# HIGHER ORDER MESH CURVING USING GEOMETRY CURVATURE EXTRAPOLATION 

By<br>Kristen Catherine Karman

| C. Bruce Hilbert | James C. Newman III |
| :--- | :--- |
| Algorithms and Optimization Engineer | Professor, Department Head of Mechanical <br> Branch Technology, Inc. <br> (Chair) |
|  | Engineering <br> (Committee Member) |
|  |  |
| Kidambi Sreenivas | Matt Matthews |
| Associate Professor | Professor, Associate Dean of the College of |
| (Committee Member) | Arts \& Sciences <br> (Committee Member) |
|  |  |

# HIGHER ORDER MESH CURVING USING GEOMETRY 

## CURVATURE EXTRAPOLATION

By<br>Kristen Catherine Karman

A Dissertation Submitted to the Faculty of the University of Tennessee at Chattanooga in Partial Fulfillment of the Requirements of the Degree of Doctor of Philosophy in Computational Engineering

The University of Tennessee at Chattanooga Chattanooga, Tennessee

December 2017

Copyright © 2017
By Kristen Catherine Karman
All Rights Reserved


#### Abstract

A higher order mesh curving method is developed which uses information from the geometry to determine the appropriate curvature of edges in the interior of the mesh. Edges are represented using four point Bézier curves to determine the positions of higher order edge points. Higher order face and volume points are positioned using the basis functions for serendipity face and volume elements. Parameters are defined which allow user specified control over element quality and the propagation of curvature in the mesh. Curved higher order meshes are shown for test cases in both two and three dimensions.


## DEDICATION

This dissertation is dedicated to my husband, Lawton Shoemake, and to our wonderful pets Reggie (dog), Ella (dog), and Taco (cat).

## ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. C. Bruce Hilbert, for his help and support and Branch Technology, Inc. for allowing Dr. Hilbert to serve as my advisor. I would also like to thank Dr. James Newman, Dr. Kidambi Sreenivas, and Dr. Matt Matthews for serving on my dissertation committee and the entire SimCenter faculty and staff for creating a friendly and helpful environment for education and research.

I would also like to thank my parents, Debra and Steve Karman, and my in-laws, Jerry and Sandra Shoemake, for their continuing support. Finally, I would like to thank my husband, Lawton Shoemake, who has been a source of constant love and support and who has spent many hours helping me debug. Thank you.

## TABLE OF CONTENTS

ABSTRACT ..... iv
DEDICATION ..... v
ACKNOWLEDGEMENTS ..... vi
LIST OF TABLES ..... ix
LIST OF FIGURES ..... X
CHAPTER

1. INTRODUCTION ..... 1
1.1 Review of Finite Element Method ..... 2
1.2 Current Curvilinear Meshing Techniques ..... 7
1.2.1 Local Mesh Modification ..... 8
1.2.2 Energy Models ..... 9
1.2.2.1 Winslow ..... 10
1.2.2.2 Optimization Smoothing ..... 10
1.2.2.3 Spring Analogy ..... 11
1.2.2.4 Solid Mechanics Analogy ..... 11
1.2.3 Summary ..... 13
2. METHODOLOGY ..... 14
2.1 Geometry ..... 15
2.1.1 Evaluation of Tangent, Normal, and Curvature ..... 15
2.1.2 Extrapolation of Curvature Information ..... 19
2.2 Edge Elevation ..... 22
2.2.1 Interpolating Curves ..... 22
2.2.1.1 Lagrange Interpolating Polynomials ..... 23
2.2.1.2 Hermite Interpolating Polynomials ..... 25
2.2.1.3 Bézier Curves. ..... 26
2.2.2 Edge Classification ..... 38
2.2.2.1 Constant Distance Edges ..... 38
2.2.2.2 Radial Edges ..... 41
2.2.2.3 Generic Edges ..... 41
2.2.2.4 Concave Edges ..... 42
2.2.2.5 Multi-Body Edges ..... 46
2.3 Face and Volume Elevation ..... 47
2.4 Scaled Jacobian Quality Metric ..... 49
2.5 Elevation Schemes ..... 50
3. RESULTS ..... 53
3.1 Two Dimensional Cases ..... 53
3.1.1 Circle ..... 53
3.1.2 Four Circles ..... 56
3.1.3 NACA0012 ..... 60
3.1.4 30P30N ..... 63
3.2 Three Dimensional Cases ..... 66
3.2.1 Ellipsoid ..... 66
3.2.2 Sphere ..... 69
3.2.3 Two Spheres ..... 71
4. CONCLUSION ..... 73
4.1 Summary ..... 73
4.2 Recommendations for Future Work ..... 73
REFERENCES ..... 76
APPENDIX
A. QUADRILATERAL SERENDIPITY BASIS FUNCTIONS ..... 79
B. TRIANGULAR SERENDIPITY BASIS FUNCTIONS ..... 84
VITA ..... 89

## LIST OF TABLES

3.1 Comparison of minimum and maximum scaled Jacobian values for mixed element circle cases ..... 56
3.2 Comparison of minimum and maximum scaled Jacobian values for mixed element four circle cases. ..... 58
3.3 Comparison of minimum and maximum scaled Jacobian values for the NACA 0012 cases ..... 62
3.4 Comparison of minimum and maximum scaled Jacobian values for ellipsoid cases ..... 66
3.5 Comparison of minimum and maximum scaled Jacobian values for sphere cases ..... 70
3.6 Comparison of minimum and maximum scaled Jacobian values for viscous two spheres cases ..... 72

## LIST OF FIGURES

1.1 Visualization of boundary conforming anisotropic elements ..... 2
1.2 Visualization of 2D p2 elements ..... 5
1.3 Visualization of a p2 serendipity quadrilateral ..... 6
2.1 Diagram showing adjustment of curvature from geometry surface to point in space ..... 19
2.2 Depiction of piecewise geometry ..... 21
2.3 Visualization of interpolation information for an example edge ..... 23
2.4 Visualization of the Lagrange interpolating polynomial for the example edge, inter- polated value are shown in red ..... 24
2.5 Visualization of the Hermite interpolating polynomial for the example edge, inter- polated values are shown in red ..... 25
2.6 Visualization of the three point Bézier curve for the example edge, interpolated values are shown in red ..... 28
2.7 Four point Bézier curves formed using various values for $\alpha_{1}$ and $\alpha_{2}$ ..... 30
2.8 Visualization of the four point Bézier curve for the example edge, interpolated values are shown in red ..... 36
2.9 Comparison of Hermite and Bézier curves ..... 37
2.10 Visualization of a constant distance edge ..... 39
2.11 Comparison of Bézier parametrization coordinate to arclength based parametriza- tion coordinate ..... 40
2.12 Visualization of a radial edge ..... 41
2.13 Visualization of a generic edge ..... 42
2.14 Determination of concavity using the intersection of normal vectors ..... 43
2.15 Visualization of curving generic edges in concave regions ..... 45
2.16 Visualization of curving an edge which crosses the medial axis ..... 47
2.17 Elevation of a p2 quadrilateral face ..... 48
2.18 Elevation of a p4 triangular face ..... 49
2.19 Visualization of the curving process for minimal curving ..... 52
3.1 Mixed element mesh around a circle ..... 54
3.2 Histograms of scaled Jacobian for mixed element circle cases ..... 55
3.3 Mixed element mesh around four circles ..... 57
3.4 Closer look at scaled Jacobian for the fully curved four circle case ..... 58
3.5 Histograms of scaled Jacobian for the mixed element four circle cases ..... 59
3.6 Results of applying partial curving to NACA0012 airfoil mesh ..... 61
3.7 Histogram of the scaled Jacobian for the NACA 0012 case ..... 62
3.8 Full view of the p4 30P30N airfoil mesh created with partial curving ..... 64
3.9 View of the scaled Jacobian near the main body of the 30P30N airfoil for partial curving ..... 64
3.10 Comparison of the minimal and partial curving for the 30P30N airfoil ..... 65
3.11 Inviscid hexahedral mesh of an ellipsoid ..... 67
3.12 View of the scaled Jacobian for the minimally curved p2 Ellipsoid mesh ..... 67
3.13 Histograms of the scaled Jacobian for ellipsoid cases ..... 68
3.14 Viscous tetrahedral mesh of a sphere. ..... 69
3.15 Histograms of the scaled Jacobian for sphere cases ..... 70
3.16 Viscous tetrahedral mesh around two spheres ..... 71
3.17 Histograms of the scaled Jacobian for the two sphere cases ..... 72
A. 1 Ideal serendipity quadrilaterals ..... 81
B. 1 Ideal serendipity triangles ..... 86

## CHAPTER 1

## INTRODUCTION

High-order finite element methods have been shown to provide many advantages in terms of accuracy and efficiency. However, a piece-wise linear approximation of curved geometries can introduce artificial numerical errors into the solution [1-3]. Thus it is necessary to resolve the geometry using curved elements in order to attain the high levels of accuracy promised by high-order finite element methods [4,5].

In the case of inviscid meshes, where the wall spacing is large compared to the boundary deformation, it is possible to curve the boundary without inverting the element or causing severe degradation to element quality. However, for viscous meshes, where the wall spacing is much smaller compared to the boundary deformation, simply deforming the boundary edge can result in the edge inverting the boundary element and crossing over several layers of interior elements, as shown in figure 1.1a. In this situation, curvature must be applied to at least the layers that were crossed, but possibly to additional layers in order to increase element quality throughout the mesh. Figure 1.1 b shows the minimal amount of curving required to produce elements with positive areas, however the third layer still contains nearly inverted elements. Propagating the curvature further into the interior, as shown in figure 1.1c, produces much better quality elements throughout the entire mesh. The challenge for high-order curvilinear meshing, therefore, is to adequately represent
increasingly complex geometries while also maintaining element validity and maximizing element quality.


Figure 1.1 Visualization of boundary conforming anisotropic elements

### 1.1 Review of Finite Element Method

The finite element method belongs to a family of methods called weighted-residual methods which operate on the weak form of a differential equation. Consider the following differential equation in one dimension:

$$
\begin{equation*}
-\frac{d}{d x}\left(a \frac{d u}{d x}\right)+c u-f(x)=0 \tag{1.1}
\end{equation*}
$$

Multiplying equation 1.1 by a weight function $w$ and integrating over the domain gives

$$
\begin{equation*}
\int_{\Omega} w\left[-\frac{d}{d x}\left(a \frac{d u}{d x}\right)+c u-f\right] d x=0 \tag{1.2}
\end{equation*}
$$

where $\Omega$ is the computational domain. Then integration by parts can be used to transfer the derivatives off of $a \frac{d u}{d x}$ and onto $w$ producing the following equation

$$
\begin{equation*}
\int_{\Omega}\left[a \frac{\partial w}{\partial x} \frac{d u}{d x}+c w u-w f\right] d x-\oint_{\Gamma} w a \frac{d u}{d x} d s=0 \tag{1.3}
\end{equation*}
$$

where $\Gamma$ is the boundary of $\Omega$. This is the weak form of equation 1.1.
In the finite element method, the computational domain is then subdivided into simpler, smaller elements. This allows simple interpolation functions to be used to approximate the variable $u$ over an element as

$$
\begin{equation*}
u(x)=\sum_{i=1}^{n} \phi_{i}(x) u_{i} \tag{1.4}
\end{equation*}
$$

where $\phi_{i}(x)$ are the interpolation (basis) functions for the element and $u_{i}$ are the values of $u$ at the $n$ nodes in the element. Substituting equation 1.4 into 1.3 and evaluating for an element in the domain gives

$$
\begin{equation*}
\int_{x_{a}}^{x_{b}}\left[a \frac{d w}{d x}\left(\sum_{j=1}^{n} u_{j} \phi_{j}(x)\right)+c w\left(\sum_{j=1}^{n} u_{j} \phi_{j}(x)\right)-w f\right] d x-\left[w a \frac{d u}{d x}\right]_{x_{a}}^{x_{b}} i=0 \tag{1.5}
\end{equation*}
$$

In the Galerkin formulation, $w$ is taken to be $\phi$ and so the final form is

$$
\begin{equation*}
\sum_{j=1}^{n} K_{i j} u_{j}-f_{i}-Q_{i}=0 \tag{1.6}
\end{equation*}
$$

where

$$
\begin{align*}
K_{i j} & =\int_{x_{a}}^{x_{b}}\left(a \frac{d \phi_{i}}{d x} \frac{d \phi_{j}}{d x}+c \phi_{i} \phi_{j}\right) d x  \tag{1.7}\\
f_{i} & =\int_{x_{a}}^{x_{b}} f \phi_{i} d x  \tag{1.8}\\
Q_{i} & =\left.a \frac{d u}{d x}\right|_{x_{i}} \tag{1.9}
\end{align*}
$$

and $i=1,2, \cdots, n$. The primary and secondary variables are $u$ and $Q$, respectively. This forms an $n \times n$ system of equations for each element which are then combined to form a global system of equations.

Since $u$ is approximated over the elements, the accuracy of that approximation, and thus the accuracy of the computational model, is determined by the order of the approximation polynomial, usually denoted as $p$. The key point is to select the number and location of the nodes in the element so that the geometry of the element is uniquely defined and continuity between elements is easily imposed [6]. For instance, a linear polynomial in two dimensions has the form

$$
\begin{equation*}
u(x, y)=c_{0}+c_{1} x+c_{2} y \tag{1.10}
\end{equation*}
$$

which has 3 undetermined coefficients. Therefore, if we desire a linear approximation of $u$ over an element, we need at least 3 unique points to determine the 3 coefficients. Similarly, a quadratic polynomial has the form

$$
\begin{equation*}
u(x, y)=c_{0}+c_{1} x+c_{2} y+c_{3} x^{2}+c_{4} x y+c_{5} y^{2} \tag{1.11}
\end{equation*}
$$

which has 6 coefficients, so a quadratic approximation over an element requires 6 unique points. Finite elements are typically referred to by the order of the approximation polynomial. For example a 3 point triangle and a 4 point quadrilateral are considered p1 elements because they can uniquely determine a polynomial of at most degree 1. A 6 point triangle and a 9 point quadrilateral can uniquely determine a polynomial of degree 2 and so are p2 elements. Figure 1.2 shows the point locations and ordering of the standard p2 triangle and quadrilateral reference elements.


Figure 1.2 Visualization of 2D p2 elements

Since the interior points of the higher order finite elements do not contribute to the connectivity between elements, it is sometimes desirable to derive elements without interior points. These types of elements are called serendipity elements. For example, the p2 quadrilateral contains 9 points but only requires 6 to determine the polynomial coefficients. The interior point is therefore not needed for polynomial completeness and so can be removed. Figure 1.3 shows the resulting serendipity quadrilateral.


