Discrete Surface Ricci Flow for General Surface Meshing in Computational Electromagnetics using Iterative Adaptive Refinement

Cam Key, *Student Member, IEEE*, Jake Harmon, *Student Member, IEEE*, and Branislav M. Notaroš, *Fellow, IEEE*

***Abstract*—We propose a surface meshing approach for computational electromagnetics (CEM) based on discrete surface Ricci flow (DSRF) with iterative adaptive refinement in the parametric domain for the automated generation of high-quality surface meshes of arbitrary element type, order, and count. Surfaces are conformally mapped by DSRF to a canonical parametric domain, allowing a canonical seed mesh to be mapped back to an approximation of the original surface. The new DSRF-based meshing technique provides a framework for generation of meshes with high element quality, aimed to greatly enhance the accuracy, conditioning properties, stability, robustness, and efficiency of surface integral equation CEM solutions. We demonstrate the ability of the proposed DSRF technique to produce meshes with near-optimal element corner angles for complicated, highly-varied surfaces such as the NASA almond and a fighter jet model, using triangular, quadrilateral, and discontinuous quadrilateral elements. Other element types are also discussed. Where high-fidelity meshing is desired, the technique can capture fine-scale detail using very few high order elements. Where low-fidelity meshing is desired, DSRF with adaptive refinement can accurately recreate course-scale detail using standard first-order elements (e.g., flat triangular patches).**

***Index Terms*—automated surface meshing; computational electromagnetics; Ricci flow; higher-order methods; large-domain modeling; mesh refinement; iterative adaptive refinement; method of moments; surface integral equation techniques; quadrilateral elements; triangular elements.**

I. Introduction

Mesh generation is a critical, yet largely neglected, aspect of research in computational electromagnetics (CEM). Surface discretization quality impacts the numerical solution of electromagnetics problems substantially, yet new surface integral equation (SIE) techniques seem to mostly defer this aspect due to its difficulty. New simulation techniques emerge and problem sizes grow, but a relatively static pool of surface meshing approaches must contend with an ever-increasing variety of surface mesh types, each with unique benefits but added geometric constraints. These constraints define an appropriate mesh quality for a specific application. Here, we use corner angle uniformity and the resolution of important surface features. Many promising MoM-SIE innovations simply cannot rely on existing meshing approaches to produce the required discretizations at any usable quality, limiting the applicability of new research and relegating practitioners to heavily-involved semi-manual meshing. SIE-based CEM methods also increasingly rely on numerical error estimate-based adaptive refinement techniques to efficiently and dynamically modify problem discretizations during computation [1], necessitating the integration of complicated surface meshing algorithms with existing CEM software. As such, meshing approaches tailored toward CEM applications are of growing importance.

Several existing surface mesh generation approaches are available to CEM researchers, but, to our knowledge, none allow for the automatic generation of meshes with user-defined element type, count, and order. Moreover, first-order triangular mesh generation is well understood and often simple due to desirable topological properties of triangles as a 2-simplex. For instance, see [2] for surface triangulation from arbitrary point clouds or [3] for improving existing triangular surface meshes. Triangular meshes can also be generated from arbitrary polygonal meshes by subdivision of any polygon [4] and are ubiquitous in SIE numerical methods, for instance, see [5]. First-order triangular surface meshing in CEM has relied largely on Delaunay triangulation-based meshing approaches due to their simplicity and robustness [6, 7, 8]. Unfortunately, applying the Delaunay triangulation directly limits its applicability strictly to 2D (plate) structures or 2D domains. Prior triangular surface meshing work in CEM has focused on refining and improving an existing triangle mesh using various implementations of node addition with local mesh rearrangement [9, 10, 11]; quad-tri conversion [12]; and iterative refinement beginning from manual vertex labels [13].

Also common in CEM are first-order quadrilateral elements, but less so than first-order triangular, see for instance [14]. First-order quadrilateral mesh generation is more difficult, as approaches typically rely on direct tri-quad conversion [15, 16], patch-based methods [17, 18], Voronoi-based methods [19], or parametrization-based methods [20, 21, 22]. See [23] for a recent overview of the state of the art in quadrilateral mesh generation. Our proposed method, applied to quadrilateral meshing, is parametrization-based but maintains generality to other surface mesh types. Discontinuous quadrilateral meshes, in which adjacent

Manuscript received June 26, 2019, revised December 12, 2019; second revision February 1, 2020; third revision April 3, 2020; accepted June 3, 2020. This work was supported in part by the National Science Foundation under grant ECCS-1810492 and by the US Air Force Research Laboratory, CREATE SENTRi, Riverside Research Institute, under contract FA8650-14-D-1725(6F1957).

Cam Key, Jake Harmon, and Branislav M. Notaroš are with Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373 USA (e-mail: camkey@rams.colostate.edu, j.harmon@colostate.edu, notaros@colostate.edu).

quadrilaterals need not share entire edges, are simple to generate at high quality from existing continuous quadrilateral surface meshes by merging or subdividing chosen quadrilaterals [23]; see [24] for the advantages of discontinuous methods. We therefore first focus on continuous quadrilateral mesh generation but generalize to the discontinuous case by this property. Direct tri-quad conversion typically leads to poor mesh quality for most CEM applications (highly irregular elements with varied corner angles). Voronoi-based approaches can lead to quad-dominant meshes in which many triangles remain, raising issues for SIE solvers unable to handle both quadrilateral and triangular elements simultaneously. Patch-based methods, meanwhile, are not guaranteed to generate a complete mesh and can fail during the generation process [23]. So, they are unreliable for quintessential surfaces in CEM such as vehicles, antennas, or building environments. Parameterization-based approaches are typically the most robust, but little work has been done to optimize them or apply them for CEM applications except for parameter sweeping on 2D surface meshes [25]. Other CEM-specific first-order quad meshing work has typically concerned iterative partitioning by sets of node placement rules [26, 27]. More recently, much first-order quad meshing research in CEM has relied on approaches that, although robust, are limited to mixed quad-tri meshes and therefore not applicable for methods relying on meshes of a single element type [28, 29, 30, 31].

Beyond the well-known first-order techniques, higher-order methods are of growing interest in CEM. Such techniques have shown great promise reducing the system dimension for comparable or higher accuracy in MoM-SIE solvers, thereby reducing the computation time and/or increasing accuracy substantially [32, 33], but the complexity of generating the needed higher-order quadrilateral or triangular meshes has limited the promised applicability for true large-domain modeling. Generating such meshes is often left out of scope [34], semi-manual, or, at best, unable to effectively generate large-domain elements, typically relying on the combination of several existing elements into larger high-order elements [32]. This can only be done on highly-structured meshes, with most others lacking the topology for merging elements to satisfy common geometric interpolation techniques. No robust meshing process has been developed for this application, and existing parametrization-based techniques, optimized for first-order elements, are not well-suited to the task. Prior attempts to produce robust and broadly applicable higher-order quadrilateral and triangular surface meshing techniques have used existing high quality first-order meshes of the desired type and subsequently interpolated them [35, 36]. This approach is excellent where such meshes are available but cannot be used when such meshes are unavailable or difficult to produce at high quality, for instance when large-domain quadrilateral meshes are desired. Meshing, potentially the most challenging and restrictive component of higher order CEM, is a highly relevant open problem.

This paper proposes an efficient and robust surface meshing technique applicable to any of the discussed mesh types and extensible easily to others. Able to seamlessly handle higher order and very high order elements, our technique can surmount the major barrier to widespread use of otherwise highly-efficient higher order methods. Due to its generality, our technique also constitutes a competitive low-order meshing approach useful for first-order triangular (flat triangles) and first-order quadrilateral mesh generation at high quality. By producing meshes with high corner angle uniformity, the technique can maintain high local regularity of surface Jacobians and high basis function orthogonality, preventing degeneration of the basis functions and associated increases in condition number. A parametrization-based approach, our technique leverages the discrete surface Ricci flow (DSRF) to map between an arbitrary triangulated mesh and an appropriate canonical parametric domain dependent on the underlying geometry of the original surface and desired mesh properties. A uniform seed mesh, known in the parametric domain, is then taken by this mapping to the original surface, on which a refinement indicator, defined for the choice of mesh type, is computed. The seed mesh is then refined in the parametric domain based on this indicator and the process is repeated until a stop criterion is met, for instance maximum element size or element count. Although we begin here with a triangular mesh of the original surface, in general we can begin with an arbitrary surface representation, which must be converted to a high-quality triangulation by sampling, subdivision, or surface reconstruction as appropriate to the type of surface representation, for instance using techniques from [2] or [4].

We believe our approach to be the first demonstration of a surface meshing technique able to seamlessly handle arbitrary (low and high) geometric order and element type. Furthermore, this appears to be the first technique able to generate high-quality very high order quadrilateral elements; the lack of such a technique previously constituting the main shortcoming of large-domain methods [32, 34]. We demonstrate our technique for a variety of common mesh types used for low-order and higher order MoM-SIE methods including triangular, quadrilateral, and discontinuous quadrilateral, and offer suggestions for simple extension to other, less common element types. For each of these types, we maintain generality in element order and show typical results for first-order (lowest-order), e.g., flat triangular patches, as well as higher-order elements. For the latter, we demonstrate meshes using both Lagrange interpolation and cubic spline interpolation, again offering suggestions for extension to other interpolation methods. We focus here on the application of this technique to MoM-SIE in CEM but do not limit its usefulness to only this application. Some preliminaries of this work are presented in a summary form in [37, 38, 39].

We note a few limitations of the proposed method. Firstly, the method as presented does not enforce continuity for multi-part objects. To work for such cases in practice, the method therefore requires either a discontinuous Galerkin solver or special treatment at part interfaces to enforce continuity. The adaptive refinement we present here also does not guarantee perfect sampling of the original surface on non-differentiable features (i.e., sharp edges). The error due to imperfect sampling of sharp features drops asymptotically to zero with increased iteration but may be unacceptable where perfect preservation of sharp features at otherwise low mesh fidelity is required. Although all refinement methods we present scale as $O(NlogN)$ with the number of iterations, the time complexity of the DSRF with the number of triangles remains formally unknown and is likely the asymptotically dominant factor. We are aware of no study of DSRF scaling with respect to the triangle count, and we consider such a derivation involved enough to warrant its own, separate study. With respect to multiple parameters, DSRF scales formally as $O(N_t N_m N_n)$, where N_t is the number of triangles in the original mesh, N_m is the average number of iterations required to solve the Hessian system to a chosen tolerance, and N_n is the number of Newton method iterations required to meet a tolerance on the curvature error. The open problem is relating N_n and N_m to N_t. We hope to address these limitations and perform a rigorous study of DSRF scaling in future work.

