation (Upper Case Roman Letters)

Symbol	Represents
β	third target orientation parameter (volume elements)
$\tilde{\beta}$, $\tilde{\beta}(X)$	extraction function for the third orientation parameter
ζ	the target size parameter
$\tilde{\zeta}$, $\tilde{\zeta}(X)$	the size extraction function
θ	the target orientation parameter (planar elements) or, first target orientation parameter (volume elements)
$\tilde{\theta}$, $\tilde{\theta}(X)$	the orientation extraction function (planar elements) or, extraction function for the first orientation parameter
ρ	the target aspect ratio parameter (planar elements)
$\tilde{\rho}$, $\tilde{\rho}(X)$	the aspect ratio extraction function (planar elements)
ρ_1, ρ_2, ρ_3	the target aspect ratio parameters (volume elements)
$\tilde{\rho}_1$, $\tilde{\rho}_2$, $\tilde{\rho}_3$	the aspect ratio extraction functions (volume elements)
υ	the target volume parameter
$\tilde{\upsilon}$	the volume extraction function
ϕ	the target skew parameter (planar elements)
$\tilde{\phi}$, $\tilde{\phi}(X)$	the skew extraction function (planar elements)
ϕ_{12}, ϕ_{13}, ϕ_{23}	the target skew parameters (volume elements)
$\tilde{\phi}_{12}$, $\tilde{\phi}_{13}$, $\tilde{\phi}_{23}$	the skew extraction functions (volume elements)
χ	target dihedral angle skew parameter (volume elements)
$\tilde{\chi}$, $\tilde{\chi}(X)$	the dihedral angle extraction function (volume elements)
ψ	second target orientation parameter (volume elements)
$\tilde{\psi}$, $\tilde{\psi}(X)$	extraction function for the second orientation parameter
ω	the determinant of the Target-matrix

Table A.3: Key to Notation (Lower Case Greek Letters)

Symbol	Represents
$\{\mathcal{A}\}$	the set of active matrices over all mesh sample points
$\mathcal{M}_d$	the set of $d \times d$ matrices with real elements
$\mathcal{M}_d^s$	the set of singular matrices in $\mathcal{M}_d$
$\mathcal{M}_d^{\sim s}$	the set of non-singular matrices in $\mathcal{M}_d$
$\mathcal{M}_d^d$	the set of degenerate matrices in $\mathcal{M}_d$
$\mathcal{M}_d^{\sim d}$	the set of non-degenerate matrices in $\mathcal{M}_d$
$\mathcal{M}_d^p$	the set of matrices in $\mathcal{M}_d$ whose determinant is positive
$\mathcal{M}_d^{asp}$	the set of aspect ratio matrices in $\mathcal{M}_d$
$\mathcal{M}_d^{rot}$	the set of rotation matrices in $\mathcal{M}_d$
$\mathcal{M}_d^{shp}$	the set of shape matrices in $\mathcal{M}_d$
$\mathcal{M}_d^{shs}$	the set of shape+size matrices in $\mathcal{M}_d$
$\mathcal{M}_d^{skw}$	the set of skew matrices in $\mathcal{M}_d$
$\mathcal{P}_A$	set of allowable active geometric parameter values
$\mathcal{P}_W$	set of allowable target geometric parameter values
$\mathcal{S}_A$	set of matrices to which A is restricted
$\mathcal{S}_W$	set of matrices to which W is restricted
$\Re^d$	Cartesian space of d dimensions
$\{\mathcal{W}\}$	the set of target matrices over all mesh sample points

Table A.4: Key to Notation (Miscellaneous Symbols)

Bibliography

[1] Brackbill, J. *An Adaptive Grid with Directional Control*, J. Comp. Phys., 108:38-50, 1993.

[2] Cao, W., *An interpolation error estimate on anisotropic meshes in $\Re^n$ and optimal meshes for mesh refinement*, SIAM J. Numer. Anal., 45(6):2368-2391, 2007.

[3] Chen, L., Sun, P. Xu, J. *Optimal anisotropic meshes for minimizing interpolation errors in L^p-norm*, Math. Comp., **76**, pp.179-204, 2007.

[4] Coulaud, O. and Loseille, A. *Very High Order Anisotropic Metric-Based Mesh Adaptation in 3D*, Procedia Engr., 163:353-365, 2016.

[5] Du, L. and Yan, N. *Gradient recovery type a posteriori error estimate for finite element approximation on non-uniform meshes*, Adv. in Comp. Math., Vol. 14, Issue 2, pp.175-193, Feb. 2001.

[6] Guo, H. Zhang, Z., Zhao, R. *Hessian Recovery for Finite Element Methods*, Math. Comp., Vol. 86, Number 306, July 2017, pp1671-1692.

[7] Hetmaniuk, U. and Knupp, P. *A Mesh Optimization Algorithm for to Decrease the Maximum Error in Finite Element Computations*, Proceedings of the 17th International Meshing Roundtable, p533- 550, 2008.

[8] Horn, R.A. and Johnson, C.R. *Matrix Analysis*, Cambridge University Press, p112, 1985.

[9] Ilyas, M., Lamichhane, B., and Meylan, M. *A gradient recovery method based on an oblique projection and boundary modification*, ANZIAM J. 58, pp C38-C45, 2017.

[10] Kamenski, L. and Huang, W. *How a nonconvergent recovered Hessian works in Mesh Adaptation*, SiAM J. Num. Anal., 2014.

[11] Knupp, P. *Introducing the Target-matrix Paradigm for Mesh Optimization via Node-movement*, Engr. with Cmptrs., 28(4):419-429, 2012.

[12] Lakkis, O. and Pryer, T. *Gradient recovery in adaptive finite-element methods for parabolic problems*, IMA J. Num. Anal., Vol. 32, Issue 1, p246-278, Jan. 2012.

[13] Lipnikov, K. and Vassilveski, Y. *Analysis of Hessian Recovery Methods for Generating Adaptive Meshes*, Proceedings of the 15th Intl. Meshing Roundtable, p163-171, 2007.

[14] Quadros W.R., Vyas V., Brewer M., Owen S.J., Shimada K. (2005) A Computational Framework for Generating Sizing Function in Assembly Meshing. In: Hanks B.W. (eds) Proceedings of the 14th International Meshing Roundtable. Springer, Berlin, Heidelberg.

[15] Thompson, J., Thames, F., Mastin, C. *Automatic numerical grid generation of boundary-fitted curvilinear coordinates for field containing any number of arbitrary two-dimensional bodies*, J. Comp. Phys., **15**, pp299-319, 1974.

[16] Winslow, A. *Numerical solution of the quasilinear Poisson equations in a non-uniform triangle mesh*, J. Comp. Phys., **2**, pp149-172, 1967.

[17] Zienkiewicz, O. and Zhu, J. *The superconvergence patch recovery and a posteriori error estimates, Part I*, Int. J. Numer. Methods Engrg., **33**, p1331-1364, 1992.