Figure 1.3 Visualization of a p2 serendipity quadrilateral

Generally, the integrals in equations 1.7 and 1.8 are evaluated using a Gauss-Legendre quadrature rule which allows an integral to be expressed as

$$
\begin{equation*}
\int_{a}^{b} F(x) d x=\int_{-1}^{1} \hat{F}(\xi) d \xi \approx \sum_{I=1}^{r} F\left(\xi_{I}\right) w_{I} \tag{1.12}
\end{equation*}
$$

where $w_{I}$ are the Gauss weights, $\xi_{I}$ are the Gauss points, $\hat{F}=F(x(\xi)) J(\xi)$ is the transformed integrand, and $J$ is the Jacobian of the transformation between $x$ and $\xi$. Due to this required change
of variable, it is customary to define the basis functions for elements in terms of a reference element that uses the natural coordinate $\xi \in[-1,1]$.

In one dimension the transformation between physical coordinate $x$ and natural coordinate $\xi$ is $d x=J d \xi$. In two and three dimensions the transformation takes the form of a matrix [J], called the Jacobian matrix of transformation. This matrix in 2D is given by

$$
[\boldsymbol{J}]=\left[\begin{array}{ll}
\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi}  \tag{1.13}\\
\frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta}
\end{array}\right]
$$

and the transformation is defined as

$$
\begin{equation*}
\binom{\frac{\partial \phi_{i}}{\partial x}}{\frac{\partial \phi_{i}}{\partial y}}=[J]^{-1}\binom{\frac{\partial \phi_{i}}{\partial \xi}}{\frac{\partial \phi_{i}}{\partial \eta}} \tag{1.14}
\end{equation*}
$$

In order to compute derivatives of the basis functions $\phi_{i}$ with respect to physical coordinates, it is necessary and sufficient that $[\boldsymbol{J}]$ be nonsingular. In other words the determinant of the Jacobian matrix must be $J=\operatorname{det}(\boldsymbol{J})>0$ at every point $(\xi, \eta)$ in the element's domain. This measure $J$ is referred to as the Jacobian.

### 1.2 Current Curvilinear Meshing Techniques

The classification of curvilinear meshing techniques has traditionally been separated into direct and indirect methods [4]. The direct approach seeks to build a curvilinear high-order mesh directly from the CAD boundary representation whereas the indirect approach seeks to elevate and untangle a linear mesh generated using pre-existing meshing and smoothing technologies.

In general, creating valid elements of arbitrary order directly from CAD is very computationally intensive, so even the direct methods often generate and elevate linear elements. Regardless of which approach is used, every curvilinear meshing technique requires a way of identifying and correcting invalid elements. The current curvilinear meshing techniques can be loosely categorized into two main groups based on the method of correcting invalid elements: local mesh modification and energy models.

### 1.2.1 Local Mesh Modification

Dey, O'Bara, and Shephard propose a technique for use in solid model applications in [4] which uses local mesh modification tools such as edge/face swapping, splitting, collapsing, and node relocation to correct invalid elements. If these operations fail to fix the invalidity, they attempt to use an analytically defined deformation to curve interior edges/faces before resorting to remeshing the localized region. Although this method proved useful, the prescribed deformation was defined for quadratic elements only, making extension to arbitrary orders difficult. Additionally, this method is not very efficient in regions of high curvature where the curvature must be propagated further into the interior of the mesh.

Luo et al. extended this method in two main ways: first, layers of anisotropic elements are extruded off of the geometry in regions that are likely to contain solution singularities; second, mesh elements are defined geometrically using Bézier functions [7, 8]. The first point allows for better resolution of the solution and thus more complex geometries. The second point allows for elevation to arbitrary orders and for better geometry approximation. The Bézier definition of elements also provides a useful means of propagating curvature in the anisotropic regions: the control points
of all elements in a stack of anisotropic elements are moved based on the movement of the base element's control points. Invalidities elsewhere in the mesh are corrected as before using mesh modification tools. Sahni, Luo, Jansen and Shephard extended this idea for a full boundary layer of anisotropic elements for use in viscous flow simulations [9].

Sherwin and Peiró developed an automatic mesh generation routine which extrudes large prismatic elements off the boundary that encapsulate the boundary layer and uses tetrahedra in the remainder of the mesh [10]. These large prisms are then subdivided into either smaller prisms or tetrahedra to form a valid curved boundary layer. Moxey, Green, Sherwin, and Peiró describe in detail their procedure of subdividing the large prisms by using modal basis functions in [11]. "The main challenge of this technique is that the prismatic layer be subdivided without affecting the rest of the mesh" [11]. This limitation was alleviated in [12] by using a linear elasticity smoothing technique to smooth the transition between the boundary layer and the isotropic tetrahedra region.

### 1.2.2 Energy Models

This group of methods can loosely be thought of as minimizing some measure of mesh energy or distortion. Winslow smoothing seeks to impose characteristics of a smooth computational space onto physical space through a transformation between the two spaces. Optimization smoothing techniques operate by minimizing a function based on elemental quality. The spring and solid mechanics analogies compare the mesh to a physical analog and use the appropriate physical equations to apply deformation.

### 1.2.2.1 Winslow

The Winslow equations are second-order nonlinear elliptic partial differential equations which are derived by applying the Laplacian operator on physical space and recasting in computational space, creating a mapping between computational and physical space which imposes the smoothness of the computational mesh on the physical mesh [13]. Fortunato and Persson developed a formulation of the Winslow equations for use with high-order curvilinear elements where a copy of the undeformed higher order mesh is used as the computational mesh [3]. After projecting the higher order surface nodes onto the geometry, the Winslow equations are solved to determine locations for interior points. Since the undeformed higher order mesh is used as the computational mesh, the original spacing and element quality are preserved and only elements near regions of curvature are changed. The discrete form of the Winslow equations are not guaranteed to produce a valid mapping between compuational and physical spaces, however, it does produce good quality meshes when the mesh resolution is sufficient [3].

### 1.2.2.2 Optimization Smoothing

Toulorge, Geuzaine, Remacle, and Lambrechts use an optimization method in [14] that has an objective function that minimizes the distance between the straight-sided mesh and curved mesh and also uses moving log barriers to ensure that the Jacobians remain in an optimal range. Geuzaine et al. in [15] propose a method which couples this optimization method with a curvature propagation technique to fix invalidities in the boundary layer.

Gargallo-Peiró, Roca, Peraire, and Sarrate propose an optimization method based on a regularized measure of the mesh distortion relative to the linear mesh [2, 16, 17]. Ruiz-Gironés,

Sarrate, and Roca combine this distortion measure with a geometric $\mathcal{L}_{2}$ disparity measure in [18]. Although the resulting technique is non-interpolative, the high-order mesh does converge to the actual geometry in the $\mathcal{L}_{2}$ disparity sense and the technique can be applied to real geometries that contain gaps.

Karman et al. in [19] apply weighted condition number (WCN) optimization smoothing, which was originally developed for smoothing linear meshes, to the problem of higher order meshing. Their technique uses the unperturbed higher order mesh to compute the weight matrices for elements and accomplishes the smoothing by breaking the higher order elements into linear sub-elements and enforcing the shapes of the linear sub-elements.

### 1.2.2.3 Spring Analogy

Liu proposes an automatic meshing technique that first generates a valid linear mesh by using a spring analogy to smooth out discontinuities at each extruding layer, then uses a vectoradding deformation method to propagate curvature from the base surface elements up through the pipe-like structure of the mesh [20,21]. It is important to note that Liu's method does not consist of a smoothing step after the elevation to higher order and although the method is very likely to produce valid curved elements, it is not guaranteed.

### 1.2.2.4 Solid Mechanics Analogy

The solid mechanics analogy methods consider the mesh to be an elastic solid where the boundary deformations are prescribed displacements. The elasticity equations are then solved for the displacements of interior points in the mesh. Although linear elasticity performs well for
small boundary deformations, it does not produce valid meshes for sufficiently large boundary deformations.

Persson and Peraire employ a nonlinear elasticity model with an adaptive Newton-Krylov solver, which breaks up larger deformations into incremental steps if necessary [22]. Xie, Sevilla, Hassan, and Morgan take advantage of the inherent structure of viscous meshes by employing linear elasticity in a layer-by-layer approach within the boundary layer, significantly reducing the problem to only those areas that need curvature [5]. Moxey et al. add thermal stress terms to the linear elasticity model to allow for larger deformations and better control of cell quality than classic linear elasticity [23]. Although this approach can handle larger deformations than the classic linear elasticity approach, sufficiently large deformations must still be broken up and applied incrementally to ensure mesh validity.

Poya, Sevilla, and Gil developed an approach in [24] which unifies all of the solid mechanics approaches. They then propose a new technique, called the Consistent Incrementally Linearised (CIL) technique, which updates both the stresses and the geometry at each increment level unlike other incremental approaches (denoted ILE) which only update the geometry. Poya et al. then performed a comparison of the classic linear elastic, ILE, CIL, and nonlinear approaches. They concluded that the linearized approaches (ILE and CIL) produce meshes of similar quality and are more robust and less computationally intensive than the nonlinear approach, which produces poor quality meshes for higher orders of approximation.

Turner et al. propose a variational framework in [25] which optimizes curvilinear meshes by minimizing an energy functional and show that many of these energy models can be reformulated to serve as the energy functional in this framework. They then proceed to compare the performance and
quality of optimization using linear elasticity, Lagrangian nonlinear elasticity, Winslow smoothing, and distortion functionals in both two and three-dimensional cases with fixed boundary nodes. In all cases, the elasticity methods produced the best quality meshes, with respect to the scaled Jacobian measure, whereas the distortion and Winslow methods had faster convergence rates.

### 1.2.3 Summary

The local mesh modification techniques are quick and apply changes to a very localized portion of the mesh, however there are situations in which these techniques alone fail to correct the invalidities. The various curvature propagation techniques are very efficient and can correct invalidities within the boundary layer, however they must be coupled with some other technique to correct invalidities at the interface of the boundary layer and the isotropic region. Additionally, the curvature propagation techniques require knowledge of the structure of the boundary layer in order to be implemented efficiently, and thus are best suited to an automatic mesh generation framework but not well suited to a posteriori framework, which starts from a valid linear mesh. The energy model techniques are more robust, can be used in either a mesh generation framework or an a posteriori framework, and can be applied to the entire mesh or to a subregion of the mesh. However, many of the energy model techniques require a solution to a system of equations, which can be computationally expensive when applied to large domains.

## CHAPTER 2

## METHODOLOGY

The method of curvature extrapolation presented here is a mesh curving method based on extending the idea presented by Luo et al. in [7]. In their work, elements in the mesh are defined geometrically by Bézier functions and curvature is propagated through the anisotropic layers based on the movement of the base element's control points. Rather than propagating curvature through the mesh, the curvature extrapolation method proposed here uses information from the geometry to explicitly determine the ideal position for the Bézier control points and can be applied to an element anywhere in the mesh. Additionally, this technique seeks to combine the steps of elevation and curving into one using the following procedure:

1. For every edge in the mesh:
(a) Project the endpoints onto the nearest geometry.
(b) Evaluate the curvature, tangent and normal at the projection locations.
(c) Extrapolate the curvature, tangent and normal information out to the endpoints.
(d) Represent the edge using a curve which interpolates the curvature and tangent.
(e) Use the interpolating curve to place higher order points along the edge.
2. For all faces in the mesh, use bounding edges to place interior higher order face points.
3. For all volumes in the mesh, use bounding faces to place interior higher order volume points. This chapter will proceed by first discussing the particulars of the geometry representation and the evaluation of curvature. Then edge, face, and volume elevation will be described in more detail. This chapter will conclude with a presentation of a curving scheme which produces a user-specified amount of curving throughout the mesh.

### 2.1 Geometry

As outlined above, the requirements of the geometry representation for this method are to have the ability to:

1. Find the nearest point on the geometry to a point in space.
2. Evaluate the curvature, tangent, and normal at a given point on the geometry.

Any geometry representation that meets these two requirements can be used in this method of mesh curving. For this study, analytic definitions of geometry were used in both two- and threedimensions. Additionally, Geode, a geometry API being developed by Pointwise, Inc. [26], was used in two dimensions to allow for more complicated geometries being tested. The section will continue with a brief summary of properties of parametric space curves and surfaces, taken from [27], followed by a discussion about applying these geometric properties to points on the interior of the mesh.

### 2.1.1 Evaluation of Tangent, Normal, and Curvature

Consider the geometry as a space curve in two dimensions with parametric representation $\boldsymbol{p}(u)$. Then intrinsic properties of a curve (unit tangent vector, unit normal vector and Gaussian
curvature) can be defined in terms of the first and second derivatives of $\boldsymbol{p}$ with respect to $u$, denoted as $\boldsymbol{p}^{u}$ and $\boldsymbol{p}^{u u}$, respectively. The the unit tangent vector $\hat{\boldsymbol{t}}_{i}$ is defined as:

$$
\begin{equation*}
\hat{\boldsymbol{t}}_{i}=\frac{\boldsymbol{p}_{i}^{u}}{\left|\boldsymbol{p}_{i}^{u}\right|} \tag{2.1}
\end{equation*}
$$

where $i$ denotes a specific value of $u$. The unit normal vector $\hat{\boldsymbol{n}}_{i}$ is defined as:

$$
\begin{align*}
\hat{\boldsymbol{n}}_{i} & =\frac{\boldsymbol{k}_{i}}{\left|\boldsymbol{k}_{i}\right|}  \tag{2.2}\\
\boldsymbol{k}_{i} & =\boldsymbol{p}_{i}^{u u}-\frac{\boldsymbol{p}_{i}^{u u} \cdot \boldsymbol{p}_{i}^{u}}{\left|\boldsymbol{p}_{i}^{u}\right|^{2}} \boldsymbol{p}_{i}^{u} \tag{2.3}
\end{align*}
$$

Finally, the Gaussian curvature at a point $i$ along the space curve is defined as

$$
\begin{align*}
\kappa_{i} & =\frac{1}{\rho_{i}}  \tag{2.4}\\
& =\frac{\left|\boldsymbol{p}_{i}^{u} \times \boldsymbol{p}_{i}^{u u}\right|}{\left|\boldsymbol{p}_{i}^{u}\right|^{3}} \tag{2.5}
\end{align*}
$$

where $\rho_{i}$ is the radius of curvature.
Now consider the geometry as a surface in three dimensions with parametric representation $\boldsymbol{p}(u, w)$. Again, intrinsic properties of a surface can be defined in terms of the derivatives of $\boldsymbol{p}$ with respect to $u$ and $w$. The unit normal vector evaluated at a location $i$ on the surface is defined as:

$$
\begin{equation*}
\hat{\boldsymbol{n}}_{i}=\frac{\boldsymbol{p}_{i}^{u} \times \boldsymbol{p}_{i}^{w}}{\left|\boldsymbol{p}_{i}^{u} \times \boldsymbol{p}_{i}^{w}\right|} \tag{2.6}
\end{equation*}
$$