We begin by summarizing our method and recapitulating the mathematics of DSRF, tailored specifically to be understandable and useful to the CEM community. We encourage readers interested in a more-formal, in-depth theoretical discussion with additional implementation details to review [40]. We use the same notation where possible. We then describe iterative adaptive refinement and offer specific implementations for common mesh types. Then, we present a variety of meshes produced by the method as well as statistics on mesh corner angle uniformity, a common measure of quality. For the examples given in the present manuscript, we focus on the common, important case where our surface is simply connected, closed, and has one symmetry plane, allowing us to reduce to the Euclidean case by cutting the surface along the plane of symmetry. This simplifies both computation and presentation of the DSRF. In future work, we plan to similarly demonstrate the method for surfaces requiring spherical or hyperbolic DSRF. We apply the method to the NASA almond, an established CEM benchmarking shape usually used to demonstrate difficulty of surface modeling given its one sharp end. We also show meshing results for a far more-complicated, and therefore challenging, fighter-jet model. For each of these models, we show continuous triangular, continuous quadrilateral, and discontinuous quadrilateral surface meshes of both low and high geometric order. We conclude with a discussion of the potential of the new DSRF meshing technique with adaptive refinement.

II. Summary of the Method and Computing the DSRF

Popularized by its role in Perelman's 2006 proof of the Poincaré conjecture [41], Ricci flow offers a mathematical framework for diffusing irregularities in the metric of a Riemannian manifold. In the context of this work, surface Ricci flow, by the discrete formulation described in [42], allows the generation of a conformal (angle-preserving) mapping between a surface of choice, and a homeomorphic (or non-homeomorphic, given a suitable cut graph) surface of prescribed Gaussian curvature, here constituting a parametric domain for the mesh and referred to as the prescribed surface. For instance, this allows the NASA almond to be mapped to the unit sphere or, as we demonstrate in this paper, cut and mapped to the plane. Information on the prescribed surface can then be conformally mapped back to the original surface. In our application, this information comprises element vertices, and in our higher order cases, element sample points.

Since much of the material covered in this paper is likely unfamiliar to CEM audiences, we begin with a high-level summary of the method before presenting the theory in more detail. As a concrete example, we demonstrate each step for a simple ellipsoid cut along a plane of symmetry for which we want to produce a structured quadrilateral mesh. The basic steps of the method are as follows.

1. If it is not already, convert the surface into a triangle mesh, shown in Fig. 1(a) for the cut ellipsoid
2. Obtain a mapping between the original surface and a simple prescribed surface by DSRF. The cut ellipsoid triangle mesh mapped to a rectangular prescribed surface is shown in Fig. 1(b).
    a. Choose a simple surface in which we can easily manipulate mesh topology (e.g. a flat rectangle)
    b. Assign a target curvature to each point in the original triangle mesh consistent with the prescribed surface
    c. Perform the DSRF to compute locations of vertices from the triangle mesh when flattened to the prescribed surface
3. Apply the mapping to resample the original surface adaptively, manipulating mesh topology in the parametric domain. A uniform quadrilateral sampling is shown overlaid on Fig. 1(b) and an adaptive sampling is shown overlaid in Fig. 1(c).
    a. Define a simple seed mesh that covers the prescribed surface
    b. Define a refinement indicator and refinement method appropriate for the target mesh type and quality measure
    c. Compute the refinement indicator for the seed mesh
    d. Refine the seed mesh using the refinement method, based on computed values of the refinement indicator
    e. Iterate steps d and e until some stop criterion is met (number of steps, maximum element size threshold, etc.)

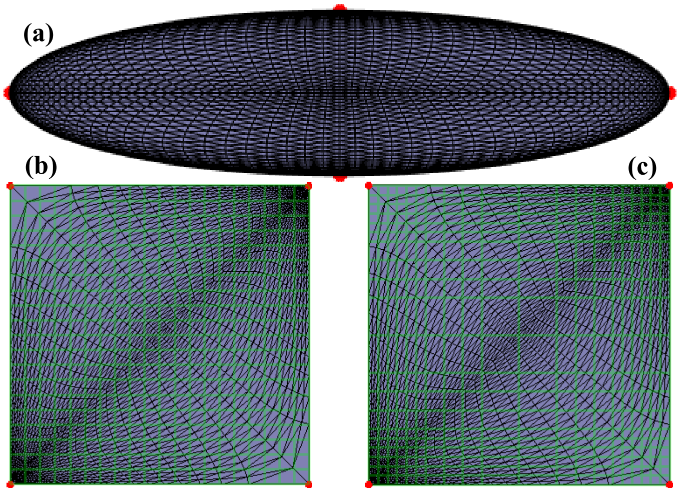

Fig. 1. Original triangle mesh for a cut ellipsoid (a), triangle mesh in the parametric domain with uniform structured quadrilateral sampling (b), and adaptive structured quadrilateral sampling (c) overlaid. Vertices in red are assigned $\pi/2$ curvature. All other vertices are assigned zero curvature.

Step 1 is simple for almost all surface descriptions, so we consider it outside the scope of this paper.

Step 2 is described in the remainder of this section in detail, but we offer key considerations here: The surface of prescribed curvature can be arbitrary but should be a surface on which it is simple to manipulate mesh topology. We use the simple and broadly-applicable example of a flat Euclidean rectangle for this paper.

Step 3 is described in section III in detail, but to summarize: The mapping produced by DSRF does not preserve relative area. This can cause details from the original surface to be missed when a uniform sampling of the parametric domain is used. The goal of adaptive refinement (AR) is to distribute mesh sample points to mitigate this.

Figure 2 compares quadrilateral surface meshes obtained for the example ellipsoid using a nonconformal versus conformal mapping. Figure 3, meanwhile, shows angle histograms for the three meshes in Fig. 1, demonstrating the desirability of a conformal approach.

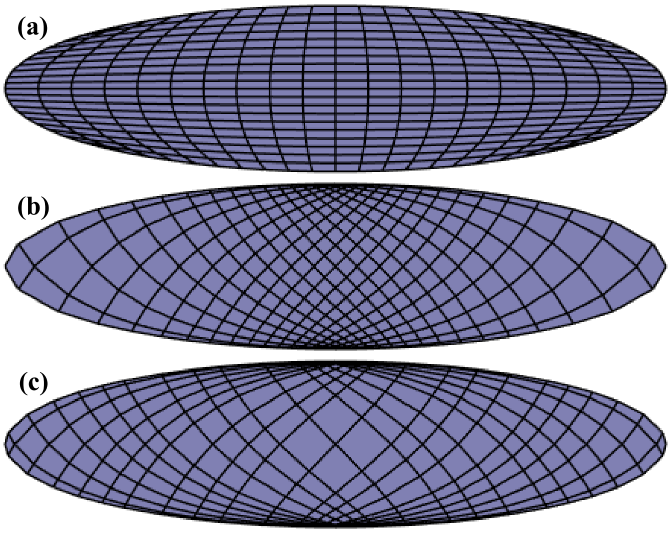

Fig. 2. 20×20 structured quadrilateral meshes mapped to an ellipsoid using (a) a nonconformal mapping; (b) a conformal mapping by DSRF, parametric domain sampling shown Fig. 1(b); and (c) a conformal mapping using DSRF with adaptive refinement, parametric domain sampling shown Fig. 1(c), to capture more detail in regions of high curvature.

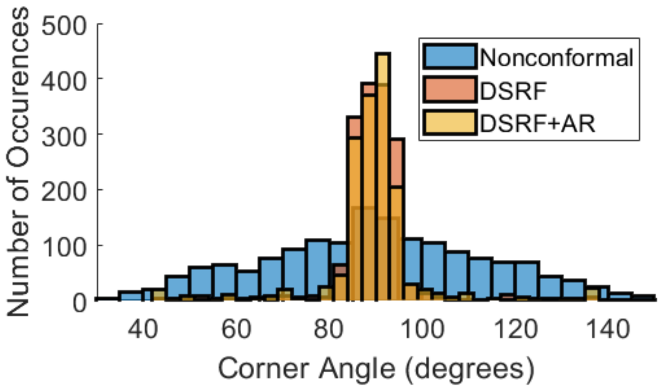

Fig. 3. Corner angle histograms for 20×20 structured quadrilateral meshes show in Fig 2. Nonconformal mesh has a poor corner angle distribution while meshes obtained using DSRF have distributions concentrated closely around 90 degrees.

The result of Step 1, we denote our original triangle mesh $\Sigma = (V, E, F)$ where V, E, and F are the sets of vertices, edges, and faces composing the mesh, respectively. Here we assume our mesh represents the boundary of a realizable three-dimensional (3D) object, i.e., the surface does not intersect itself, is continuous, and is finite. We refer to this as the initial surface. The initial surface may either be closed or have a boundary (a one-dimensional curve in 3D space) $\partial\Sigma$, as in our cut example case. By Step 2, we wish to deform this original surface to a much simpler, prescribed surface on which we can easily define and manipulate mesh topology. We denote the mapping of this surface to the parametric domain $\bar{\Sigma} = (\bar{V}, \bar{E}, \bar{F})$, and an associated map $M\colon \bar{\Sigma} \to \Sigma$ from the parametric domain to the initial surface. The discrete Gaussian curvature of a surface is given by

$$K(v) = \begin{cases} 2\pi - \sum_{jk} \theta_i^{jk} \ , & v \notin \partial\Sigma \\ \pi - \sum_{jk} \theta_i^{jk} \ , & v \in \partial\Sigma \end{cases}, \qquad (1)$$

where v refers to a given vertex and $\sum_{jk} \theta_i^{jk}$ denotes the sum of all triangle corner angles of which v is a part. Here i denotes the index of v, and θ_i^{jk} denotes the corner angle formed by vertex v and two adjacent (connected by an edge) vertices with indices j and k. We define the Euler characteristic as,

$$X = \mathrm{N_V} - \mathrm{N_E} + \mathrm{N_F}\,. \qquad (2)$$

Here $\mathrm{N_V}$, $\mathrm{N_E}$, and $\mathrm{N_F}$ represent the number of vertices, edges, and faces of the surface, respectively. With this, the Gauss-Bonnet theorem asserts,

$$\sum_V K(v) + \epsilon \mathrm{A} = 2\pi \mathrm{X}, \qquad (3)$$

with A denoting the total surface area of the mesh and the scheme coefficient term ϵ determined by the chosen background geometry, taking a value of 0 for the Euclidean case. X takes a value of 1 for the case of a simply-connected half surface. For examples given in this paper, we consider only the Euclidean background geometry and simply-connected half surface, but plan to give more-complicated examples for spherical and hyperbolic background geometry in future work.

Once a prescribed surface is selected, a target curvature $\bar{K}(v)$ is chosen constrained by (2). For instance, using Euclidean background geometry on an open surface (produced by cutting a closed surface along its symmetry plane) and mapping to the Euclidean plane, the total curvature of 2π can be allocated to four boundary vertices (highlighted red in Fig. 1), assigning each curvature $\pi/2$. These vertices will then become the vertices of a rectangular image of the original surface in the Euclidean plane after mapping, shown Fig. 1(b-c). Here, we chose these four vertices to be evenly-spaced on the boundary of the cut surface. By assigning zero curvature to the non-boundary vertices, we enforce that they fall on a flat plane shared by neighboring vertices. Similarly, by assigning all other boundary vertices zero curvature, we enforce that they fall on a straight line shared by neighboring boundary vertices. Once the target curvature is selected, the discrete surface Ricci flow semi-discrete system

$$\frac{du_i(t)}{dt} = \bar{K}_i - K_i(t) \tag{4}$$

can be solved by any number of standard numerical methods to obtain the final conformal mapping between surfaces. The above semi-discrete form retains continuous time, t, which in practice is discretized into a finite set of iterations. A nonlinear equation, (4) must be solved iteratively. For a thorough background on computing the Hessian of the Ricci energy for this system and applying it through Newton's method to obtain the final mapping, see [40]. We give a brief overview here.

To compute and solve the DSRF system, we must define several discrete parameters and structures over Σ. We first define a circle packing metric. We associate with each vertex $v_i \in V$ nonnegative radius γ_i corresponding to a circle centered on v_i. We also define a real-valued discrete conformal structure coefficient on E denoted η. Together with the scheme coefficient from (3), our circle packing metric is then defined by the tuple $(\Sigma, \gamma, \eta, \epsilon)$, from which we can then determine any edge length $l_{ij} \in E$ between vertices $v_i, v_j \in V$. Defining the discrete conformal factor u_i for the Euclidean case,

$$u_i = \log\gamma_i, \tag{5}$$

we can compute l_{ij} by

$$l_{ij}^2 = 2\eta_{ij}\mathrm{e}^{u_i+u_j} + \varepsilon_i\mathrm{e}^{2u_i} + \varepsilon_j\mathrm{e}^{2u_j} \tag{6}$$

The ε coefficients and range of the conformal structure coefficient for several common circle packing schemes [40] are defined in Table I. We use inversive distance circle packing for the results presented in this paper.

Table I. Range of conformal structure coefficient and ε coefficient values for common circle packing schemes.

Scheme	η_{ij}	ε_i	ε_j
Thurston's	[0,1]	+1	+1
Tangential	+1	+1	+1
Virtual radius	> 0	-1	-1
Inversive Distance	> 0	+1	+1

With a circle packing scheme defined and chosen, we can now solve the DSRF system (4) iteratively as follows. At every iteration we begin by computing all circle radii γ_i from the discrete conformal factor (5). Following this, we use η and γ values to compute all edge lengths by application of (6). From the edge lengths, we compute the corner angles θ_i^{jk} from the cosine law appropriate to Euclidean background geometry,

$$l_k^2 = \gamma_i^2 + \gamma_j^2 - 2l_il_j\cos\theta_k, \tag{7}$$

and subsequently the vertex curvature K from the angle deficit (1). We then compute the Hessian matrix, H, from the local (face) Hessian matrices:

$$\frac{\partial(\theta_i,\theta_j,\theta_k)}{\partial(u_i,u_j,u_k)} = -\frac{1}{2A}L\Theta L^{-1}D \tag{8}$$

where

$$L = \begin{bmatrix} s(l_i) & 0 & 0 \\ 0 & s(l_j) & 0 \\ 0 & 0 & s(l_k) \end{bmatrix} \tag{9}$$

$$A = \sin\theta_i s(l_j)s(l_k) \tag{10}$$

and

$$D = \begin{bmatrix} 0 & \tau(i,j,k) & \tau(i,k,j) \\ \tau(j,i,k) & 0 & \tau(j,k,i) \\ \tau(k,i,j) & \tau(k,j,i) & 0 \end{bmatrix} \tag{11}$$

Although we maintain consistency with [40] here, for the Euclidean case we have simply that

$$s(x) = x \tag{12}$$

and

$$\tau(i,j,k) = \frac{l_i^2+\epsilon_j\gamma_j^2+\epsilon_k\gamma_k^2}{2} \tag{13}$$

Finally, we solve the linear system

$$H\delta u = \bar{K} - K \tag{14}$$

for δu, updating the discrete conformal factor to be used in the next iteration as

$$u \leftarrow u - \delta t\delta u. \tag{15}$$

This process is iterated until a convergence criterion is met, most simply until the maximum difference between the current and target discrete curvature falls below some threshold, i.e., until

$$\max_i|\bar{K}_i - K_i| < threshold \tag{16}$$

From the final η, γ, u, and θ values, we can compute the final vertex locations in the target domain by flattening from a seed face as in [42]. For additional discussion of convergence rate, stability, and modifications to improve the robustness of the above approach, see [40, 43, 44].

Once the locations of all vertices are known in the parametric domain, any point within that domain can be mapped back to the initial surface using barycentric coordinates, defining a piecewise-linear approximation of *M*. A point *p* with parametric coordinate (u_0, w_0) in the parametric domain is found to lie in face *f*. If *f* has parametric vertices v_1, v_2, and v_3, each with parametric coordinate of form (u_i, w_i) and nonparametric coordinate of form (x_i, y_i, z_i), the image of *p* on the original surface, here denoted p' with coordinate (x_0, y_0, z_0), is given by

$$(x_0, y_0, z_0) = \sum_{i=1}^{3} k_i (x_i, y_i, z_i), \tag{17}$$

where addition is understood component-wise and the Barycentric coordinates are given by

$$k_1 = \frac{\begin{vmatrix} u_3-u_0 & v_3-v_0 \\ u_2-u_3 & v_2-v_3 \end{vmatrix}}{s} \tag{18a}$$

$$k_2 = \frac{\begin{vmatrix} u_1-u_0 & v_1-v_0 \\ u_3-u_1 & v_3-v_1 \end{vmatrix}}{s} \tag{18b}$$

$$k_3 = \frac{\begin{vmatrix} u_2-u_0 & v_2-v_0 \\ u_1-u_2 & v_1-v_2 \end{vmatrix}}{s} \tag{18c}$$

with scaling factor

$$s = \begin{vmatrix} u_1 - u_2 & v_1 - v_2 \\ u_3 - u_1 & v_3 - v_1 \end{vmatrix} \tag{19}$$

III. Iterative Adaptive Refinement

By choosing a simple prescribed surface, we exert a high degree of control over the resultant re-mapping. However, the mapping produced by Ricci flow preserves only angles, not relative areas, so simply mapping a uniform grid of sample points from the prescribed surface to the surface of choice produces poor results for our application, leading to wide discrepancies in mesh fidelity between minimally-warped and highly-warped portions of the resulting surface mesh, as demonstrated in Fig. 2. Figure 4 motivates this from another perspective for the more-complicated cut fighter jet mesh. The parent jet mesh contained 115,967 triangles, and the DSRF took 14.3 seconds to compute on an i7 3770k at 3.50 GHz with a fully parallelized implementation. Fig. 4(a) shows the original triangle mesh, while Figs. 4(a) and (b) show the mesh mapped to the parametric domain and the degree of area warping, respectively. The induced area warping is highly concentrated and irregular.

This is the motivation for beginning with an initial seed mesh and iteratively refining, an approach that allows the unknown degree of local warping to be compensated for adaptively.

We describe here how to construct a mesh informed by some refinement indicator, in general motivated by either geometric error or numerical solution error estimates. Focusing on the geometric properties of the method, we offer specific examples of refinement indicators to reduce geometric error but maintain generality for easy application of the method to adaptive refinement (AR) based instead on solution error.

Given $M: \bar{\Sigma} \rightarrow \Sigma$, we wish to construct a new surface mesh, $\tilde{\Sigma} = (\tilde{V}, \tilde{E}, \tilde{F})$, of arbitrary type. Beginning with a seed mesh $\tilde{\Sigma}_0$ of the chosen mesh type in the parametric domain, we must define a refinement indicator, $R_n(\tilde{\Sigma}, M)$ and a refinement method $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$. We may then iterate Q on $\tilde{\Sigma}_0$ N times, updating R_n at each iteration, to produce a final surface mesh $\tilde{\Sigma} = \tilde{\Sigma}_N$. N may be user-defined or may be a function of $\tilde{\Sigma}$, e.g., some stop criterion like total element count or maximum element size. We give examples of such R_n and Q for a variety of common mesh types and offer suggestions to extend these to other mesh types. We also define the simplest seed mesh for each mesh type covered if Euclidean DSRF is used to map to a rectangle. Note that the refinement methods described here have linearithmic time complexity with the number of refinement iterations.

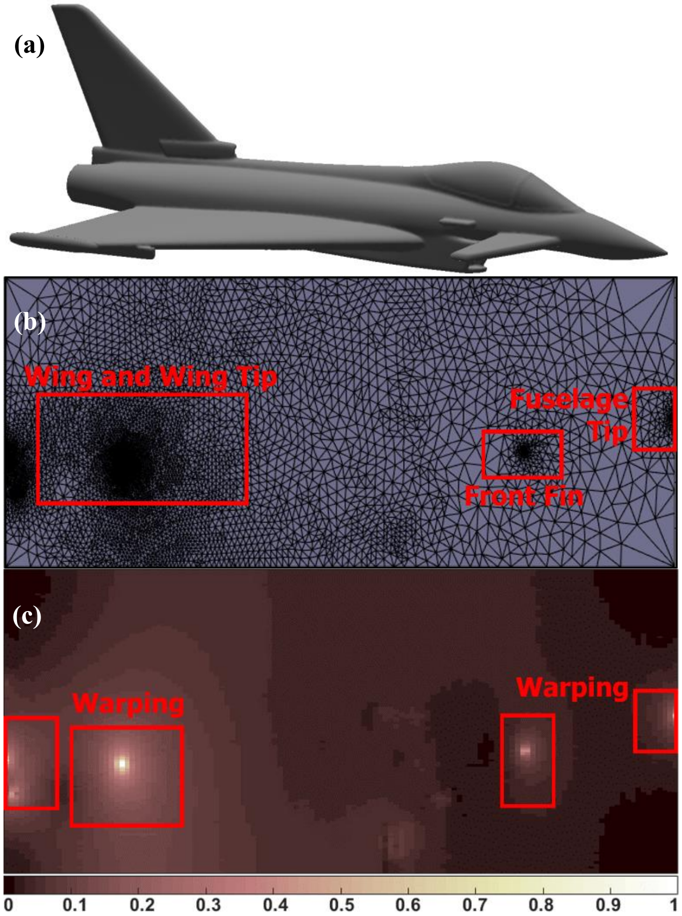

Fig. 4. (a) Initial fighter jet surface. (b) Triangle mesh of fighter jet cut and conformally flattened to the plane with highly warped areas boxed in red. (c) Refinement indicator (normalized) for each quadrilateral element demonstrating high degree of warping at fighter jet fin, wing, and fuselage tip when uniform sampling is used.