In three dimensions, curvature is a vector rather than a scalar and Gaussian curvature is defined as $\kappa=\kappa_{1} \kappa_{2}$ where $\kappa_{1}$ and $\kappa_{2}$ are the principal normal curvatures. Recall in the procedure outline at the beginning of this chapter that curvature is applied to the edges of the mesh, regardless of dimension. Thus it is more beneficial to define the tangent vector and curvature using the geodesic curve formed by projecting the edge onto the surface $\boldsymbol{p}(u, w)$. This will result in values for the tangent vector and curvature that are analogous to the 2 D definitions and can be accomplished without actually forming the geodesic curve. The tangent vector, $\hat{\boldsymbol{t}}_{i}$, is defined as the projection of the linear edge onto the tangent plane as follows:

$$
\begin{align*}
& \boldsymbol{t}_{i}=\boldsymbol{e}-\left(\boldsymbol{e} \cdot \hat{\boldsymbol{n}}_{i}\right) \hat{\boldsymbol{n}}_{i}  \tag{2.7}\\
& \hat{\boldsymbol{t}}_{i}=\frac{\boldsymbol{t}_{i}}{\left\|\boldsymbol{t}_{i}\right\|} \tag{2.8}
\end{align*}
$$

where $\boldsymbol{e}=\boldsymbol{p}_{1}-\boldsymbol{p}_{0}$ and $\boldsymbol{p}_{0}, \boldsymbol{p}_{1}$ are the endpoints of the edge. The Gaussian curvature of the geodesic curve is the normal component of the curvature vector, which can be defined in terms of the coefficients of the first and second fundamental forms as:

$$
\begin{equation*}
\kappa_{i}=\frac{L\left(\frac{d u}{d t}\right)^{2}+2 M\left(\frac{d u}{d t} \frac{d w}{d t}\right)+N\left(\frac{d w}{d t}\right)^{2}}{E\left(\frac{d u}{d t}\right)^{2}+2 F\left(\frac{d u}{d t} \frac{d w}{d t}\right)+G\left(\frac{d w}{d t}\right)^{2}} \tag{2.9}
\end{equation*}
$$

where the coefficients of the first fundamental form are

$$
\begin{align*}
& E=\boldsymbol{p}_{i}^{u} \cdot \boldsymbol{p}_{i}^{u}  \tag{2.10}\\
& F=\boldsymbol{p}_{i}^{u} \cdot \boldsymbol{p}_{i}^{w}  \tag{2.11}\\
& G=\boldsymbol{p}_{i}^{w} \cdot \boldsymbol{p}_{i}^{w} \tag{2.12}
\end{align*}
$$

and the coefficients of the second fundamental form are

$$
\begin{align*}
L & =\boldsymbol{p}_{i}^{u u} \cdot \hat{\boldsymbol{n}}_{i}  \tag{2.13}\\
M & =\boldsymbol{p}_{i}^{u w} \cdot \hat{\boldsymbol{n}}_{i}  \tag{2.14}\\
N & =\boldsymbol{p}_{i}^{w w} \cdot \hat{\boldsymbol{n}}_{i} \tag{2.15}
\end{align*}
$$

Multiplying both the numerator and denominator of equation 2.9 by $\frac{d t^{2}}{d u^{2}}$ gives

$$
\begin{equation*}
\kappa_{i}=\frac{L+2 M\left(\frac{d w}{d u}\right)+N\left(\frac{d w}{d u}\right)^{2}}{E+2 F\left(\frac{d w}{d u}\right)+G\left(\frac{d w}{d u}\right)^{2}} \tag{2.16}
\end{equation*}
$$

Finally, $d u$ and $d w$ are the components of $\hat{\boldsymbol{t}}_{i}$ with respect to $\boldsymbol{p}_{i}^{u}$ and $\boldsymbol{p}_{i}^{w}$ and can be found using

$$
\begin{align*}
d u & =\hat{\boldsymbol{t}}_{i} \cdot \boldsymbol{p}_{i}^{u}  \tag{2.17}\\
d w & =\hat{\boldsymbol{t}}_{i} \cdot \boldsymbol{p}_{i}^{w} \tag{2.18}
\end{align*}
$$

### 2.1.2 Extrapolation of Curvature Information

At this point, the curvature, normal, and tangent have been evaluated at the geometry. However, the goal is to define the shape of an edge to imitate the behaviour of the section of geometry it projects to, regardless of the position of the edge. Consider figure 2.1 which shows a point $\boldsymbol{p}_{i}$ that exists at a distance of $d$ from a circular geometry. Projecting $\boldsymbol{p}_{i}$ onto the geometry and evaluating the tangent, normal, and curvature gives


Figure 2.1 Diagram showing adjustment of curvature from geometry surface to point in space

$$
\begin{align*}
& \hat{\boldsymbol{t}}_{g}=\left[\begin{array}{lll}
1 & 0 & 0
\end{array}\right]^{\top}  \tag{2.19}\\
& \hat{\boldsymbol{n}}_{g}=\left[\begin{array}{lll}
0 & 1 & 0
\end{array}\right]^{\top}  \tag{2.20}\\
& \kappa_{g}=\frac{1}{\rho_{g}} \tag{2.21}
\end{align*}
$$

where $\rho_{g}$ is the radius of the circle. Now scale up the geometry such that $\boldsymbol{p}_{i}$ now lies on the geometry. Evaluating the tangent and normal vectors on the scaled up geometry returns $\hat{\boldsymbol{t}}_{i}=\hat{\boldsymbol{t}}_{g}$ and $\hat{\boldsymbol{n}}_{i}=\hat{\boldsymbol{n}}_{g}$. However, the curvature has changed to

$$
\begin{equation*}
\kappa_{i}=\frac{1}{\rho_{i}}=\frac{1}{\rho_{g}+d} \tag{2.22}
\end{equation*}
$$

Thus extrapolation of curvature information into the interior of the mesh can be achieved by setting the tangent and normal vectors equal to that evaluated at the geometry and adjusting the curvature based on the distance the point lies from the geometry as shown in equation 2.22.

An additional problem arises when the geometry is represented piecewise by a collection of curves, as shown in figure 2.2 which depicts a NACA 0012 airfoil broken into 2 pieces, the top airfoil $g_{t}$ and the bottom airfoil $g_{b}$ which intersect at the leading edge and the trailing edge. Considering the point at the trailing edge, evaluation of the tangent and normal vectors using $g_{t}$ and $g_{b}$ gives $\hat{\boldsymbol{t}}_{t}, \hat{\boldsymbol{n}}_{t}$ and $\hat{\boldsymbol{t}}_{b}, \hat{\boldsymbol{n}}_{b}$, respectively. Similarly, evaluation of curvature will yield two different values $\kappa_{t}$ and $\kappa_{b}$ based on which geometry is used. Furthermore, as shown in the figure, extending the normal vectors out forms a region of discontinuity within which all points will project onto the intersection of $g_{t}$ and $g_{b}$. Let $\boldsymbol{p}_{i}$ be a point in this discontinuous region. Then the normal is defined


Figure 2.2 Depiction of piecewise geometry

$$
\begin{equation*}
\hat{\boldsymbol{n}}_{i}=\frac{\boldsymbol{p}_{i}-\boldsymbol{p}_{g}}{\left\|\boldsymbol{p}_{i}-\boldsymbol{p}_{g}\right\|} \tag{2.23}
\end{equation*}
$$

where $\boldsymbol{p}_{g}$ is the intersection of $g_{t}$ and $g_{b}$. Defining $\hat{\boldsymbol{n}}_{i}$ this way produces a smooth transition between $\hat{\boldsymbol{n}}_{t}$ and $\hat{\boldsymbol{n}}_{b}$ throughout the region. A similar transition can be obtained for the tangent $\hat{\boldsymbol{t}}_{i}$ by taking $\hat{\boldsymbol{n}}_{i}$ and rotating by $\frac{\pi}{2}$.

Let $\theta$ be the angle between $\hat{\boldsymbol{n}}_{t}$ and $\hat{\boldsymbol{n}}_{b}$ and let $\theta_{t}, \theta_{b}$ be the angles of $\hat{\boldsymbol{n}}_{i}$ with $\hat{\boldsymbol{n}}_{t}$ and $\hat{\boldsymbol{n}}_{b}$, respectively. Then the curvature $\kappa_{i}$ can be found using a weighted average as

$$
\begin{equation*}
\kappa_{i}=w_{t} \kappa_{t}+w_{b} \kappa_{b} \tag{2.24}
\end{equation*}
$$

where $w_{t}=1-\frac{\theta_{t}}{\theta}$ and $w_{b}=1-\frac{\theta_{b}}{\theta}$. This weighting will return $\kappa_{t}$ if $\hat{\boldsymbol{n}}_{i}=\hat{\boldsymbol{n}}_{t}$ and $\kappa_{b}$ if $\hat{\boldsymbol{n}}_{i}=\hat{\boldsymbol{n}}_{b}$.
In two dimensions, intersections of geometry curves always occur between only two curves. However, in three dimensions intersections can occur between any number of curves and surfaces, meaning that discontinuity exists at both individual points and along curves on the geometry. Due to this additional difficulty, extension to using 3D piecewise geometries is left to future work.

### 2.2 Edge Elevation

Recall from the beginning of this chapter that each edge in the mesh is represented by an interpolating curve whose shape is defined by the nearest geometry entity. This interpolating curve is then used to place higher order points along the curved edge. The remainder of this section presents and compares different choices for interpolating curves and then discusses the elevation procedure for different types of edges in more detail.

### 2.2.1 Interpolating Curves

Following the conventional notation used in [27], the geometric form of a generic interpolating curve be written as

$$
\begin{equation*}
\boldsymbol{p}(u)=\sum_{i} F_{i}(u) \boldsymbol{p}_{i} \tag{2.25}
\end{equation*}
$$

where $u$ is the parametric coordinate of the blending functions $F_{i}(u)$ and $\boldsymbol{p}_{i}$ are geometric coefficients. The geometric form is preferable to other forms because it allows a curve to be defined in terms of conditions at the endpoints (such as locations, tangents, curvature, torsion, etc.) and provides a more intuitive means of controlling the shape of the curve.

The information obtained from the geometry regarding shape are the normal, tangent, and binormal vectors (usually of unit length) and Gaussian curvature. These will be the conditions considered for interpolation. Throughout this section, visualizations will be provided for the different interpolating curves using the example edge endpoints shown in Figure 2.3. Although many different types of interpolating curves exist, we will consider the three most common: Lagrange interpolating polynomials, Hermite interpolating polynomials, and Bézier curves.


Figure 2.3 Visualization of interpolation information for an example edge

### 2.2.1.1 Lagrange Interpolating Polynomials

Given a set a set of $n+1$ distinct points $x_{0}, x_{1}, \ldots, x_{n}$ and a corresponding set of values $y_{0}, y_{1}, \ldots, y_{n}$ there is a unique polynomial of degree at most $n$ such that $p\left(x_{i}\right)=y_{i}$ for $0 \leq i \leq n$
[28]. This is the Lagrange polynomial which has the form

$$
\begin{equation*}
p_{n}(x)=\sum_{i=0}^{n} L_{i}(x) y_{i} \tag{2.26}
\end{equation*}
$$

where $L_{i}(x)$ has the property $L_{i}\left(x_{j}\right)=\delta_{i j}$ and the form

$$
\begin{equation*}
L_{i}(x)=\prod_{j \neq i, j=0}^{n} \frac{x-x_{j}}{x_{i}-x_{j}} \tag{2.27}
\end{equation*}
$$

Lagrange polynomials only interpolate the points themselves and since the initial edge has no interior points, this produces a polynomial of at most degree 1, which gives us back the linear edge, as shown in Figure 2.4. We will instead consider other interpolating polynomials.


Figure 2.4 Visualization of the Lagrange interpolating polynomial for the example edge, interpolated value are shown in red

### 2.2.1.2 Hermite Interpolating Polynomials

Hermite polynomials interpolate both the coordinates and the derivatives at the control points. Given an edge with two endpoints $\boldsymbol{p}_{0}$ and $\boldsymbol{p}_{1}$ and corresponding tangents $\boldsymbol{t}_{0}$ and $\boldsymbol{t}_{1}$, the Hermite interpolation is defined by [27]

$$
\begin{equation*}
\boldsymbol{p}(u)=\left(2 u^{3}-3 u^{2}+1\right) \boldsymbol{p}_{0}+\left(-2 u^{3}+3 u^{2}\right) \boldsymbol{p}_{1}+\left(u^{3}-2 u^{2}+u\right) \boldsymbol{t}_{0}+\left(u^{3}-u^{2}\right) \boldsymbol{t}_{1} \tag{2.28}
\end{equation*}
$$

It is important to note that the general form of the Hermite interpolating polynomial does not require the tangent vectors be normalized. Therefore it is possible to determine the appropriate lengths which would also interpolate the Gaussian curvature at the end points, however this technique was not pursued for this study. Figure 2.5 shows the visualization of the resulting Hermite interpolating curve for the example edge.


Figure 2.5 Visualization of the Hermite interpolating polynomial for the example edge, interpolated values are shown in red

### 2.2.1.3 Bézier Curves

Whereas the Lagrange and Hermite interpolating polynomials interpolate a given set of points and conditions, an alternative approach is to define a curve that can be easily manipulated which approximates a given set of points [27,28]. Bézier started with the principle that any point on a curve segment must be given by a parametric function of the form

$$
\begin{equation*}
\boldsymbol{p}(u)=\sum_{i=0}^{n} \boldsymbol{p}_{i} f_{i}(u) \tag{2.29}
\end{equation*}
$$

where $u$ is the parametric coordinate with the restriction $u \in[0,1]$ and $\boldsymbol{p}_{i}$ represent the $n+1$ vertices of a characteristic polygon (also called control points) [27]. He also set forth the following properties for the blending functions $f_{i}(u)$ :

1. The functions must interpolate the first and last vertex points.
2. The tangent at $\boldsymbol{p}_{0}$ must be given by $\boldsymbol{p}_{1}-\boldsymbol{p}_{0}$ and the tangent at $p_{n}$ must be given by $\boldsymbol{p}_{n}-\boldsymbol{p}_{n-1}$. This allows for direct control of the tangent.
3. The previous requirement was also generalized for higher orders, namely that the $r^{\text {th }}$ derivative at an endpoint must be determined by its $r$ neighboring vertices. This allows for control of the continuity at joints between segments of a composite Bézier curve.
4. The functions $f_{i}(u)$ must be symmetric with respect to $u$ and $(1-u)$. This allows for reversing the sequence of vertex points without altering the shape of the curve.