A. Continuous Structured Quadrilateral Meshes of Arbitrary Order

Here we give an example of a refinement indicator and associated refinement method for the continuous quadrilateral case using Euclidean DSRF. For every edge $e \in \tilde{E}$, we find the Euclidean distance, d, between its endpoints, (x_1, y_1, z_1) and (x_2, y_2, z_2) in the nonparametric domain

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{20}$$

We assign such a distance to each $e \in \tilde{E}$, constituting $R_n(\tilde{\Sigma}, M)$ with domain $\tilde{E}$.

Starting from a seed mesh in the parametric domain consisting of one quadrilateral element aligned with the parametric coordinate axes as in Fig. 5(a), we find the row and column containing the edge with highest d for the vertical ($w_1 = w_2$) and horizontal ($u_1 = u_2$) edges, respectively. We then subdivide the appropriate row and column in half in the parametric domain, taking one row to two rows and one column to two columns. This can be repeated N times and constitutes one possible $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$. Figures 5(b)-(c) show this subdivision process in the parametric domain for the fighter jet mesh from Fig. 4(a) for N = 2, N = 10, and N = 20 iterations, respectively.

Note that the example Q given here constitutes only a simple and informative refinement method to produce a structured quadrilateral mesh. If unstructured meshes are permissible, any existing quadrilateral mesh refinement method could be used in the parametric domain, with the resulting mesh then mapped back to the original surface conformally. For instance, node-placement schemes like those in [26], [27], and [45] could be adapted to serve as the refinement method.

If higher order elements are chosen, we may subsample $\tilde{\Sigma}_N$ in the parametric domain to produce the necessary interpolation nodes. For instance, if high order elements requiring a grid of k×k nodes per element are chosen, we may split each row and column of $\tilde{\Sigma}_N$ k-2 times to obtain the necessary sample density. In this paper, to improve accuracy and maintain adaptivity for the given examples, we do this implicitly. If an L×L grid of higher order elements, each requiring k×k nodes, is chosen, we define N to be L(k-1)-1 to obtain the necessary sample points for all elements. For the given examples, we order higher order sample points for quadrilateral elements as defined in [32].

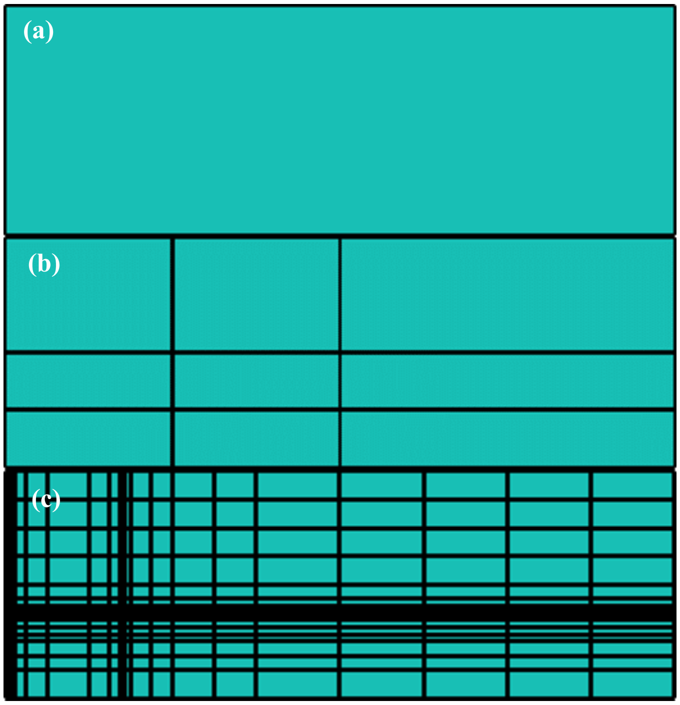

Fig. 5. Iterative adaptive refinement in the parametric domain for a continuous curved quadrilateral mesh of a fighter jet in Fig. 4(a) to increase mesh quality intelligently, with more elements being allocated to high density areas in the parametric domain, leading to a more-uniform final jet mesh: (a) seed mesh with one element aligned with coordinate axes in parametric domain, (b) refined mesh with N = 2 iterations, (c) refined mesh with N = 20 iterations

B. Continuous Triangular Meshes of Arbitrary Order

We now give a similar example for the continuous triangular case using Euclidean DSRF. For every edge $e \in \tilde{E}$, we again find the Euclidean distance, d, between its endpoints in the nonparametric domain (20). We assign such a distance to each $e \in \tilde{E}$, again constituting $R_n(\tilde{\Sigma}, M)$ with domain $\tilde{E}$. For $e \in \tilde{E}$ with maximum d and parametric endpoints (u_1, w_1) and (u_2, w_2), we compute the parametric midpoint p as

$$p = \left(\frac{u_1+u_2}{2}, \frac{v_1+v_2}{2}\right), \tag{21}$$

and include it in the set of existing vertices in $\tilde{\Sigma}$. We then update a Delaunay tessellation of this augmented $\tilde{V}$ in the parametric domain to update $\tilde{E}$ and $\tilde{F}$ to include the added vertex. This can be repeated N times and constitutes a possible $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$. Figures 6(a)-(c) show similar parametric-domain adaptive refinement results for N = 2, 50, and 100 iterations, respectively.

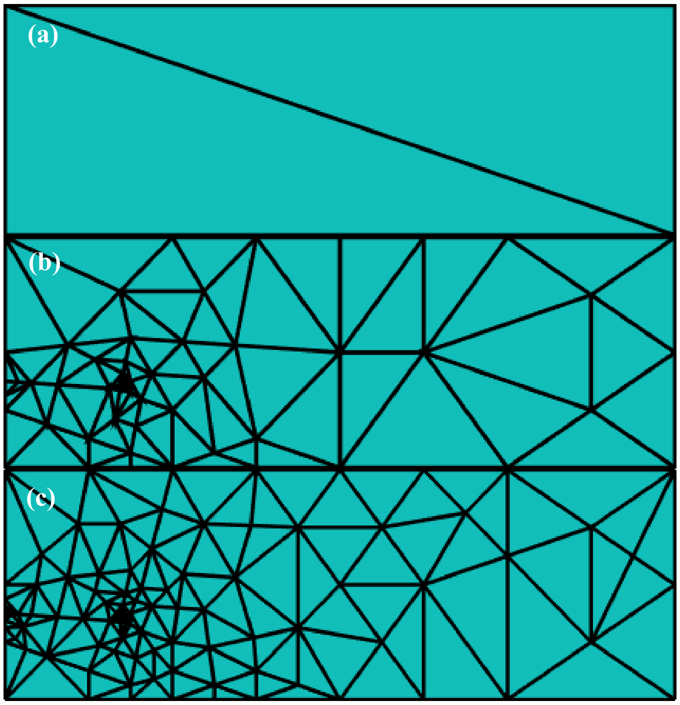

Fig. 6. Iterative adaptive refinement in the parametric domain for a continuous curved triangular mesh of a fighter jet in Fig. 4(a): (a) seed mesh with two elements in the parametric domain, (b) refined mesh with N = 50 elements, and (c) refined mesh with N = 100 iterations.

As in the continuous quadrilateral case, extension to higher order elements is simple, requiring only additional sampling of the mapping at interpolation nodes. For the examples given in this paper, we define higher order sample points for each triangle as in [32].

C. Discontinuous Quadrilateral Meshes of Arbitrary Order

To define a suitable refinement indicator in the discontinuous quadrilateral case using Euclidean DSRF, we again compute (20) for all $e \in \tilde{E}$, beginning from the seed mesh defined for the continuous quadrilateral case and shown in Fig. 5(a). For e with maximum d, we split an adjacent face in the direction perpendicular to such e, introducing two new vertices and one new edge. Note that, although an edge may have two adjacent faces, it is in practice inconsequential which face is split on a given iteration, as the unsplit face is guaranteed to be refined on a subsequent (typically the next) iteration, dependent on the number of edges with identical maximum d. Such face splitting constitutes a possible $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$ and introduces one new face per iteration. Additional constraints could be imposed on which element to split at each iteration to satisfy potential requirements of specific discontinuous quadrilateral codes. For instance, if an implementation requires that one edge joins to at most two, conditions that would violate this if refined could be detected at each iteration, with e with the next highest d chosen instead. Figure 7 demonstrates this refinement method for various N on the jet fighter mesh with no such additional constraints. For extension to higher order, we sample on a quadrilateral-by-quadrilateral basis and again use the sampling convention defined for higher order quadrilaterals in [32].

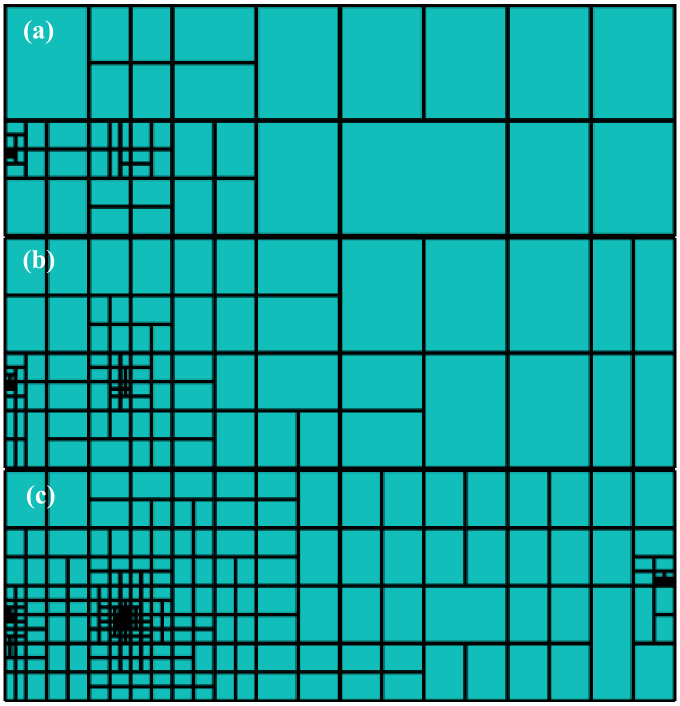

Fig. 7. Adaptive refinement for discontinuous quadrilaterals in the parametric domain: (a) $N = 50$, (b) $N = 100$, and (c) $N = 300$ iterations.

D. Continuous Unstructured Quadrilateral Meshes of Arbitrary Order

Here we demonstrate how DSRF-AR can be used to convert an existing 2D meshing technique into a 3D surface meshing technique, in this case for continuous structured quadrilateral meshes. We begin with the 2D continuous quadrilateral subdivision method described in [45]. The method in [45] first refines elements uniformly by splitting each refined element into a 3×3 grid of quadrilaterals. A set of 4 irregular subdivision patterns is then applied to adjacent elements to repair any discontinuities introduced during refinement. To apply this method to 3D surfaces using our DSRF-AR approach, we again compute (20) for all $e \in \tilde{E}$. For e with maximum d, we split an adjacent face into a 3×3 grid of quadrilaterals. Any neighboring faces of the refined face are also refined uniformly if needed to maintain the criterion that no edge joins to more than 3. This is iterated until some stop criterion is met, after which the irregular subdivision templates from [45] are applied to repair all discontinuities. Note that no irregular element is ever subdivided, as this would lead to unbounded mesh quality deterioration. This constitutes another possible $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$. Figure 8 demonstrates this refinement method on the fighter jet mesh for several N. As in previous quadrilateral examples, extension to higher order constitutes quadrilateral-by-quadrilateral resampling using the convention defined in [32]. Note that, since [45] assumes square elements to maintain reasonable corner angles in the irregular subdivision templates, we use a different seed mesh here, splitting the rectangular parametric domain into approximately square elements (here 3). This can be automated by comparing the width and height of the rectangular parametric domain, subdividing it appropriately.

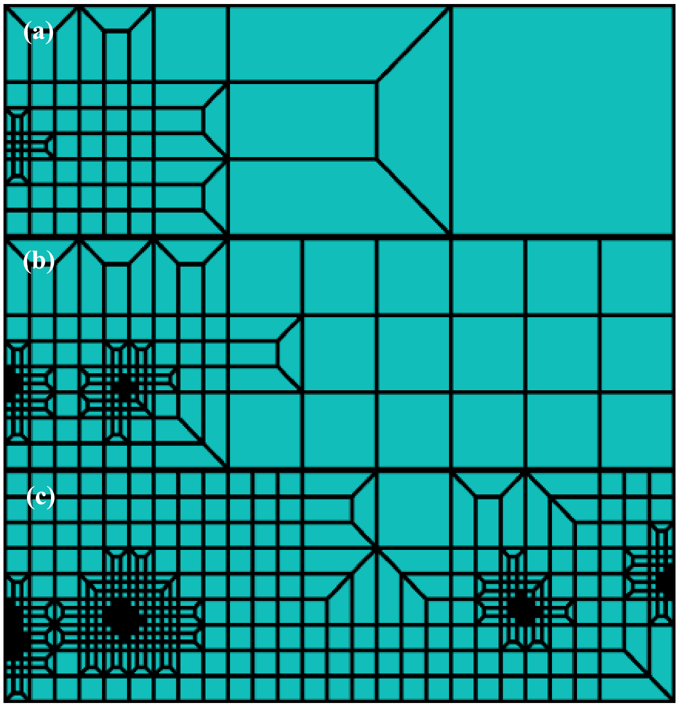

Fig. 8. Adaptive refinement for continuous unstructured quadrilaterals in the parametric domain: (a) $N = 5$ (b) $N = 10$, and (c) $N = 50$ iterations.

E. Generalization to Mesh Types Not Covered

Although we have covered three common mesh types, we by no means wish to limit the applicability of DSRF with adaptive refinement to production of only continuous quadrilateral, continuous triangular, and discontinuous quadrilateral meshes. We hope the given examples offer clear guidance for generalizing to other mesh types, but we additionally give recommendations for generalization by offering the following guidance: it is crucial to choose $R_n(\tilde{\Sigma}, M)$ and especially $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$. to prevent the creation of malformed elements during refinement in the parametric domain. “Malformed” depends on the mesh type and user application, but typically is related to the regularity of corner angles and local Jacobian within and between elements. As $M(p) = p'$ is conformal, malformed elements in the parametric domain become malformed elements in the final surface mesh.

IV. Results and Discussion

A. Example Surface Meshes Produced by DSRF with Adaptive Refinement

Here we demonstrate meshing by DSRF with adaptive refinement for the well-known NASA almond model and the complicated fighter jet model. Higher order elements were reconstructed using Lagrange interpolation as in [32] or cubic spline interpolation of the sample points on an element-by-element basis. The same initial Σ used for all cases (one for the almond, one for the fighter jet). The DSRF was computed on these initial meshes and used to generate one M for each model. The parametric domain adaptive refinement methods described in Section III were then iterated through these mappings to produce a variety of surface meshes shown in Figs. 9-20. We show the robustness of the proposed technique to recreate complicated surfaces for arbitrary mesh types with arbitrary element counts and orders, for instance accurately representing the fighter jet model with as few as 32 elements. We are not aware of any other meshing technique that can reliably produce such large-domain meshes.

Figure 9(a) shows a high-resolution higher order continuous quadrilateral surface mesh generated using DSRF with the refinement scheme outlined in Section III.A. Figure 9(b) shows the equivalent surface mesh instead using uniform sampling in the parametric domain. All parameters including element count, element order, Σ, and M were identical between Figs. 9(a) and 9(b). Spline interpolation was chosen in both cases. Extreme loss of fidelity can be seen around the fuselage tip and wing tips in the uniformly-sampled case, these details meanwhile excellently captured in the adaptive case. This shows not only the importance but also the effectiveness of the proposed adaptive sampling methods for accurately capturing detail in the desired model.

Figure 10 shows the same mesh as Fig. 9(a) from an oblique angle, making the high fidelity with which the adaptive sampling technique captures fine detail in the initial surface apparent. A comparison between this higher order continuous quadrilateral mesh and the 1st-order triangular mesh (chosen for Σ and shown in Fig. 4(a)) shows the near perfection with which this instance of $\tilde{\Sigma}$ recreates the original surface. A similar result is shown in Fig. 11 for the NASA almond, here using 32 16th-order continuous quadrilateral elements with Lagrange interpolation. The parent almond mesh contained 2,023 triangles and the DSRF took 0.168 seconds to compute. Adaptive sampling was also used for Fig. 11 as outlined in Section III.A. Note that, for most practical use cases, such large, curved elements would be supported by extremely high-order current expansions, most importantly to compensate for their large electrical size.

Figures 12 and 13 show 1st-order discontinuous quadrilateral meshes generated using the technique outlined in Section III.C. Figure 12 shows the fighter jet model recreated using 6490 1st-order elements, while Fig. 13 shows the NASA almond featuring 2000 elements of the same type. In both cases, the original surface is well-reconstructed.

Figures 14 and 15 show higher-order analogues of Figs. 12 and 13, now using 300 30th-order and 300 10th-order

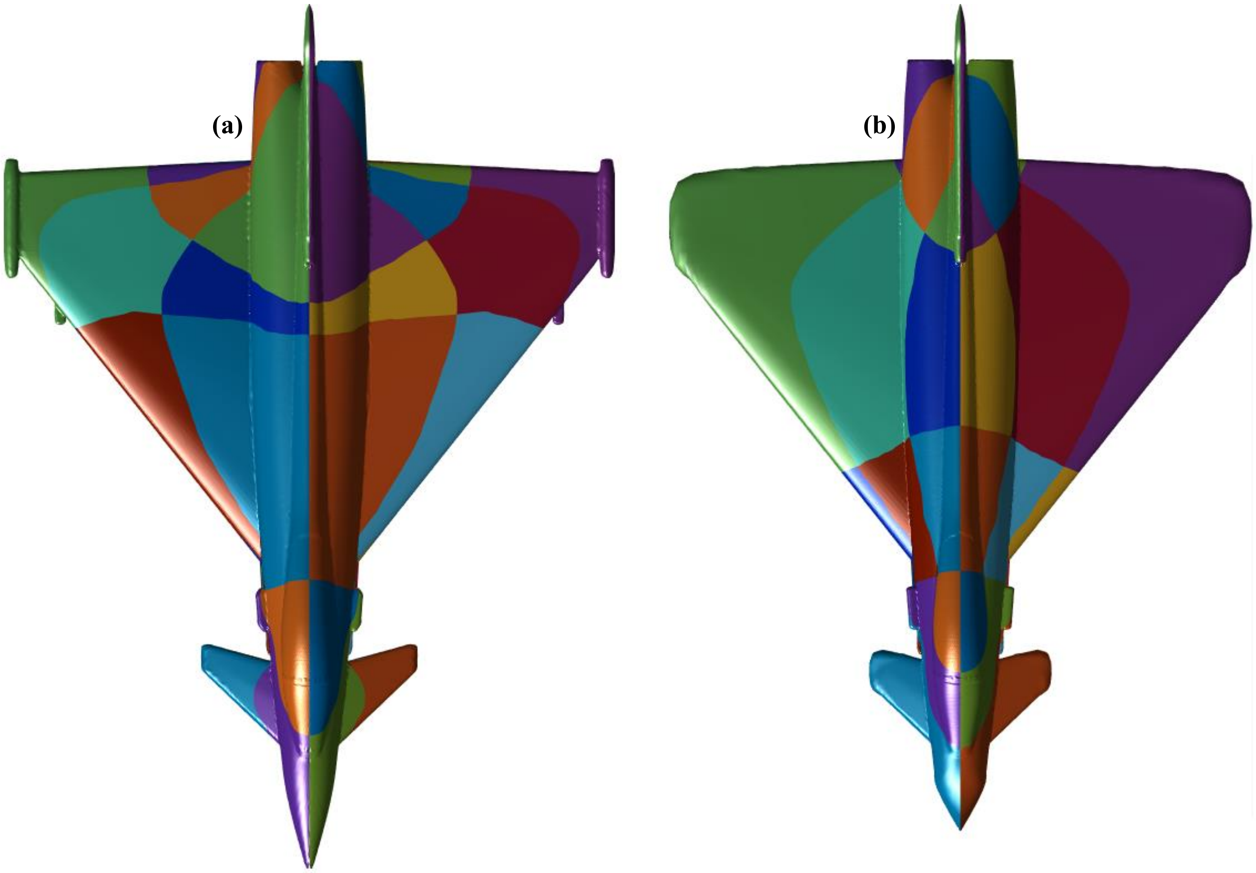

Fig. 9. Comparing effects of adaptive iterative refinement vs. uniform sampling on mesh quality: (a) continuous quadrilateral mesh of a fighter jet in Fig. 4(a) with 32 64th-order elements using DSRF with iterative adaptive refinement outlined in Fig. 5 and (b) the same using uniform sampling.

discontinuous quadrilateral elements, respectively. Spline interpolation was used in both cases. Note that roughness present on the almond surface in Fig. 15 is not an artifact of the proposed meshing technique, but rather shows an accurate recreation of roughness due to 1st-order triangular facets in the original almond mesh chosen as Σ.