Bézier chose a family of functions known as the Bernstein polynomials to fulfill these properties [27]. The Bernstein polynomials are given by

$$
\begin{equation*}
B_{i, n}(u)=C(n, i) u^{i}(1-u)^{n-i} \tag{2.30}
\end{equation*}
$$

where $C(n, i)=\frac{n!}{i!(n-i)!}$. The final form of the Bézier curve is given by

$$
\begin{equation*}
\boldsymbol{p}(u)=\sum_{i=0}^{n} \boldsymbol{p}_{i} B_{i, n}(u) \tag{2.31}
\end{equation*}
$$

which produces a polynomial of degree $n$.
Evaluating equations 2.31 for $n=2$ produces the following three point curve

$$
\begin{equation*}
\boldsymbol{p}(u)=(1-u)^{2} \boldsymbol{p}_{0}+2 u(1-u) \boldsymbol{p}_{1}+u^{2} \boldsymbol{p}_{2} \tag{2.32}
\end{equation*}
$$

Recall that the normalized tangents are known at the endpoints $\boldsymbol{p}_{0}, \boldsymbol{p}_{2}$. Therefore, using Bézier's second property we have:

$$
\begin{align*}
& \hat{\boldsymbol{t}}_{0}=\frac{\boldsymbol{p}_{1}-\boldsymbol{p}_{0}}{\left\|\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right\|}  \tag{2.33}\\
& \hat{\boldsymbol{t}}_{2}=\frac{\boldsymbol{p}_{2}-\boldsymbol{p}_{1}}{\left\|\boldsymbol{p}_{2}-\boldsymbol{p}_{1}\right\|} \tag{2.34}
\end{align*}
$$

The only choice for $\boldsymbol{p}_{1}$ which satisfies both conditions is the intersection of the lines defined by $\boldsymbol{p}_{0}+c_{0} \hat{\boldsymbol{t}}_{0}$ and $\boldsymbol{p}_{2}+c_{2} \hat{\boldsymbol{t}}_{2}$. It is important to note that although the tangents have been interpolated
at the endpoints, in general this placement of $\boldsymbol{p}_{1}$ will not interpolate the curvature at the endpoints. The visualization of this curve for the example edge is shown in Figure 2.6.


Figure 2.6 Visualization of the three point Bézier curve for the example edge, interpolated values are shown in red

Now consider the four point Bézier curve produced using $n=3$ in equation 2.31:

$$
\begin{equation*}
\boldsymbol{p}(u)=(1-u)^{3} \boldsymbol{p}_{0}+3 u(1-u)^{2} \boldsymbol{p}_{1}+3 u^{2}(1-u) \boldsymbol{p}_{2}+u^{3} \boldsymbol{p}_{3} \tag{2.35}
\end{equation*}
$$

Using similar logic as before we get the following definitions of the tangents at the endpoints $\boldsymbol{p}_{0}, \boldsymbol{p}_{3}$ :

$$
\begin{align*}
& \hat{\boldsymbol{t}}_{0}=\frac{\boldsymbol{p}_{1}-\boldsymbol{p}_{0}}{\left\|\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right\|}  \tag{2.36}\\
& \hat{\boldsymbol{t}}_{3}=\frac{\boldsymbol{p}_{3}-\boldsymbol{p}_{2}}{\left\|\boldsymbol{p}_{3}-\boldsymbol{p}_{2}\right\|} \tag{2.37}
\end{align*}
$$

which means that $\boldsymbol{p}_{1}, \boldsymbol{p}_{2}$ must lie along the lines formed by:

$$
\begin{align*}
& \boldsymbol{l}_{1}=\boldsymbol{p}_{0}+c_{1} \hat{\boldsymbol{t}}_{0}  \tag{2.38}\\
& \boldsymbol{l}_{2}=\boldsymbol{p}_{3}+c_{2} \hat{\boldsymbol{t}}_{3} \tag{2.39}
\end{align*}
$$

Let $\boldsymbol{p}_{T}$ be the intersection of $\boldsymbol{l}_{1}$ and $\boldsymbol{l}_{2}$. Then equations 2.38 and 2.39 can be rescaled as:

$$
\begin{align*}
& \boldsymbol{l}_{1}=\boldsymbol{p}_{0}+\alpha_{1} \boldsymbol{V}_{1}  \tag{2.40}\\
& \boldsymbol{l}_{2}=\boldsymbol{p}_{3}+\alpha_{2} \boldsymbol{V}_{2} \tag{2.41}
\end{align*}
$$

where $\boldsymbol{V}_{1}=\boldsymbol{p}_{T}-\boldsymbol{p}_{0}$ and $\boldsymbol{V}_{2}=\boldsymbol{p}_{T}-\boldsymbol{p}_{3}$. Now $\boldsymbol{p}_{1}, \boldsymbol{p}_{2}$ can be rewritten as

$$
\begin{align*}
& \boldsymbol{p}_{1}=\boldsymbol{p}_{0}+\alpha_{1} \boldsymbol{V}_{1}  \tag{2.42}\\
& \boldsymbol{p}_{2}=\boldsymbol{p}_{3}+\alpha_{2} \boldsymbol{V}_{2} \tag{2.43}
\end{align*}
$$

The variables $\alpha_{1}, \alpha_{2}$ control the distances along the tangent lines, with a value of 0 returning $\boldsymbol{p}_{0}, \boldsymbol{p}_{3}$, respectively, and a value of 1 returning $\boldsymbol{p}_{T}$.


Figure 2.7 Four point Bézier curves formed using various values for $\alpha_{1}$ and $\alpha_{2}$

Figure 2.7 shows plots of four point Bézier curves produced using different values of $\alpha_{1}$ and $\alpha_{2}$ in equations 2.40 and 2.41 for the following conditions:

$$
\begin{align*}
& \boldsymbol{p}_{0}=\left[\begin{array}{lll}
1 & 0 & 0
\end{array}\right]^{\top}  \tag{2.44}\\
& \boldsymbol{p}_{3}=\left[\begin{array}{lll}
0 & 1 & 0
\end{array}\right]^{\top}  \tag{2.45}\\
& \hat{\boldsymbol{t}}_{0}=\left[\begin{array}{lll}
0 & 1 & 0
\end{array}\right]^{\top}  \tag{2.46}\\
& \hat{\boldsymbol{t}}_{3}=\left[\begin{array}{lll}
1 & 0 & 0
\end{array}\right]^{\top}  \tag{2.47}\\
& \boldsymbol{p}_{T}=\left[\begin{array}{lll}
1 & 1 & 0
\end{array}\right]^{\top} \tag{2.48}
\end{align*}
$$

Notice that $\alpha_{1}, \alpha_{2}<0$ produces inverted curves and $\alpha_{1}, \alpha_{2}>1$ produces curves with inflection points. Thus, it is reasonable to restrict $\alpha_{1}, \alpha_{2} \in[0,1]$ to produce well-behaved edges for finite elements.

For a curve parametrized by coordinate $u$, the curvature at a point $\boldsymbol{p}_{i}$ is given by:

$$
\begin{equation*}
\kappa_{i}=\frac{\left|\boldsymbol{p}_{i}^{u} \times \boldsymbol{p}_{i}^{u u}\right|}{\left|\boldsymbol{p}_{i}^{u}\right|^{3}} \tag{2.49}
\end{equation*}
$$

where $\boldsymbol{p}_{i}^{u}, \boldsymbol{p}_{i}^{u u}$ are the first and second derivatives of $\boldsymbol{p}_{i}$ with respect to $u$. Taking the derivatives of equation 2.35 gives:

$$
\begin{align*}
& \boldsymbol{p}^{u}=-3(1-u)^{2} \boldsymbol{p}_{0}+3(1-u)(1-3 u) \boldsymbol{p}_{1}+3 u(2-3 u) \boldsymbol{p}_{2}+3 u^{2} \boldsymbol{p}_{3}  \tag{2.50}\\
& \boldsymbol{p}^{u u}=6(1-u) \boldsymbol{p}_{0}+3(6 u-4) \boldsymbol{p}_{1}+3(2-6 u) \boldsymbol{p}_{2}+6 u \boldsymbol{p}_{3} \tag{2.51}
\end{align*}
$$

Evaluating equations 2.50 and 2.51 at $u=0$ gives:

$$
\begin{gather*}
\boldsymbol{p}_{0}^{u}=\boldsymbol{p}^{u}(0)=3\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right)  \tag{2.52}\\
\boldsymbol{p}_{0}^{u u}=\boldsymbol{p}^{u u}(0)=6\left(\boldsymbol{p}_{0}-2 \boldsymbol{p}_{1}+\boldsymbol{p}_{2}\right) \tag{2.53}
\end{gather*}
$$

Substituting equations 2.52 and 2.53 into equation 2.49 gives the following definition for $\kappa_{0}$ :

$$
\begin{align*}
\kappa_{0} & =\frac{\left|\boldsymbol{p}_{0}^{u} \times \boldsymbol{p}_{0}^{u u}\right|}{\left|\boldsymbol{p}_{0}^{u}\right|^{3}}  \tag{2.54}\\
& =\frac{\left|3\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right) \times 6\left(\boldsymbol{p}_{0}-2 \boldsymbol{p}_{1}+\boldsymbol{p}_{2}\right)\right|}{\left|3\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right)\right|^{3}}  \tag{2.55}\\
& =\frac{18\left|\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right) \times\left(\boldsymbol{p}_{0}-\boldsymbol{p}_{1}+\boldsymbol{p}_{2}-\boldsymbol{p}_{1}\right)\right|}{27\left|\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right|^{3}}  \tag{2.56}\\
& =\frac{2\left|\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right) \times\left(\boldsymbol{p}_{0}-\boldsymbol{p}_{1}\right)+\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right) \times\left(\boldsymbol{p}_{2}-\boldsymbol{p}_{1}\right)\right|}{3\left|\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right|^{3}}  \tag{2.57}\\
& =\frac{2\left|\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right) \times\left(\boldsymbol{p}_{2}-\boldsymbol{p}_{1}\right)\right|}{3\left|\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right|^{3}} \tag{2.58}
\end{align*}
$$

Using similar logic gives the following expression for $\kappa_{3}$ :

$$
\begin{equation*}
\kappa_{3}=\frac{2\left|\left(\boldsymbol{p}_{2}-\boldsymbol{p}_{1}\right) \times\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{2}\right)\right|}{3\left|\boldsymbol{p}_{3}-\boldsymbol{p}_{2}\right|^{3}} \tag{2.59}
\end{equation*}
$$

Substituting the equations 2.40 and 2.41 in for $\boldsymbol{p}_{1}$ and $\boldsymbol{p}_{2}$ gives:

$$
\begin{align*}
\kappa_{0} & =\frac{2\left|\left(\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right) \times\left(\boldsymbol{p}_{2}-\boldsymbol{p}_{1}\right)\right|}{3\left|\boldsymbol{p}_{1}-\boldsymbol{p}_{0}\right|^{3}}  \tag{2.60}\\
& =\frac{2\left|\left(\boldsymbol{p}_{0}+\alpha_{1} \boldsymbol{V}_{1}-\boldsymbol{p}_{0}\right) \times\left(\boldsymbol{p}_{3}+\alpha_{2} \boldsymbol{V}_{2}-\boldsymbol{p}_{0}-\alpha_{1} \boldsymbol{V}_{1}\right)\right|}{3\left|\boldsymbol{p}_{0}+\alpha_{1} \boldsymbol{V}_{1}-\boldsymbol{p}_{0}\right|^{3}}  \tag{2.61}\\
& =\frac{2\left|\alpha_{1}\right|\left|\boldsymbol{V}_{1} \times\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{0}+\alpha_{2} \boldsymbol{V}_{2}-\alpha_{1} \boldsymbol{V}_{1}\right)\right|}{3\left|\alpha_{1}\right|^{3}\left|\boldsymbol{V}_{1}\right|^{3}}  \tag{2.62}\\
& =\frac{2 \alpha_{1}\left|\boldsymbol{V}_{1} \times\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{0}\right)+\boldsymbol{V}_{1} \times\left(\alpha_{2} \boldsymbol{V}_{2}\right)-\boldsymbol{V}_{1} \times\left(\alpha_{1} \boldsymbol{V}_{1}\right)\right|}{3 \alpha_{1}^{3}\left|\boldsymbol{V}_{1}\right|^{3}}  \tag{2.63}\\
& =\frac{2\left|\boldsymbol{V}_{1} \times\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{0}\right)+\alpha_{2}\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{1}^{2}\left|\boldsymbol{V}_{1}\right|^{3}} \tag{2.64}
\end{align*}
$$

$$
\begin{equation*}
\kappa_{3}=\frac{2\left|\left(\boldsymbol{p}_{2}-\boldsymbol{p}_{1}\right) \times\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{2}\right)\right|}{3\left|\boldsymbol{p}_{3}-\boldsymbol{p}_{2}\right|^{3}} \tag{2.65}
\end{equation*}
$$

$$
\begin{equation*}
=\frac{2\left|\left(\boldsymbol{p}_{3}+\alpha_{2} \boldsymbol{V}_{2}-\boldsymbol{p}_{0}-\alpha_{1} \boldsymbol{V}_{1}\right) \times\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{3}-\alpha_{2} \boldsymbol{V} \boldsymbol{V}_{2}\right)\right|}{3\left|\boldsymbol{p}_{3}-\boldsymbol{p}_{3}-\alpha_{2} \boldsymbol{V}_{2}\right|^{3}} \tag{2.66}
\end{equation*}
$$

$$
\begin{equation*}
=\frac{2\left|-\alpha_{2}\right|\left|\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{0}+\alpha_{2} \boldsymbol{V}_{2}-\alpha_{1} \boldsymbol{V}_{1}\right) \times \boldsymbol{V}_{2}\right|}{3\left|-\alpha_{2}\right|^{3}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.67}
\end{equation*}
$$

$$
\begin{equation*}
=\frac{2 \alpha_{2}\left|\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{0}\right) \times \boldsymbol{V}_{2}+\left(\alpha_{2} \boldsymbol{V}_{2}\right) \times \boldsymbol{V}_{2}-\left(\alpha_{1} \boldsymbol{V}_{1}\right) \times \boldsymbol{V}_{2}\right|}{3 \alpha_{2}^{3}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.68}
\end{equation*}
$$

$$
\begin{equation*}
=\frac{2\left|\left(\boldsymbol{p}_{3}-\boldsymbol{p}_{0}\right) \times \boldsymbol{V}_{2}-\alpha_{1}\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{2}^{2}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.69}
\end{equation*}
$$