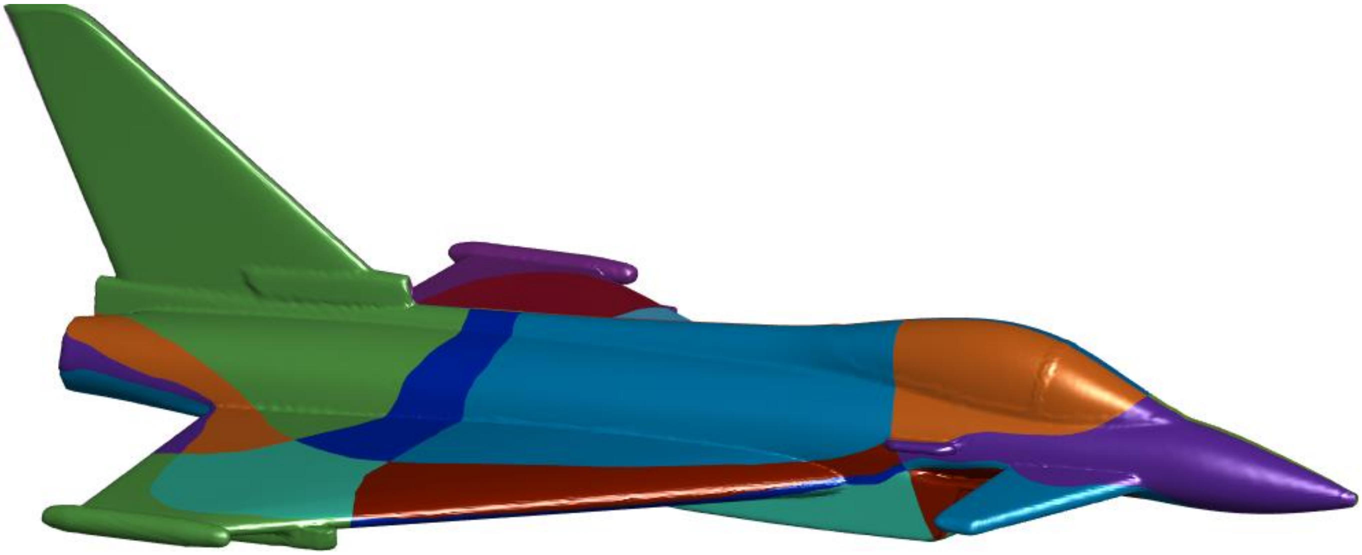

Fig. 10. Fighter jet model featuring as few as 32 64th-order quadrilateral elements in Fig. 9(a) viewed from oblique angle. Note excellent curvature/detail modeling with hyper-large hyper-curved quadrilateral patches.

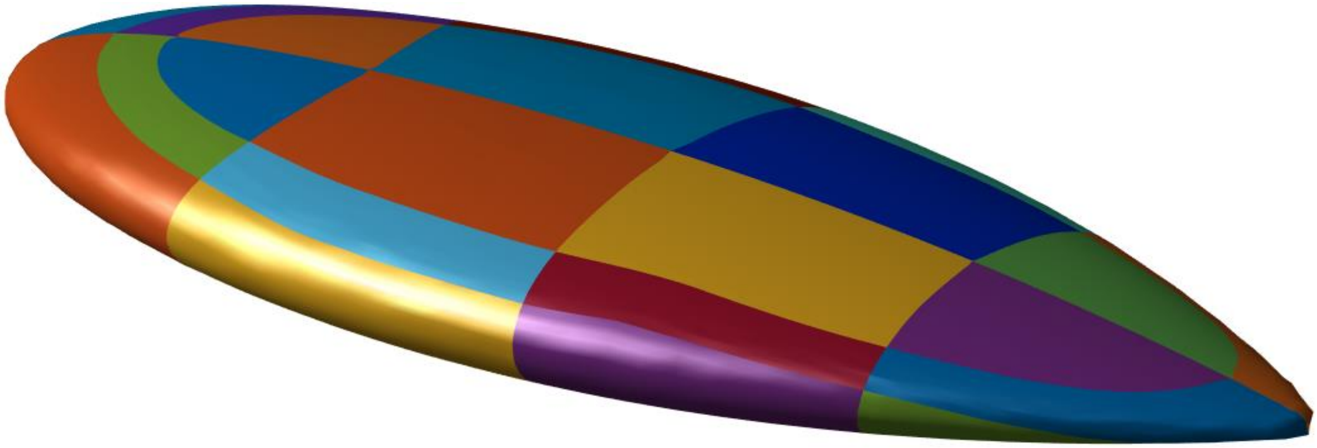

Fig. 11. NASA almond model using adaptive refinement from Fig. 5 with only 32 16th-order continuous quadrilateral elements.

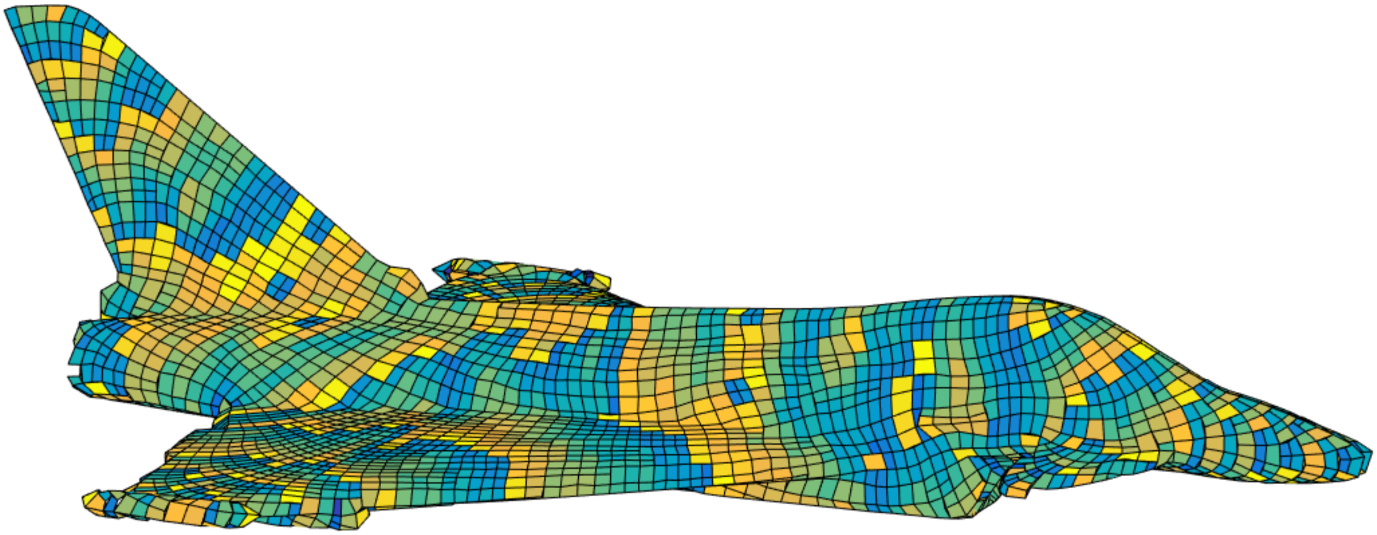

Fig. 12. Fighter jet model constructed from 6490 1st-order discontinuous quadrilateral elements using iterative adaptive refinement from Fig. 7.

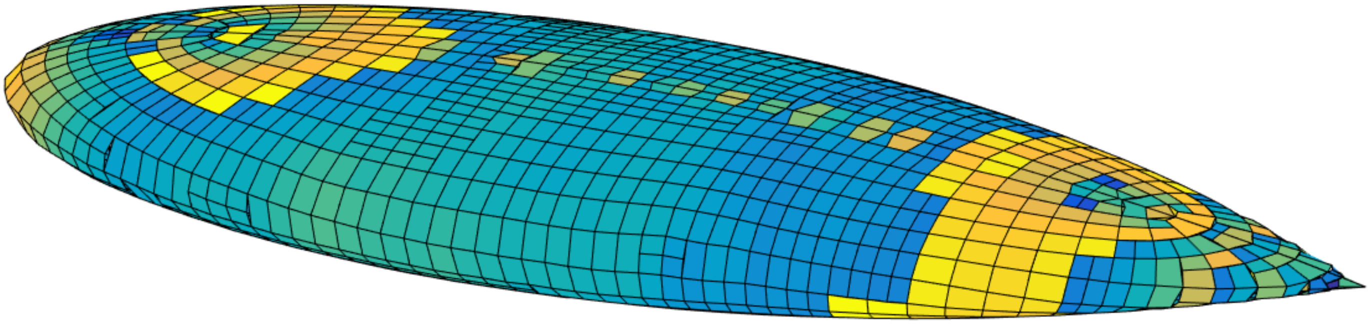

Fig. 13. NASA almond model featuring 2000 1st-order discontinuous quadrilateral elements obtained by the adaptive refinement technique from Fig. 7.