Noticing that

$$
\begin{equation*}
\boldsymbol{V}_{1}-\boldsymbol{V}_{2}=\left(\boldsymbol{p}_{T}-\boldsymbol{p}_{0}\right)-\left(\boldsymbol{p}_{T}-\boldsymbol{p}_{3}\right)=\boldsymbol{p}_{3}-\boldsymbol{p}_{0} \tag{2.70}
\end{equation*}
$$

and substituting into equations 2.64 and 2.69 we can isolate the vector terms as follows:

$$
\begin{align*}
\kappa_{0} & =\frac{2\left|\boldsymbol{V}_{1} \times\left(\boldsymbol{V}_{1}-\boldsymbol{V}_{2}\right)+\alpha_{2}\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{1}^{2}\left|\boldsymbol{V}_{1}\right|^{3}}  \tag{2.71}\\
& =\frac{2\left|\boldsymbol{V}_{1} \times \boldsymbol{V}_{1}-\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}+\alpha_{2}\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{1}^{2}\left|\boldsymbol{V}_{1}\right|^{3}}  \tag{2.72}\\
& =\frac{2\left|\left(\alpha_{2}-1\right)\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{1}^{2}\left|\boldsymbol{V}_{1}\right|^{3}}  \tag{2.73}\\
& =\frac{2\left|\alpha_{2}-1\right|\left|\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right|}{3 \alpha_{1}^{2}\left|\boldsymbol{V}_{1}\right|^{3}} \tag{2.74}
\end{align*}
$$

$$
\begin{equation*}
\kappa_{3}=\frac{2\left|\left(\boldsymbol{V}_{1}-\boldsymbol{V}_{2}\right) \times \boldsymbol{V}_{2}-\alpha_{1}\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{2}^{2}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.75}
\end{equation*}
$$

$$
\begin{equation*}
=\frac{2\left|\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}-\boldsymbol{V}_{2} \times \boldsymbol{V}_{2}-\alpha_{1}\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{2}^{2}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.76}
\end{equation*}
$$

$$
\begin{equation*}
=\frac{2\left|\left(1-\alpha_{1}\right)\left(\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right)\right|}{3 \alpha_{2}^{2}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.77}
\end{equation*}
$$

$$
\begin{equation*}
=\frac{2\left|1-\alpha_{1}\right|\left|\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right|}{3 \alpha_{2}^{2}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.78}
\end{equation*}
$$

Since $\alpha_{1}, \alpha_{2} \in[0,1]$ we get $0 \leq 1-\alpha_{1} \leq 1$ and $0 \leq 1-\alpha_{2} \leq 1$, which gives:

$$
\begin{align*}
& \kappa_{0}=\frac{2\left(1-\alpha_{2}\right)\left|\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right|}{3 \alpha_{1}^{2}\left|\boldsymbol{V}_{1}\right|^{3}}  \tag{2.79}\\
& \kappa_{3}=\frac{2\left(1-\alpha_{1}\right)\left|\boldsymbol{V}_{1} \times \boldsymbol{V}_{2}\right|}{3 \alpha_{2}^{2}\left|\boldsymbol{V}_{2}\right|^{3}} \tag{2.80}
\end{align*}
$$

Finally, rearranging terms, we get the following:

$$
\begin{align*}
& C_{1} \alpha_{1}^{2}+\alpha_{2}-1=0  \tag{2.81}\\
& C_{2} \alpha_{2}^{2}+\alpha_{1}-1=0 \tag{2.82}
\end{align*}
$$

where $C_{1}, C_{2}$ are constants defined as

$$
\begin{align*}
C_{1} & =\frac{3 k_{0}\left|\boldsymbol{V}_{1}\right|^{3}}{2\left|\boldsymbol{V}_{1} \times V_{2}\right|}  \tag{2.83}\\
C_{2} & =\frac{3 k_{3}\left|\boldsymbol{V}_{2}\right|^{3}}{2\left|\boldsymbol{V}_{1} \times V_{2}\right|} \tag{2.84}
\end{align*}
$$

Equations 2.81 and 2.82 form a system of nonlinear equations which can be solved iteratively for $\alpha_{1}$ and $\alpha_{2}$ using Newton's method, making sure to enforce the conditions $\alpha_{1}, \alpha_{2} \in[0,1]$. Once $\alpha_{1}, \alpha_{2}$ are found, the ideal positions for $\boldsymbol{p}_{1}, \boldsymbol{p}_{2}$ can be determined using equations 2.42 and 2.43. Figure 2.8 shows the visualization of the four point Bézier curve as derived here.

Figure 2.9 shows the plots of the Hermite interpolating curve as well as the three and four point Bézier curves for a circular and elliptical geometry. In both cases, the four point Bézier curve most closely matches the true shape of the geometry. This can be attributed to the fact that this curve is the only one of the three that interpolates curvature as well as tangents. Therefore, the four point Bézier curve was chosen as the interpolating curve to represent edges in this study.

Figure 2.8 Visualization of the four point Bézier curve for the example edge, interpolated values are shown in red


Figure 2.9 Comparison of Hermite and Bézier curves

### 2.2.2 Edge Classification

Gaussian curvature is defined as $\kappa=\frac{1}{\rho}$ where $\rho$ is the radius of curvature. We are interested in approximating curvature out in space, not just at the geometry. We can achieve this by first projecting the points out in space onto the geometry, evaluating the curvature at the projection sites, and then adjusting the curvature to account for the distance as discussed in section 2.1.2.

However, for an edge in space, this raises a problem. Recall in equation 2.49 that $\kappa$ can be defined using the first and second derivatives. Although the directions of these derivatives are constant along the radius of curvature, the magnitudes are not constant. Therefore, the interpolating curve defined for an edge in space using adjusted values of curvature adequately represents the geometry only if both points lie along a line of constant distance from the geometry. In general, this is not the case. Let us consider the different classifications that exist for edges: constant distance, radial, generic, multi-body, and concave.

### 2.2.2.1 Constant Distance Edges

Edges where both endpoints are the same distance from a geometric body are considered to lie along a line of constant distance, as shown in figure 2.10. Elevation of these edges requires setting the interior two points of the four point Bézier curve as discussed in the previous section and evaluating equation 2.35 at the desired locations along the curve. Consider, for instance, a p 4


Figure 2.10 Visualization of a constant distance edge
edge. Then the higher order points can be evaluated at $u=0.25,0.5,0.75$ as follows:

$$
\begin{align*}
\boldsymbol{p}(0.25) & =\frac{27}{64} \boldsymbol{p}_{0}+\frac{27}{64} \boldsymbol{p}_{1}+\frac{9}{64} \boldsymbol{p}_{2}+\frac{1}{64} \boldsymbol{p}_{3}  \tag{2.85}\\
\boldsymbol{p}(0.5) & =\frac{1}{8} \boldsymbol{p}_{0}+\frac{3}{8} \boldsymbol{p}_{1}+\frac{3}{8} \boldsymbol{p}_{2}+\frac{1}{8} \boldsymbol{p}_{3}  \tag{2.86}\\
\boldsymbol{p}(0.75) & =\frac{1}{64} \boldsymbol{p}_{0}+\frac{9}{64} \boldsymbol{p}_{1}+\frac{27}{64} \boldsymbol{p}_{1}+\frac{27}{64} \boldsymbol{p}_{3} \tag{2.87}
\end{align*}
$$

Equations 2.85-2.87 show that the parameter of the blending function coefficients $u$ in equation 2.35 produces a spacing that reflects the influence of the different control points on the shape of the curve. This can produce undesired clustering, as shown in figure 2.11 which depicts an edge along an ellipse with semi-major axis of $r_{a}=4$ and semi-minor axis of $r_{b}=\frac{1}{4}$. In this case, the distance between the endpoints and the first interior point is shorter than the distance between the first interior points and the midpoint. Simply put, evenly spaced Bézier coordinates $u$ do not produce evenly spaced physical coordinates $\boldsymbol{p}$.


Figure 2.11 Comparison of Bézier parametrization coordinate to arclength based parametrization coordinate

In order to acheive more equally spaced points, the locations of the higher order points need to be chosen based on arclength $s \in[0,1]$ instead of $u$. This can be done in an approximate sense by evaluating a number of points $\boldsymbol{p}_{i}$ for $i=0,1, \ldots, n$ which are equally spaced in $u$ and then summing the Euclidean distance between the points to approximate the total arclength $s_{t}$ as shown in equation 2.88.

$$
\begin{equation*}
s_{t}=\sum_{i=1}^{n}\left\|\boldsymbol{p}_{i}-\boldsymbol{p}_{i-1}\right\| \tag{2.88}
\end{equation*}
$$

Once $s_{t}$ is computed, the higher order points are evaluated by finding the value of $u$ that gives an arclength of $s * s_{t}$. Figure 2.11 also shows this reparametrization for higher order points located at $s=0.25,0.5,0.75$.

### 2.2.2.2 Radial Edges

Radial edges are edges which are orthogonal to the geometry and are identified by detecting that the tangents are parallel $\left(\left|\hat{t}_{0} \cdot \hat{t}_{3}\right|=1\right)$. Since these edges are orthogonal to the geometry, no curving is necessary and the higher order points are spaced equally along the straight edge, as shown in figure 2.12.


Figure 2.12 Visualization of a radial edge

### 2.2.2.3 Generic Edges

Edges that are neither radial nor constant distance but exist somewhere in between are considered generic edges. Since the endpoints exist at different distances from the geometry, curvature cannot be applied directly to these edges. However, as figure 2.13 shows, we can create a point $p_{0}^{\prime}$ at a distance of $d_{0}$ along the normal of $p_{3}$. Similarly, we can create $p_{3}^{\prime}$ at a distance of $d_{3}$ along the normal of $p_{0}$. This allows us to create 2 radial edges $\left(p_{0}^{\prime}-p_{3}\right.$ and $\left.p_{3}^{\prime}-p_{0}\right)$ and 2


Figure 2.13 Visualization of a generic edge
constant distance edges $\left(p_{3}-p_{3}^{\prime}\right.$ and $\left.p_{0}-p_{0}^{\prime}\right)$. These 4 edges form a serendipity quadrilateral and the locations of the higher order points of the generic edge can be computed using

$$
\begin{equation*}
\boldsymbol{p}(\xi)=\sum N_{i}(\xi, \xi) \boldsymbol{p}_{i} \tag{2.89}
\end{equation*}
$$

where $\xi$ is the desired parametric location of the point along the edge, $\boldsymbol{p}_{i}$ are the points of the serendipity quadrilateral, and $N_{i}$ are the basis functions of the serendipity quadrilateral. The specific ordering of points of serendipity quadrilaterals and their respective basis functions can be found in appendix A .

### 2.2.2.4 Concave Edges

So far the methods of curving edges that have been discussed address only edges that project onto a convex region of the geometry. However, most real geometries will have both convex
and concave regions. Consider the projections of an edge's end points onto a geometry and the corresponding normal vectors defined to be pointing toward the mesh. If the normal vectors are not parallel, they can be extended from the projections on the geometry until an intersection, $\boldsymbol{p}_{N}$, is found, as shown in figure 2.14. If the intersection of the normal vectors lies on the same side of the geometry as the mesh, then that region of the geometry is concave and the edge is defined to be in a region of concavity, as shown in figure 2.14 a . Otherwise, the region of the geometry is convex, as shown in figure 2.14b.


Figure 2.14 Determination of concavity using the intersection of normal vectors

Concave regions are problematic for two main reasons. First,the center of curvature creates a singularity in terms of the geometric properties. As a point approaches the center of curvature, $\kappa=\frac{1}{\rho} \rightarrow \infty$. Also, by definition, the center of curvature has inifinitely many closest points on the geometry, thus there are infinitely many definitions for tangent and normal vectors. This makes it
difficult to determine the ideal geometric conditions for points very close to the center of curvature and could potentially produce edges on the interior of the mesh that do not conform to the geometry.

The second problem arises in the formation of serendipity quadrilaterals for elevating generic edges in concave regions, as shown in figure 2.15. Consider the positions of $\boldsymbol{p}_{0}$ and $\boldsymbol{p}_{3}$ shown in figure 2.15 a . If we follow the procedure for placing $\boldsymbol{p}_{0}^{\prime}$ and $\boldsymbol{p}_{3}^{\prime}$, we are able to create a valid serendipity quadrilateral that conforms to the shape of the geometry, as shown. Now consider the positions for $\boldsymbol{p}_{0}$ and $\boldsymbol{p}_{3}$ shifted closer to the center of curvature as shown in figure 2.15 b. Following the same procedure as before creates a serendipity quadrilateral that is very poorly formed and does not conform to the geometry. This is due to the fact that although $\boldsymbol{p}_{3}^{\prime}$ is located along the normal vector of $\boldsymbol{p}_{0}$, it actually projects to a different part of the geometry than $\boldsymbol{p}_{0}$. Since the center of curvature has infinitely many closest points, the only points along any normal vector extending from the geometry that share a projection are those that lie between the geometry and the center of curvature. As long as $\boldsymbol{p}_{0}$ shares a projection with $\boldsymbol{p}_{3}^{\prime}$ and $\boldsymbol{p}_{3}$ with $\boldsymbol{p}_{0}^{\prime}$, a valid serendipity quadrilateral can be formed. If we consider the intersecton of the two normal vectors, $\boldsymbol{p}_{N}$, to be a rough approximation of the center of curvature, then we can define the following amendments to the procedure for elevating edges that are identified in concave regions:

- Constant distant edges: If both points lie between the geometry and $\boldsymbol{p}_{N}$, proceed as in convex regions, otherwise leave the edge straight.
- Generic edges: If all four vertices of the serendipity quadrilateral lie between the geometry and $\boldsymbol{p}_{N}$, proceed as in convex regions, otherwise leave the edge straight.


Figure 2.15 Visualization of curving generic edges in concave regions

### 2.2.2.5 Multi-Body Edges

If the geometry is composed of multiple closed bodies, then there will be edges which cross the medial axis. Let $\boldsymbol{p}_{0}, \boldsymbol{p}_{1}$ be the end points of one such edge and $g_{0}, g_{1}$ be the geometrical bodies. Assume that the closest point projection of $\boldsymbol{p}_{0}$ lies on $g_{0}$. Then we have the following possibilities: the closest point projection of $\boldsymbol{p}_{1}$ either lies on $g_{1}$ or $\boldsymbol{p}_{1}$ is situated equidistant from $g_{0}$ and $g_{1}$. If the latter is true, then elevation can proceed as before using information obtained from $g_{0}$. Otherwise we can compute the curved elevated edges with respect to both $g_{0}$ and $g_{1}$ and take the average position for each higher order point. Figure 2.16 shows an example of such an edge where the endpoints $\boldsymbol{p}_{0}$ and $\boldsymbol{p}_{1}$ project to separate bodies. The green edge is the result of curving with respect to $g_{0}$, the blue edge is the result of curving with respect to $g_{1}$, and the red edge is the average of the blue and green edges. This averaging has the effect that if both geometries have opposing curvature, the resulting edge is straight, as in figure 2.16, but if they curve in the same direction the resulting edge will have the averaged curvature. For any edges that cross multiple medial axes it is possible to determine a preferred curving, however that is beyond the scope of this work and so these edges default to straight.