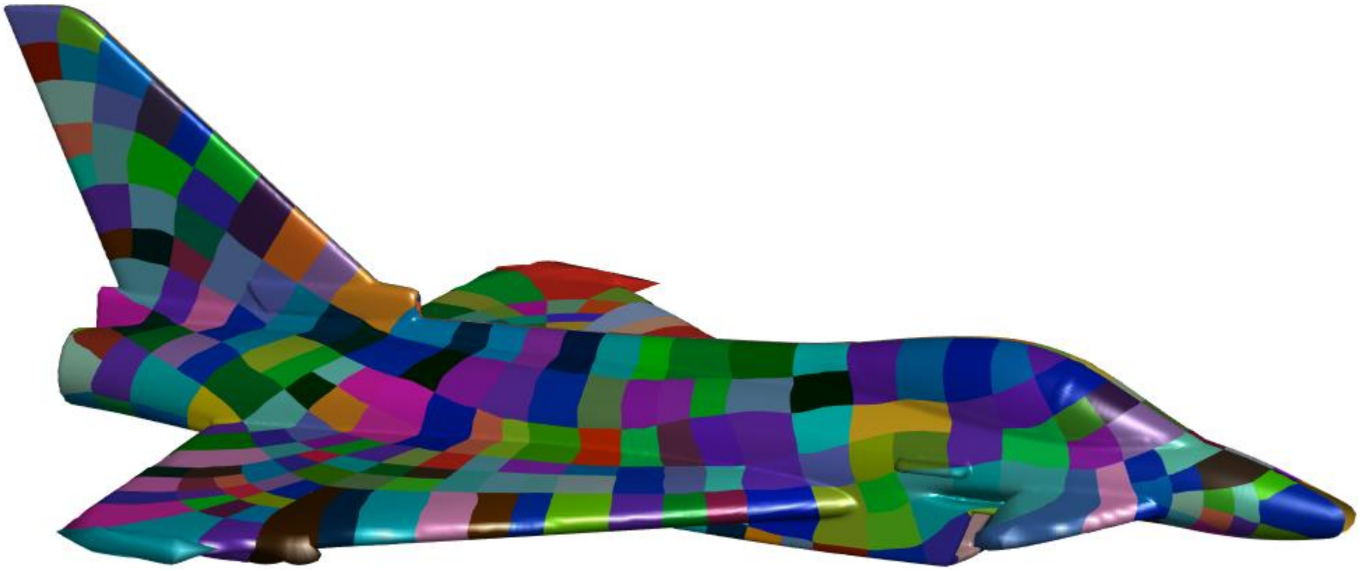

Fig. 14. Fighter jet model composed of 300 30th-order discontinuous quadrilateral elements by the technique in Fig. 7.

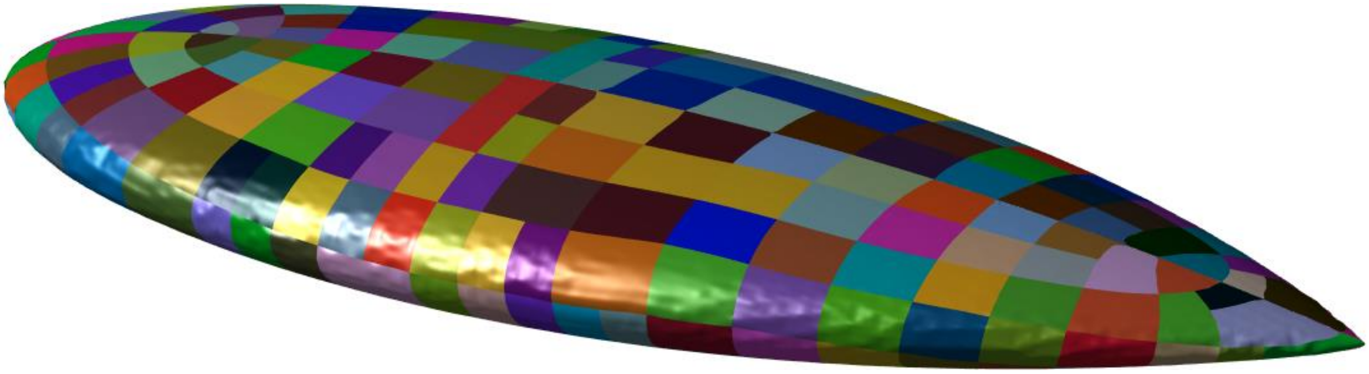

Fig. 15. NASA almond model using refinement from Fig. 7 with 300 10th-order discontinuous quadrilateral elements.

Figure 16 shows a low-resolution meshing of the fighter jet model using 2898 first-order triangular elements. Adaptive sampling was used as outlined in Section III.B. Despite the low element count and lowest possible geometric order, the model is well represented at coarse-scale, showing that the proposed method works well even as a first-order triangular mesher.

Figures 17 and 18 show higher order triangular meshes for the fighter jet and almond, respectively, using the technique outlined in Section III.B. The fighter jet was meshed using 3702 10th-order elements interpolated by cubic spline, while the almond was meshed using 1098 10th-order elements interpolated using Lagrange polynomials. We see good fidelity in both cases. Note that roughness from facets in Σ can again be seen in Fig. 18, similar to Fig. 15.

Figures 19 and 20 show first order quadrilateral meshes for the fighter jet and almond, respectively, using the technique from Section III.D. The fighter jet was meshed using 4562 first order elements while the almond was meshed using 1544 first order elements. Detail from the original surfaces is captured well in these continuous quadrilateral meshes despite their low element count and low order.

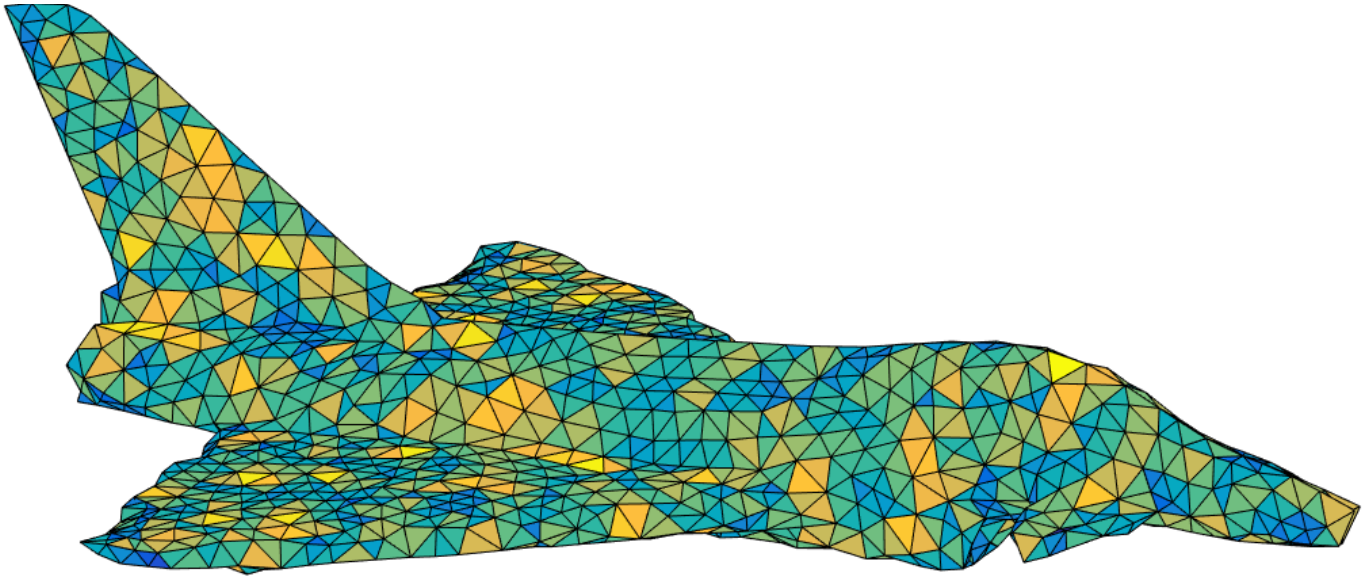

Fig. 16. Low-resolution fighter jet model using 2898 1st-order continuous triangular elements with adaptive refinement as in Fig. 6.

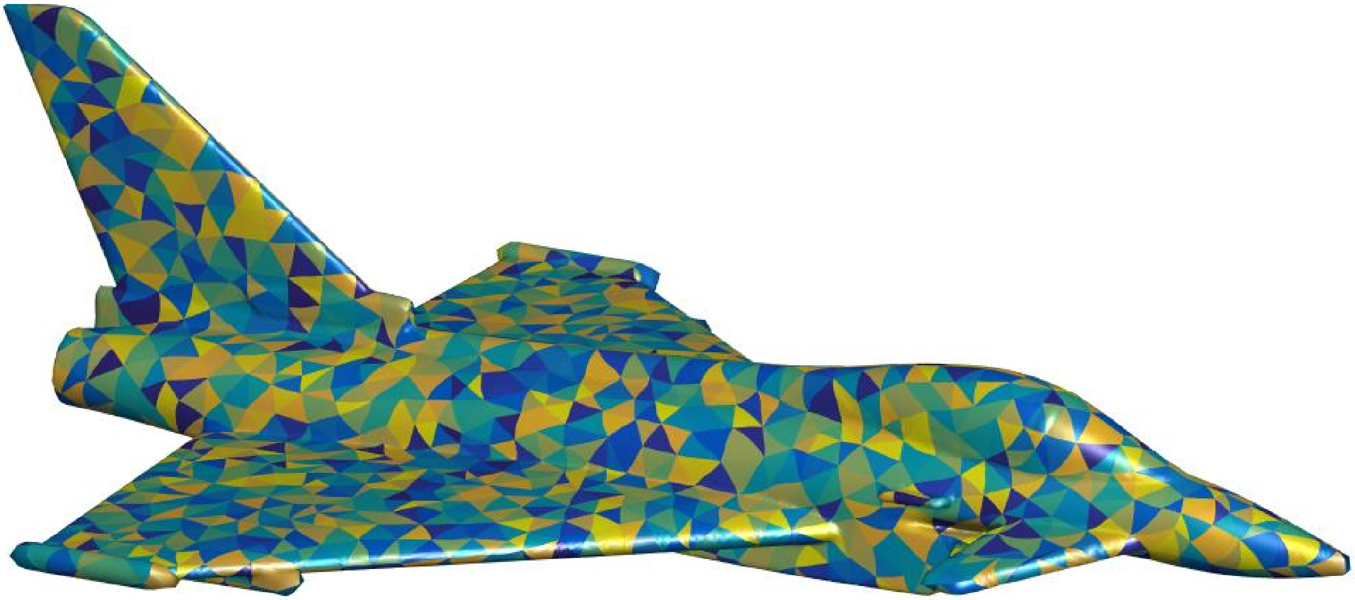

Fig. 17. Fighter jet model featuring 3702 10th-order continuous triangular elements generated by the iterative adaptive refinement technique in Fig. 6.