Figure 2.16 Visualization of curving an edge which crosses the medial axis

### 2.3 Face and Volume Elevation

After all the edges have been elevated, the higher-order edges can be used to form serendipity face elements. Just as with the generic edges, the basis functions of serendipity faces can be used to position higher order face points for the standard face elements. This process comes down to identifying the appropriate parametric coordinates $(\xi, \eta)$ for the higher order face points using the reference element and then evaluating

$$
\begin{equation*}
\boldsymbol{p}(\xi, \eta)=\sum N_{i}(\xi, \eta) \boldsymbol{p}_{i} \tag{2.90}
\end{equation*}
$$

using the edge points and the serendipity basis functions $N_{i}$. For example, consider the p2 quadrilateral shown in figure 2.17a, the points shown in red are the edge points that form the serendipity quadrilateral. This quadrilateral requires one interior face point $\boldsymbol{p}_{9}$, shown in blue. Figure 2.17 b shows that the parametric coordinates of $\boldsymbol{p}_{9}$ are $(\xi=0, \eta=0)$ in the standard p 2
reference quadrilateral, so the physical location of $\boldsymbol{p}_{9}$ is given by

$$
\begin{equation*}
\boldsymbol{p}_{9}=\boldsymbol{p}(0,0)=\sum_{i=1}^{8} N_{i}(0,0) \boldsymbol{p}_{i} \tag{2.91}
\end{equation*}
$$

where $\boldsymbol{p}_{i}$ are the 8 edge points in figure 2.17 a and $N_{i}$ are the 8 basis functions of the p 2 serendipity quadrilateral. Figure 2.18 shows this same process for a p4 triangle, which has 12 edge points and requires 3 interior face points.


Figure 2.17 Elevation of a p2 quadrilateral face

Similarly, once faces have been elevated, the elevated faces form serendipity volumes which are used to evaluate interior higher order volume points. The serendipity shape functions used for quadrilaterals and hexahedrals are the generalized complete Lagrange family presented by Rathod and Kilari in [29] and [30]. The serendipity shape functions used for triangular elements are the Lagrangian boundary described shape functions derived by El-Zafrany and Cookson in [31]. For

(a)

(b)

Figure 2.18 Elevation of a p4 triangular face
simplicity, cases in 3D were restricted to quadratic elements, which does not require serendipity shape functions for tetrahedra. The basis functions for up to p 4 quadrilaterals and triangles are reproduced in appendices A and B .

### 2.4 Scaled Jacobian Quality Metric

The most commonly used metric throughout the literature is the scaled Jacobian, which is defined here as

$$
\begin{equation*}
J_{s}=\frac{J_{\min }}{J_{\max }} \tag{2.92}
\end{equation*}
$$

where $J_{\min }$ and $J_{\max }$ are the minimum and maximum Jacobians evaluated at the gauss points. Since the Jacobian matrix is a mapping between the reference element and the physical element, the scaled Jacobian is a relative measure of the distortion of this mapping across the element. The scaled Jacobian, however, is not a good measure of element quality since even poorly shaped
straight sided elements will have optimal $J_{s}=1$. This is due to the fact that $J$ is often constant over straight sided elements. Even with its flaws, however, the scaled Jacobian is useful for curved meshing since it can be used to detect invalid elements ( $J_{s}<0$ ) and, if we assume the initial linear mesh has good quality elements, then maximizing $J_{s}$ corresponds to minimizing the variation of $J$ over the element, which suggests a well-formed curved element.

### 2.5 Elevation Schemes

Thus far a method has been presented which can create curved elements throughout the mesh. However, curving the entire mesh can be costly and is generally not desirable. Non-curved elements are more beneficial to higher order methods since they have a linear mapping between the reference element and the physical element. This linear mapping reduces the degrees of the integrals that have to be evaluated over the element and simplifies the assembly of the global mass matrix since the mass matrix of the local element becomes a scalar mutliple of the mass matrix of the reference element for straight sided elements [22]. Thus we seek to minimize the total number of curved elements within a mesh while maximizing the element quality.

With this goal in mind, there are two user defined parameters for controlling quality. The first parameter is the target Jacobian, $J_{t} \in(0,1]$, which specifies the target minimum quality for curved elements. The second parameter is the curving distance $d_{c}$, which is defined as $d_{c}=f d^{*}$ where $d^{*}$ is the longest distance from the geometry in the mesh and $f \in[0,1]$ is a user defined multiplier controlling the total amount of curvature. The procedure for curving now becomes an iterative process as follows:

1. Elevate the mesh, curving edges with $d<d_{c}$ and leaving all other edges straight.
2. Evaluate $J_{s}$ for each element. If $J_{s}<J_{t}$, add the element to a list of failed elements.
3. Iterate over the failed list, performing these steps until the list is empty:
(a) If $J_{s}>J_{t}$, remove element from list.
(b) Otherwise,
i. Curve all remaining edges in the element and add affected neighboring elements to the failed list.
ii. If $J_{s}>0$, remove element from list.

This process is shown in figure 2.19 for a simple mesh around a circular geometry. The target Jacobian was set to $J_{t}=0.6$ and $f=0$ specifies that the minimal amount of curving is desired. Initially only the boundary edges were curved, forming inverted elements in the first layer, as shown in figure 2.19a. Correcting the elements of the first layer inverts the elements of the second layer as shown in figure 2.19 b, so the second layer is added to the list for curving. This process continues until the last layer of elements is reached. This layer already has $J>J_{t}$ and so these elements require no additional curving and the process is completed.


Figure 2.19 Visualization of the curving process for minimal curving

## CHAPTER 3

## RESULTS

All images of meshes were obtained using GMSH, which is a free 3D finite element mesh generator and post-processor [32]. GMSH was chosen for its ability to display curved meshes for arbitrary high ordered elements. For all cases $J_{t}=0.6$ and three different values were chosen for $f: f=0.0$ (minimal curving), $f=0.1$ (partial curving), $f=1.0$ (full curving). Even though fully curving a mesh is not typically desired, this was performed so as to fully test the capabilities of the curvature extrapolation method. Higher order meshes were generated for all 2D cases for orders $p=2,3,4$, although only p 4 meshes are shown. The tables for the minimum and maximum scaled Jacobians include all orders of elevation.

### 3.1 Two Dimensional Cases

### 3.1.1 Circle

The first example is designed to show how the proposed method handles meshes created with different meshing strategies. Figure 3.1a shows a linear mixed element mesh created around a circular geometry using the following strategies: structured (upper left), diagonalized structured (upper right), quad-dominant TREX (lower right), and triangular TREX (lower left). The mesh contains 476 edges and 284 faces. Figures $3.1 b-3.1 d$ show the results of applying the three curving schemes. The only difference between the minimal and partial curving schemes for this mesh
occur in the quadrilateral regions of the mesh. The first layer of quadrilateral elements maintained a quality above the threshhold and so the minimal curving scheme did not need to propagate curvature further into the interior.

Table 3.1 shows the extreme values obtained for $J_{s}$ for all orders and curving schemes.
Figure 3.2 shows histograms of $J_{s}$ for the quartic curved meshes. All cases were able to maintain elemental quality above the target threshhold.


Figure 3.1 Mixed element mesh around a circle


Figure 3.2 Histograms of scaled Jacobian for mixed element circle cases

Table 3.1 Comparison of minimum and maximum scaled Jacobian values for mixed element circle cases

| P | Curving | $J_{\min }$ | $J_{\max }$ |
| :--- | :--- | :--- | :--- |
| 1 |  | 0.830064 | 1.0 |
| 2 | Minimal | 0.630112 | 1.0 |
|  | Partial | 0.630112 | 1.0 |
|  | Full | 0.76738 | 0.984489 |
| 3 | Minimal | 0.608873 | 1.0 |
|  | Partial | 0.608873 | 1.0 |
|  | Full | 0.717091 | 0.971073 |
| 4 | Minimal | 0.600055 | 1.0 |
|  | Partial | 0.600055 | 1.0 |
|  | Full | 0.6090536 | 0.967453 |

### 3.1.2 Four Circles

The next example is a mixed element mesh around four equally spaced circles. The mesh contains 4772 edges and 2452 faces. Figure 3.3 shows the initial linear mesh as well as the 3 p4 meshes obtained from the curving schemes. The minimal and partial curving schemes achieve optimal quality within the quadrilateral region, and so do not encounter any edges that cross the medial axis between geometry bodies. However, the fully curved example contains two types of multi-body edges. Edges that cross the medial axis are found in the layer of triangular elements between two circles. Edges that are aligned with the medial axis are found in the region between all four circles. Table 3.2 shows that all cases were able to maintain valid elements, even though the fully curved $p 2$ and $p 4$ meshes contain a minimal $J_{s}$ slightly below the threshhold.

Figure 3.4 shows a gradation of the scaled Jacobian metric for the fully curved mesh where dark color corresponds to poor quality. The worst quality elements are the triangular elements containing an edge that crosses the medial axis. The edges in question remain straight due to the
opposing curvatures of the geometric bodies. Since these elements are aligned with the quadrilateral regions surrounding both geometric bodies, it may be better to use quadrilateral elements in this region instead of triangular elements. The histogram in figure 3.5 shows that very few elements actually fall below the target threshhold.


Figure 3.3 Mixed element mesh around four circles

Table 3.2 Comparison of minimum and maximum scaled Jacobian values for mixed element four circle cases

| P | Curving | $J_{\min }$ | $J_{\max }$ |
| :--- | :--- | :--- | :--- |
| 1 |  | 0.921386 | 1.0 |
| 2 | Minimal | 0.640501 | 1.0 |
|  | Partial | 0.807472 | 1.0 |
|  | Full | 0.77281 | 0.992513 |
| 3 | Minimal | 0.655836 | 1.0 |
|  | Partial | 0.83189 | 1.0 |
|  | Full | 0.59041 | 0.999807 |
| 4 | Minimal | 0.657429 | 1.0 |
|  | Partial | 0.793702 | 1.0 |
|  | Full | 0.542847 | 0.999814 |



Figure 3.4 Closer look at scaled Jacobian for the fully curved four circle case


Figure 3.5 Histograms of scaled Jacobian for the mixed element four circle cases

### 3.1.3 NACA0012

The next case is NACA 0012 airfoil with a viscous boundary layer. This case was chosen for two reasons: first, the NACA 0012 airfoil is a very common test case for two dimensional flow solvers, and second, the geometry is defined as two B-splines with intersection points at the leading and trailing edges. Since the angle between the two definitions for the normal vector is very large at the trailing edge, this case is a good choice for testing the smoothness of the handling of curvature and tangent vectors throughout the discontinuous region. The mesh has 8100 edges and 5354 faces. Figure 3.6 shows the p4 mesh and figure 3.7 shows the histogram of the scaled Jacobian for partial curving. Table 3.3 shows the minimum and maximum values of the scaled Jacobian for all elevation order and curving schemes. This example shows that the method can generate meshes with a smooth transition in curvature even in the presence of a very large discontinuous region emanating from an intersection point of the geometry. Since it is not generally desired to have curved elements around the trailing edge, there is a built in option to turn off the edge curving in regions where the curvature of the geometry is essentially zero.

(a) Full view

(b) Closer look at the trailing edge

Figure 3.6 Results of applying partial curving to NACA0012 airfoil mesh


Figure 3.7 Histogram of the scaled Jacobian for the NACA 0012 case

Table 3.3 Comparison of minimum and maximum scaled Jacobian values for the NACA 0012 cases

| P | Curving | $J_{\min }$ | $J_{\max }$ |
| :--- | :--- | :--- | :--- |
| 1 |  | 1.0 | 1.0 |
| 2 | Minimal | 0.602597 | 1.0 |
|  | Partial | 0.707991 | 1.0 |
|  | Full | 0.717991 | 1.0 |
| 3 | Minimal | 0.602729 | 1.0 |
|  | Partial | 0.524281 | 1.0 |
|  | Full | 0.524281 | 1.0 |
| 4 | Minimal | 0.60444 | 1.0 |
|  | Partial | 0.357641 | 1.0 |
|  | Full | 0.357641 | 1.0 |

### 3.1.4 30P30N

The 30P30N airfoil was chosen because it is a multi-body airfoil that has concave and convex regions as well as sharp angles. Just as with the NACA 0012 case, curvature was allowed around the sharp corners to fully test the capabilities of the method. This mesh contains 19935 edges and 11990 faces, both quadrilateral and triangular. Figure 3.8 shows a full view of the the mixed element partially curved mesh. Figure 3.9 shows the cell quality near the main body of the airfoil. Figures 3.10a and 3.10b show a comparison of minimal and partial curving schemes around the front slat, which contains the regions of greatest concave and convex curvature. The partial curving scheme produces a boundary layer that better conforms to the shape of the geometry and provides a smoother change in element quality.

The poor quality elements away from the geometry in figures 3.9 and 3.10 b contain an edge which has a point that projects onto the intersection of a convex geometry segment and a concave geometry segment. The bad quality elements in figure 3.9 could be fixed by either choosing a smaller value of $f$ or by detecting that one of the geometry segments has no curvature. The problem element of figure 3.10 b however, requires additional logic for handling the curving of edges that are within both a concave region and the discontinuous region formed by a concave and convex geometry segment. It is possible that the edge could be properly handled by ignoring the information from the convex geometry segment, however this was not tested.


Figure 3.8 Full view of the p4 30P30N airfoil mesh created with partial curving

(a)

Figure 3.9 View of the scaled Jacobian near the main body of the 30P30N airfoil for partial curving


Figure 3.10 Comparison of the minimal and partial curving for the 30P30N airfoil

### 3.2 Three Dimensional Cases

The three dimensional cases were restricted to an elevation order of $p=2$, as discussed previously, and were further restricted to meshes containing all tetrahedra or all hexahedra. The remaining parameters are the same as for the two dimensional cases, namely, $f=0.0$ for minimal curving, $f=0.1$ for partial curving, and $f=1.0$ for full curving with $J_{t}=0.6$.

### 3.2.1 Ellipsoid

The first three dimensional case is an inviscid hexahedral mesh around an ellipsoid centered at $(0.0,0.0,0.0)$ with radii of 3 in the x -direction, 2 in the y -direction, and 1 in the z -direction. The mesh contains 74048 edges, 73000 faces, and 24000 hexahedra. Table 3.4 shows the extreme values of the scaled Jacobian for all curving schemes. Figure 3.11 shows an x-plane cut of the initial linear mesh as well as the minimal, partial, and fully curved meshes. Figure 3.12 shows a y-plane cut of the scaled Jacobian for the minimally curved mesh. Figure 3.13 shows histograms of element quality for each of the curving schemes. Although the minimal quality falls below the target threshhold, all elements are valid and all curving schemes produce a similar distribution of quailty. As figure 3.12 shows, the poor quality of these elements is inherited from the linear mesh, which also contained elements below the target threshhold.

Table 3.4 Comparison of minimum and maximum scaled Jacobian values for ellipsoid cases

| P | Curving | $J_{\min }$ | $J_{\max }$ |
| :--- | :--- | :--- | :--- |
| 1 |  | 0.51302 | 0.962559 |
| 2 | Minimal | 0.36534 | 0.902765 |
|  | Partial | 0.36534 | 0.942781 |
|  | Full | 0.36534 | 0.942781 |



Figure 3.11 Inviscid hexahedral mesh of an ellipsoid


Figure 3.12 View of the scaled Jacobian for the minimally curved p2 Ellipsoid mesh


Figure 3.13 Histograms of the scaled Jacobian for ellipsoid cases

### 3.2.2 Sphere

The next case is a viscous tetrahedral mesh around a sphere. This mesh contains 4176 edges, 4000 faces, and 1280 tetrahedra. Figure 3.14 shows an x-plane cut of the initial linear mesh and the minimal, partial, and fully curved higher order meshes. As shown in figure 3.15 and table 3.5, the great majority of elements have $J_{s}>0.8$ for all curving schemes and only the fully curved mesh contained any elements below the quality threshhold.


Figure 3.14 Viscous tetrahedral mesh of a sphere

Table 3.5 Comparison of minimum and maximum scaled Jacobian values for sphere cases

| P | Curving | $J_{\min }$ | $J_{\max }$ |
| :--- | :--- | :--- | :--- |
| 1 |  | 1.0 | 1.0 |
| 2 | Minimal | 0.600114 | 1.0 |
|  | Partial | 0.833532 | 1.0 |
|  | Full | 0.570848 | 1.0 |



Figure 3.15 Histograms of the scaled Jacobian for sphere cases

### 3.2.3 Two Spheres

The final three dimensional case is a tetrahedral viscous mesh around two spheres. This mesh contains 108000 edges, 182060 faces, and 90030 tetrahedra. Figure 3.16 shows a y-plane cut of the initial linear mesh and minimal, partial, and fully curved higher order meshes. As shown in table 3.6 and figure 3.17 , all curving schemes produced meshes that achieved the target quality.


Figure 3.16 Viscous tetrahedral mesh around two spheres

Table 3.6 Comparison of minimum and maximum scaled Jacobian values for viscous two spheres cases

| P | Curving | $J_{\min }$ | $J_{\max }$ |
| :--- | :--- | :--- | :--- |
| 1 |  | 1.0 | 1.0 |
| 2 | Minimal | 0.600114 | 1.0 |
|  | Partial | 0.833532 | 1.0 |
|  | Full | 0.711468 | 1.0 |



Figure 3.17 Histograms of the scaled Jacobian for the two sphere cases

## CHAPTER 4

## CONCLUSION

### 4.1 Summary

The goal of this work was to determine a method of creating valid higher order curvilinear meshes that does not require a global system of equations to be solved. Given a valid initial linear mesh and a definition of the geometry, the proposed method of curvature extrapolation starts by defining an underlying four point Bézier curve for each edge which is defined to imitate the behavior of the geometry. After higher order edge points are placed along these Bézier curves, the serendipity basis functions are used to place higher order face and volume points. A procedure was defined which untangles the mesh and allows the user to specify cell quality and a distance for the propagation of curvature. The examples provided in 2D show that this method is capable of producing valid higher order meshes for real geometries containing concave regions, convex regions, sharp corners, and multiple geometric bodies. This method was also shown to be extensible to 3D, although more work is needed to handle real CAD geometries in 3D.

### 4.2 Recommendations for Future Work

GEODE proved very useful for projecting points and evaluating the tangent, normal, and curvature in both 2D and 3D. However, differences in tolerances between GEODE and the curvature extrapolation code proved difficult to overcome. Additionally, it was very difficult to consistently
determine whether a surface intersected with a curve or another surface, which would enable the geometry to be split into contiguous bodies. Although not entirely necessary, this step is useful for determining whether an edge projects to two separate bodies or just two regions of the same body. It was for these reasons that the 3D examples were limited to analytically defined geometries.

Full extension to 3D would also require the following improvements. First, additional logic is required to handle the discontinuous regions around intersection points and curves in 3D. Secondly, real cases in 3D will be much more computationally intensive than any examples shown in this research and will benefit from parallelization. Finally, the most time intensive parts of this method are the projection of points onto the geometry and the evaluation of cell quality, both of which could be made much more efficient.

The Lagrange serendipity basis functions proved to be a poor choice for determining placement of higher order face and volume points for the following reasons. First, the Lagrange serendipity basis functions for quadrilaterals and hexahedra for p 4 and higher cannot be defined using the edge points alone. Second, definitions of complete Lagrange serendipity basis functions for tetrahedra were not found. Finally, it was observed that the complete serendipity basis functions for p 4 triangles with high aspect ratios placed interior points outside of the bounding edges. Bézier face and volume elements, on the other hand, have the nice property that they are bound by other Bézier elements and have been used elsewhere in the literature. Since the edges are already being represented with Bézier curves, this would be a more natural choice for determining the higher order face and volume points.

Finally, the curvature extrapolation method presented here is purely a mesh curving method and can therefore do little to improve element quality when a poor quality linear mesh is provided.

Coupling this method with an optimization smoothing technique could further improve the quality of meshes generated with this method. Additionally, optimization smoothing can be time intensive when the initial mesh configuration is far from the optimal mesh configuration. The curvature extrapolation method could be used to quickly generate a mesh which is much closer to the optimal mesh, thus reducing the work for an optimization smoothing technique.

## REFERENCES

[1] Bassi, F. and Rebay, S., "High-order accurate discontinuous finite element solution of the 2D Euler equations," Journal of computational physics, Vol. 138, No. 2, 1997, pp. 251-285.
[2] Gargallo-Peiró, A., Roca, X., Peraire, J., and Sarrate, J., "Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes," International Journal for Numerical Methods in Engineering, Vol. 103, No. 5, 2015, pp. 342-363.
[3] Fortunato, M. and Persson, P.-O., "High-order unstructured curved mesh generation using the Winslow equations," Journal of Computational Physics, Vol. 307, 2016, pp. 1-14.
[4] Dey, S., O'Bara, R. M., and Shephard, M. S., "Towards curvilinear meshing in 3D: the case of quadratic simplices," Computer-Aided Design, Vol. 33, No. 3, 2001, pp. 199-209.
[5] Xie, Z. Q., Sevilla, R., Hassan, O., and Morgan, K., "The generation of arbitrary order curved meshes for 3D finite element analysis," Computational Mechanics, 2013, pp. 1-14.
[6] Reddy, J., An Introduction to the finite element method, McGraw-Hill, third edition ed., 2006.
[7] Luo, X.-J., Shephard, M. S., O’bara, R. M., Nastasia, R., and Beall, M. W., "Automatic p-version mesh generation for curved domains," Engineering with Computers, Vol. 20, No. 3, 2004, pp. 271-285.
[8] Luo, X., Shephard, M. S., Lee, L.-Q., Ng, C., and Ge, L., "Tracking adaptive moving mesh refinements in 3d curved domains for large-scale higher order finite element simulations," Proceedings of the 17th International Meshing Roundtable, 2008, pp. 585-601.
[9] Sahni, O., Luo, X., Jansen, K., and Shephard, M., "Curved boundary layer meshing for adaptive viscous flow simulations," Finite Elements in Analysis and Design, Vol. 46, No. 1, 2010, pp. 132-139.
[10] Sherwin, S. and Peiró, J., "Mesh generation in curvilinear domains using high-order elements," International Journal for Numerical Methods in Engineering, Vol. 53, No. 1, 2002, pp. 207223.
[11] Moxey, D., Green, M., Sherwin, S., and Peiró, J., "An isoparametric approach to highorder curvilinear boundary-layer meshing," Computer Methods in Applied Mechanics and Engineering, Vol. 283, 2015, pp. 636-650.
[12] Turner, M., Moxey, D., Sherwin, S. J., and Peiró, J., "Automatic generation of 3d unstructured high-order curvilinear meshes," Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, 2016.
[13] Karman, S. L., Wyman, N., and Steinbrenner, J. P., "Mesh Generation Challenges: A Commercial Software Perspective," 23rd AIAA Computational Fluid Dynamics Conference, 2017, p. 3790.
[14] Toulorge, T., Geuzaine, C., Remacle, J.-F., and Lambrechts, J., "Robust untangling of curvilinear meshes," Journal of Computational Physics, Vol. 254, 2013, pp. 8-26.
[15] Geuzaine, C., Johnen, A., Lambrechts, J., Remacle, J.-F., and Toulorge, T., "The generation of valid curvilinear meshes," IDIHOM: industrialization of high-order methods - a top-down approach, Springer, 2015, pp. 15-39.
[16] Roca, X., Gargallo-Peiró, A., and Sarrate, J., "Defining quality measures for high-order planar triangles and curved mesh generation," Proceedings of the 20th International Meshing Roundtable, 2012, pp. 365-383.
[17] Gargallo-Peiró, A., Roca, X., Peraire, J., and Sarrate, J., "High-order mesh generation on CAD geometries," Proceedings of the VI International Conference on Adaptive Modeling and Simulation (ADMOS 2013). CIMNE, Barcelona, Spain, 2013.
[18] Ruiz-Gironés, E., Sarrate, J., and Roca, X., "Generation of curved high-order meshes with optimal quality and geometric accuracy," Procedia Engineering, Vol. 163, 2016, pp.315-327.
[19] Karman, S. L., Erwin, J. T., Glasby, R. S., and Stefanski, D. L., "High-Order Mesh Curving Using WCN Mesh Optimization," 46th AIAA Fluid Dynamics Conference, 2016, p. 3178.
[20] Liu, T., Wang, L., Karman, S. L., and Hilbert, B., "Automatic 2D high-order viscous mesh generation by Spring-Field and vector-adding," 54th AIAA Aerospace Sciences Meeting, 2016, p. 1673.
[21] Liu, T., Hilbert, C. B., Lin, W., and Hereth, E. A., "Automatic high-order mesh generation with Spring-Field and vector-adding: 3D domain mesh generation extension I," 55th AIAA Aerospace Sciences Meeting, 2017.
[22] Persson, P.-O. and Peraire, J., "Curved mesh generation and mesh refinement using lagrangian solid mechanics," Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit, Vol. 204, 2009.
[23] Moxey, D., Ekelschot, D., Keskin, Ü., Sherwin, S., and Peiró, J., "A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality," Procedia engineering, Vol. 82, 2014, pp. 127-135.
[24] Poya, R., Sevilla, R., and Gil, A. J., "A unified approach for a posteriori high-order curved mesh generation using solid mechanics," Computational Mechanics, Vol. 58, No. 3, 2016, pp. 457-490.
[25] Turner, M., Peiró, J., and Moxey, D., "A variational framework for high-order mesh generation," Procedia Engineering, Vol. 163, 2016, pp. 340-352.
[26] Pointwise, Inc., "Mesh Generation Software for CFD - Pointwise," http://www. pointwise.com/, 2017.
[27] Mortenson, M. E., Geometric Modeling, John Wiley \& Sons, Inc., New York, NY, USA, 1985.
[28] Cheney, W. and Kincaid, D., Numerical Mathematics and Computing, Brooks/Cole, Belmont, CA, sixth ed., 2008.
[29] Rathod, H. and Sridevi, K., "General complete Lagrange family for the cube in finite element interpolations," Computer Methods in Applied Mechanics and Engineering, Vol. 181, No. 1, 2000, pp. 295-344.
[30] Rathod, H. and Sridevi, K., "General complete Lagrange interpolations with applications to three-dimensional finite element analysis," Computer Methods in Applied Mechanics and Engineering, Vol. 190, No. 26, 2001, pp. 3325-3368.
[31] El-Zafrany, A. and Cookson, R., "Derivation of Lagrangian and Hermitian shape functions for triangular elements," International Journal for Numerical Methods in Engineering, Vol. 23, No. 2, 1986, pp. 275-285.
[32] Geuzaine, C. and Remacle, J.-F., "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," International journal for numerical engineering, Vol. 79, No. 11, 2009, pp. 1309-1331.

## APPENDIX A

QUADRILATERAL SERENDIPITY BASIS FUNCTIONS

Presented below are the serendipity quadrilateral basis functions corresponding to the elements in figure A.1. For quadratic and cubic quadrilaterals, the standard definition for the serendipity basis functions form a complete polynomial, however the standard definition forms an incomplete polynomial for higher order elements. The definition of the quartic element and basis functions was derived by Rathod and Kilari [29]. In this work, the centroid location required to complete the polynomial for quartic elements is computed using the quadratic serendipity basis functions with nodes $1,2,3,4,6,9,12$, and 15 . To achieve even higher orders, additional interior points need to be added to complete the polynomial.

Quadratic quadrilateral [6]:

$$
\begin{align*}
& N_{1}(\xi, \eta)=-\frac{1}{4}(1-\xi)(1-\eta)(1+\xi+\eta)  \tag{A.1}\\
& N_{2}(\xi, \eta)=-\frac{1}{4}(1+\xi)(1-\eta)(1-\xi+\eta)  \tag{A.2}\\
& N_{3}(\xi, \eta)=-\frac{1}{4}(1+\xi)(1+\eta)(1-\xi-\eta)  \tag{A.3}\\
& N_{4}(\xi, \eta)=-\frac{1}{4}(1-\xi)(1+\eta)(1+\xi-\eta)  \tag{A.4}\\
& N_{5}(\xi, \eta)=\frac{1}{2}\left(1-\xi^{2}\right)(1-\eta)  \tag{A.5}\\
& N_{6}(\xi, \eta)=\frac{1}{2}(1+\xi)\left(1-\eta^{2}\right)  \tag{A.6}\\
& N_{7}(\xi, \eta)=\frac{1}{2}\left(1-\xi^{2}\right)(1+\eta)  \tag{A.7}\\
& N_{8}(\xi, \eta)=\frac{1}{2}(1-\xi)\left(1-\eta^{2}\right) \tag{A.8}
\end{align*}
$$



Figure A. 1 Ideal serendipity quadrilaterals

Cubic quadrilateral [6]:

$$
\begin{align*}
& N_{1}(\xi, \eta)=\frac{1}{32}(1-\xi)(1-\eta)\left[-10+9\left(\xi^{2}+\eta^{2}\right)\right]  \tag{A.9}\\
& N_{2}(\xi, \eta)=\frac{1}{32}(1+\xi)(1-\eta)\left[-10+9\left(\xi^{2}+\eta^{2}\right)\right]  \tag{A.10}\\
& N_{3}(\xi, \eta)=\frac{1}{32}(1+\xi)(1+\eta)\left[-10+9\left(\xi^{2}+\eta^{2}\right)\right]  \tag{A.11}\\
& N_{4}(\xi, \eta)=\frac{1}{32}(1-\xi)(1+\eta)\left[-10+9\left(\xi^{2}+\eta^{2}\right)\right]  \tag{A.12}\\
& N_{5}(\xi, \eta)=\frac{9}{32}(1-\eta)\left(1-\xi^{2}\right)(1-3 \xi)  \tag{A.13}\\
& N_{6}(\xi, \eta)=\frac{9}{32}(1-\eta)\left(1-\xi^{2}\right)(1+3 \xi)  \tag{A.14}\\
& N_{7}(\xi, \eta)=\frac{9}{32}(1+\xi)\left(1-\eta^{2}\right)(1-3 \eta)  \tag{A.15}\\
& N_{8}(\xi, \eta)=\frac{9}{32}(1+\xi)\left(1-\eta^{2}\right)(1+3 \eta)  \tag{A.16}\\
& N_{9}(\xi, \eta)=\frac{9}{32}(1+\eta)\left(1-\xi^{2}\right)(1+3 \xi)  \tag{A.17}\\
& N_{10}(\xi, \eta)=\frac{9}{32}(1+\eta)\left(1-\xi^{2}\right)(1-3 \xi)  \tag{A.18}\\
& N_{11}(\xi, \eta)=\frac{9}{32}(1-\xi)\left(1-\eta^{2}\right)(1+3 \eta)  \tag{A.19}\\
& N_{12}(\xi, \eta)=\frac{9}{32}(1-\xi)\left(1-\eta^{2}\right)(1-3 \eta) \tag{A.20}
\end{align*}
$$

Quartic quadrilateral [29]:

$$
\begin{align*}
& N_{1}(\xi, \eta)=\frac{1}{12}(1-\xi)(1-\eta)\left[-4\left(\xi^{3}+\eta^{3}\right)+4(\xi+\eta)+3 \xi \eta\right]  \tag{A.21}\\
& N_{2}(\xi, \eta)=\frac{1}{12}(1+\xi)(1-\eta)\left[4\left(\xi^{3}-\eta^{3}\right)-4(\xi-\eta)-3 \xi \eta\right]  \tag{A.22}\\
& N_{3}(\xi, \eta)=\frac{1}{12}(1+\xi)(1+\eta)\left[4\left(\xi^{3}+\eta^{3}\right)-4(\xi+\eta)+3 \xi \eta\right]  \tag{A.23}\\
& N_{4}(\xi, \eta)=\frac{1}{12}(1-\xi)(1+\eta)\left[-4\left(\xi^{3}-\eta^{3}\right)+4(\xi-\eta)-3 \xi \eta\right]  \tag{A.24}\\
& N_{5}(\xi, \eta)=\frac{2}{3}\left(1-\xi^{2}\right)(1-\eta) \xi(2 \xi-1)  \tag{A.25}\\
& N_{6}(\xi, \eta)=-\frac{1}{2}\left(1-\xi^{2}\right)(1-\eta)\left(4 \xi^{2}+\eta\right)  \tag{A.26}\\
& N_{7}(\xi, \eta)=\frac{2}{3}\left(1-\xi^{2}\right)(1-\eta) \xi(2 \xi+1)  \tag{A.27}\\
& N_{8}(\xi, \eta)=\frac{2}{3}(1+\xi)\left(1-\eta^{2}\right) \eta(2 \eta-1)  \tag{A.28}\\
& N_{9}(\xi, \eta)=-\frac{1}{2}(1+\xi)\left(1-\eta^{2}\right)\left(4 \eta^{2}-\xi\right)  \tag{A.29}\\
& N_{10}(\xi, \eta)=\frac{2}{3}(1+\xi)\left(1-\eta^{2}\right) \eta(2 \eta+1)  \tag{A.30}\\
& N_{11}(\xi, \eta)=\frac{2}{3}\left(1-\xi^{2}\right)(1+\eta) \xi(2 \xi+1)  \tag{A.31}\\
& N_{12}(\xi, \eta)=-\frac{1}{2}\left(1-\xi^{2}\right)(1+\eta)\left(4 \xi^{2}-\eta\right)  \tag{A.32}\\
& N_{13}(\xi, \eta)=\frac{2}{3}\left(1-\xi^{2}\right)(1+\eta) \xi(2 \xi-1)  \tag{A.33}\\
& N_{14}(\xi, \eta)=\frac{2}{3}(1-\xi)\left(1-\eta^{2}\right) \eta(2 \eta+1)  \tag{A.34}\\
& N_{15}(\xi, \eta)=-\frac{1}{2}(1-\xi)\left(1-\eta^{2}\right)\left(4 \eta^{2}+\xi\right)  \tag{A.35}\\
& N_{16}(\xi, \eta)=\frac{2}{3}(1-\xi)\left(1-\eta^{2}\right) \eta(2 \eta-1)  \tag{A.36}\\
& N_{17}(\xi, \eta)=\left(1-\xi^{2}\right)\left(1-\eta^{2}\right) \tag{A.37}
\end{align*}
$$

## APPENDIX B

TRIANGULAR SERENDIPITY BASIS FUNCTIONS

Presented below are the serendipity triangle basis functions corresponding to the cubic and quartic elements in figure B.1. El-Zafrany and Cookson derived a general form for deriving basis functions for boundary-described triangles as follows [31]:

Let $a b c$ be a general boundary-described triangular element with sides $a b, b c$, and $c a$ of $m$ th, $n$ th, and $l$ th degree, respectively. Then the shape functions for points along the edges $a b, b c$, and $c a$ can be expressed as

$$
\begin{align*}
& N_{a b, i}=\frac{L_{1}}{2\left(L_{3}+L_{1}\right)} \lambda_{m-i}^{m}\left(L_{3}+L_{1}\right) \lambda_{i}^{m}\left(L_{2}\right)+\frac{L_{2}}{2\left(L_{2}+L_{3}\right)} \lambda_{m-i}^{m}\left(L_{1}\right) \lambda_{i}^{m}\left(L_{2}+L_{3}\right)  \tag{B.1}\\
& N_{b c, j}=\frac{L_{2}}{2\left(L_{1}+L_{2}\right)} \lambda_{n-j}^{n}\left(L_{1}+L_{2}\right) \lambda_{j}^{n}\left(L_{3}\right)+\frac{L_{3}}{2\left(L_{3}+L_{1}\right)} \lambda_{n-j}^{n}\left(L_{2}\right) \lambda_{j}^{n}\left(L_{3}+L_{1}\right)  \tag{B.2}\\
& N_{c a, k}=\frac{L_{3}}{2\left(L_{2}+L_{3}\right)} \lambda_{l-k}^{l}\left(L_{2}+L_{3}\right) \lambda_{k}^{l}\left(L_{1}\right)+\frac{L_{1}}{2\left(L_{1}+L_{2}\right)} \lambda_{l-k}^{l}\left(L_{3}\right) \lambda_{k}^{l}\left(L_{1}+L_{2}\right) \tag{B.3}
\end{align*}
$$

where

$$
\begin{align*}
& 1 \leq i \leq m-1  \tag{B.4}\\
& 1 \leq j \leq n-1  \tag{B.5}\\
& 1 \leq k \leq l-1 \tag{B.6}
\end{align*}
$$

and

$$
\begin{align*}
& \lambda_{0}^{q}\left(L_{r}\right)=1  \tag{B.7}\\
& \lambda_{s}^{q}\left(L_{r}\right)=\prod_{t=0}^{s-1}\left(\frac{q L_{r}-t}{s-t}\right), \quad s=1,2, \cdots, q \tag{B.8}
\end{align*}
$$

The shape functions at the corner nodes can be written in terms of the above expressions as follows:

$$
\begin{align*}
& N_{a}=N_{c a, l}+N_{a b, 0}-\frac{L_{1}}{2}  \tag{B.9}\\
& N_{b}=N_{a b, m}+N_{b c, 0}-\frac{L_{2}}{2}  \tag{B.10}\\
& N_{c}=N_{b c, n}+N_{c a, 0}-\frac{L_{3}}{2} \tag{B.11}
\end{align*}
$$



Figure B. 1 Ideal serendipity triangles

Evaluating the general form for the cubic triangle gives the following basis functions which are complete:

$$
\begin{align*}
& N_{1}\left(L_{1}, L_{2}, L_{3}\right)=\frac{1}{2} L_{1}\left[\left(3 L_{1}-1\right)\left(3 L_{1}-2\right)-9 L_{2} L_{3}\right]  \tag{B.12}\\
& N_{2}\left(L_{1}, L_{2}, L_{3}\right)=\frac{1}{2} L_{2}\left[\left(3 L_{2}-1\right)\left(3 L_{2}-2\right)-9 L_{1} L_{3}\right]  \tag{B.13}\\
& N_{3}\left(L_{1}, L_{2}, L_{3}\right)=\frac{1}{2} L_{3}\left[\left(3 L_{3}-1\right)\left(3 L_{3}-2\right)-9 L_{1} L_{2}\right]  \tag{B.14}\\
& N_{4}\left(L_{1}, L_{2}, L_{3}\right)=\frac{9}{4} L_{1} L_{2}\left[1+3\left(L_{1}-L_{2}\right)\right]  \tag{B.15}\\
& N_{5}\left(L_{1}, L_{2}, L_{3}\right)=\frac{9}{4} L_{1} L_{2}\left[1+3\left(L_{2}-L_{1}\right)\right]  \tag{B.16}\\
& N_{6}\left(L_{1}, L_{2}, L_{3}\right)=\frac{9}{4} L_{2} L_{3}\left[1+3\left(L_{2}-L_{3}\right)\right]  \tag{B.17}\\
& N_{7}\left(L_{1}, L_{2}, L_{3}\right)=\frac{9}{4} L_{2} L_{3}\left[1+3\left(L_{3}-L_{2}\right)\right]  \tag{B.18}\\
& N_{8}\left(L_{1}, L_{2}, L_{3}\right)=\frac{9}{4} L_{1} L_{3}\left[1+3\left(L_{3}-L_{1}\right)\right]  \tag{B.19}\\
& N_{9}\left(L_{1}, L_{2}, L_{3}\right)=\frac{9}{4} L_{1} L_{3}\left[1+3\left(L_{1}-L_{3}\right)\right] \tag{B.20}
\end{align*}
$$

Similarly, evaluating the general form for the quartic triangle gives the following basis functions which are complete:

$$
\begin{align*}
N_{1}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{1}{12} L_{1}\left[\left(4 L_{1}-1\right)\left(4 L_{1}-2\right)\left(4 L_{1}-3\right)\right. \\
& +\left(4 L_{1}+4 L_{2}-1\right)\left(4 L_{1}+4 L_{2}-2\right)\left(4 L_{1}+4 L_{2}-3\right) \\
& \left.+\left(4 L_{3}+4 L_{1}-1\right)\left(4 L_{3}+4 L_{1}-2\right)\left(4 L_{3}+4 L_{1}-3\right)\right]-\frac{1}{2} L_{1} \tag{B.21}
\end{align*}
$$

$$
\begin{align*}
N_{2}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{1}{12} L_{2}\left[\left(4 L_{2}-1\right)\left(4 L_{2}-2\right)\left(4 L_{2}-3\right)\right. \\
& +\left(4 L_{2}+4 L_{3}-1\right)\left(4 L_{2}+4 L_{3}-2\right)\left(4 L_{2}+4 L_{3}-3\right) \\
& \left.+\left(4 L_{1}+4 L_{2}-1\right)\left(4 L_{1}+4 L_{2}-2\right)\left(4 L_{1}+4 L_{2}-3\right)\right]-\frac{1}{2} L_{2}  \tag{B.22}\\
N_{3}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{1}{12} L_{3}\left[\left(4 L_{3}-1\right)\left(4 L_{3}-2\right)\left(4 L_{3}-3\right)\right. \\
& +\left(4 L_{3}+4 L_{1}-1\right)\left(4 L_{3}+4 L_{1}-2\right)\left(4 L_{3}+4 L_{1}-3\right) \\
& \left.+\left(4 L_{2}+4 L_{3}-1\right)\left(4 L_{2}+4 L_{3}-2\right)\left(4 L_{2}+4 L_{3}-3\right)\right]-\frac{1}{2} L_{3} \\
N_{4}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{4}{3} L_{1} L_{2}\left[\left(4 L_{3}+4 L_{1}-1\right)\left(4 L_{3}+4 L_{1}-2\right)+\left(4 L_{1}-1\right)\left(4 L_{1}-2\right)\right]  \tag{B.24}\\
N_{5}\left(L_{1}, L_{2}, L_{3}\right)= & 2 L_{1} L_{2}\left[\left(4 L_{2}-1\right)\left(4 L_{3}+4 L_{1}-1\right)+\left(4 L_{1}-1\right)\left(4 L_{2}+4 L_{3}-1\right)\right]  \tag{B.25}\\
N_{6}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{4}{3} L_{1} L_{2}\left[\left(4 L_{2}-1\right)\left(4 L_{2}-2\right)+\left(4 L_{2}+4 L_{3}-1\right)\left(4 L_{2}+4 L_{3}-2\right)\right]  \tag{B.26}\\
N_{7}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{4}{3} L_{2} L_{3}\left[\left(4 L_{1}+4 L_{2}-1\right)\left(4 L_{1}+4 L_{2}-2\right)+\left(4 L_{3}-1\right)\left(4 L_{3}-2\right)\right]  \tag{B.27}\\
N_{8}\left(L_{1}, L_{2}, L_{3}\right)= & 2 L_{2} L_{3}\left[\left(4 L_{3}-1\right)\left(4 L_{1}+4 L_{2}-1\right)+\left(4 L_{2}-1\right)\left(4 L_{3}+4 L_{1}-1\right)\right]  \tag{B.28}\\
N_{9}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{4}{3} L_{2} L_{3}\left[\left(4 L_{3}-1\right)\left(4 L_{3}-2\right)+\left(4 L_{3}+4 L_{1}-1\right)\left(4 L_{3}+4 L_{1}-2\right)\right]  \tag{B.29}\\
N_{10}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{4}{3} L_{1} L_{3}\left[\left(4 L_{2}+4 L_{3}-1\right)\left(4 L_{2}+4 L_{3}-2\right)+\left(4 L_{3}-1\right)\left(4 L_{3}-2\right)\right]  \tag{B.30}\\
N_{11}\left(L_{1}, L_{2}, L_{3}\right)= & 2 L_{1} L_{3}\left[\left(4 L_{1}-1\right)\left(4 L_{2}+4 L_{3}-1\right)+\left(4 L_{3}-1\right)\left(4 L_{1}+4 L_{2}-1\right)\right]  \tag{B.31}\\
N_{12}\left(L_{1}, L_{2}, L_{3}\right)= & \frac{4}{3} L_{1} L_{3}\left[\left(4 L_{1}-1\right)\left(4 L_{1}-2\right)+\left(4 L_{1}+4 L_{2}-1\right)\left(4 L_{1}+4 L_{2}-2\right)\right] \tag{B.32}
\end{align*}
$$

## VITA

Kristen Karman was born in Arlington, TX on August of 1988. She moved to Chattanooga, TN in 2003, where she finished high school in 2006. Kristen earned a Bachelor of Science in Mathematics and Classical Languages from the University of Tennessee at Knoxville in 2011. She continued her education at the University of Tennessee at Chattanooga, earning a Master of Engineering in Computational Engineering in December of 2013 and a Ph.D. in Computational Engineering in December of 2017.