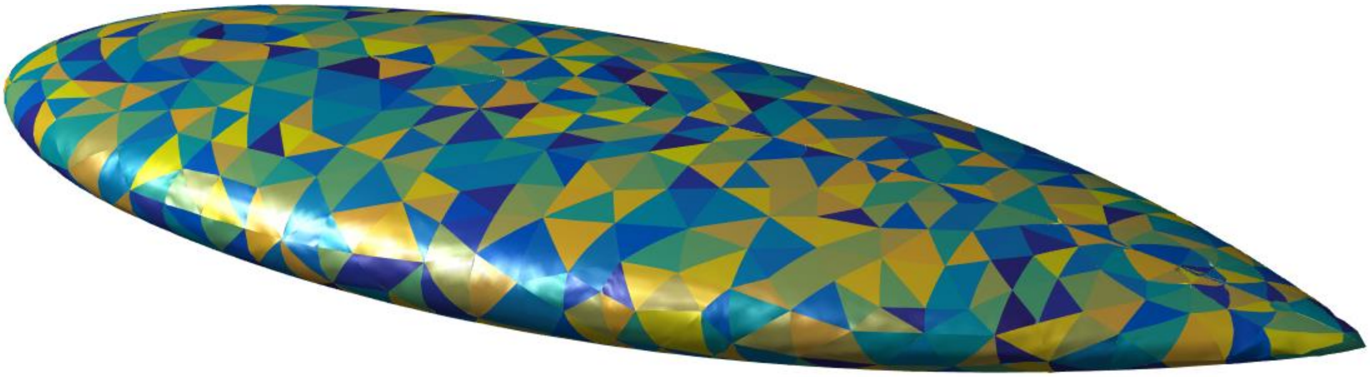

Fig. 18. NASA almond model containing 1098 10th-order continuous triangular elements based on the adaptive refinement from Fig. 6.

B. Corner Angle Measurements

Here we demonstrate the conformality of the DSRF method with adaptive refinement for the almond and fighter jet models.

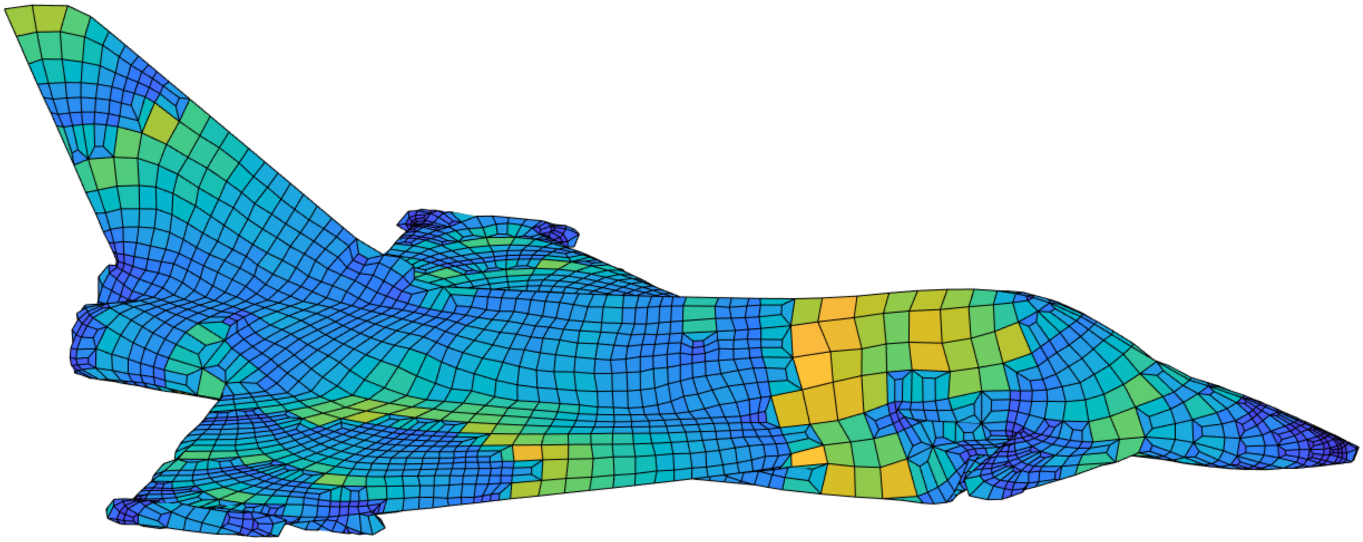

Fig. 19. Fighter jet model featuring 4562 1st-order continuous quadrilateral elements generated by the iterative adaptive refinement technique in Fig. 8.

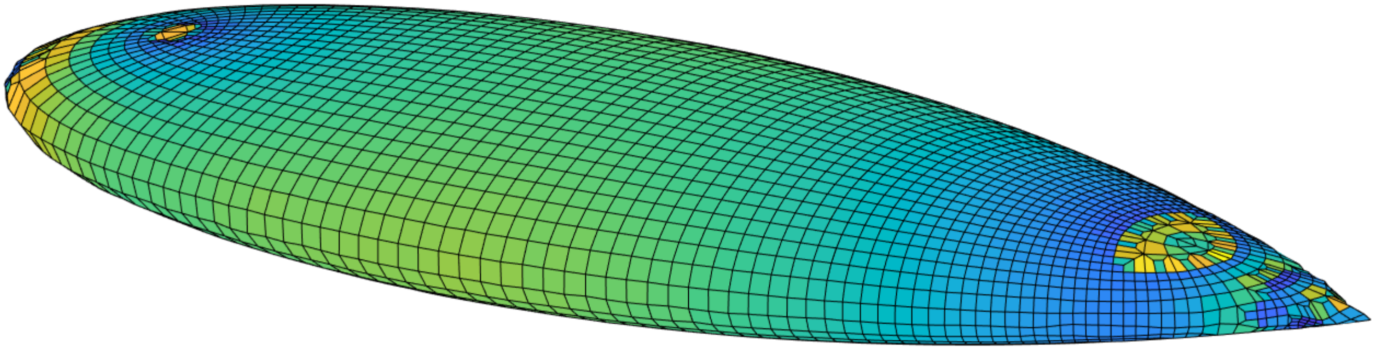

Fig. 20. NASA almond model containing 1544 1st-order continuous quadrilateral elements based on the adaptive refinement from Fig. 8.

For the given examples, we sample both the jet and the almond in the parametric domain adaptively using the refinement indicator and refinement method given for the continuous quadrilateral case in Section III.A. Both meshes were sampled using $N = 256$ to obtain a dense sampling of the conformality of $M(p)$ for both models. Corner angles were computed for every vertex $v \in \tilde{V}$ in the resulting surface meshes, and histograms were produced from the resulting set of corner angles for each mesh and are shown in Fig. 21. Note that almost all corner angles are close or equal to 90°, indicating excellent conformality of the DSRF method. This is of utmost importance for many singularity-extraction techniques used in MoM that are not robust to poor corner angles, but otherwise offer excellent accuracy [32]. Additionally, the conformality of the method is critical to maintain high local orthogonality of the basis functions, thereby controlling system condition number.

V. Conclusions

This paper has addressed a crucial but largely under-investigated aspect of modern computational electromagnetics research: surface mesh generation. We have introduced a robust surface meshing approach intended for use as a geometric discretization technique for MoM-SIE problems in electromagnetics but easily extensible to other applications. The proposed technique makes use of new mathematics that has, to our knowledge, not previously breached the field of CEM. The method uses the discrete surface Ricci flow to generate an accurate discrete conformal mapping from an input surface to a parametric domain in which a seed mesh is defined. Iterative adaptive refinement is then used to refine the seed mesh, from which the final surface mesh is produced by

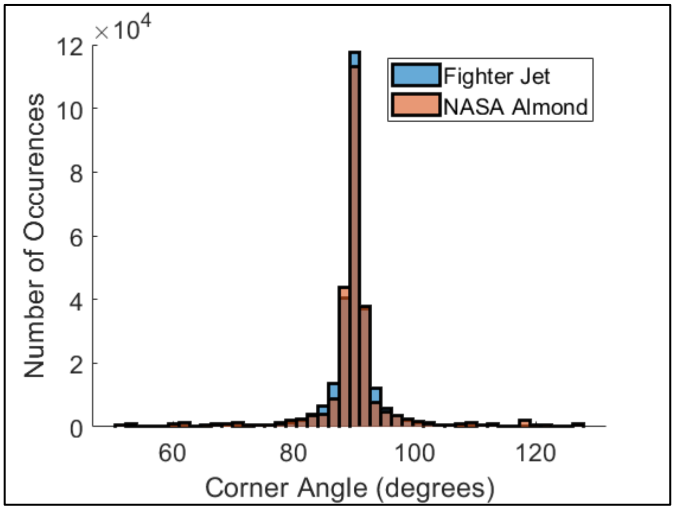

Fig. 21. Corner angle histograms for mappings generated using DSRF on continuous quadrilateral meshes. Mappings were sampled using a 256×256 grid of sample points as in Fig. 5 for the fighter jet in Fig. 4(a) and the NASA almond, respectively.

an inversion of the mapping using barycentric interpolation. The novel proposed technique has been demonstrated capable of high-quality mesh generation for a variety of surface mesh types, given suitable refinement indicators and methods, including triangular, continuous quadrilateral, and discontinuous quadrilateral of both low and high order. We have defined example refinement indicators and methods for the studied mesh types and have offered guidelines for extension to mesh types not covered in this paper.

The mesh generation results presented have shown that DSRF with adaptive refinement easily recreates even complicated initial surfaces using several mesh types over a large range of orders and fidelities. The ability of the new DSRF-based meshing technique to produce high quality meshes even for complicated, highly-varied surfaces has been demonstrated for the NASA almond and a fighter jet model. Where high-fidelity meshing is desired, the proposed DSRF technique has been able to capture fine-scale detail using very few high order elements, here demonstrated with as few as 32 elements of up to 64th-order, unprecedented in the field of CEM. Where low-fidelity meshing is desired, DSRF with adaptive refinement has been able to accurately recreate course-scale detail using standard first-order elements. Corner angle measurements have shown that the generated discrete mappings are highly conformal, leading to excellent angle conservation between parametric and final surface meshes when inverse mapped, yielding meshes ideal of angle-sensitive singularity extraction techniques used in MoM.

While the DSRF method with adaptive refinement has been shown to be effective for the cases tested, we consider this the first publication in a relatively experimental line of research and appropriately, we have noted some drawbacks of the method as presented. The method is not applicable to complicated multi-part objects when continuity between meshes of individual parts is required. It is also not formally applicable to non-differentiable surfaces where perfect preservation of sharp (non-differentiable) features is required. The method can only asymptotically approach preservation of sharp features, so the error introduced may be unacceptable where sharp feature preservation on otherwise low-fidelity meshes is required. We therefore anticipate several areas for future work including improvement of refinement methods to include sharp-feature preference; extension to multi-part objects while maintaining mesh continuity between parts; improvements to the potentially poor computational scaling of the DSRF; and utilization with simulation-derived error data for adaptive refinement to mitigate not only geometric error during the meshing process, but also numerical error in CEM solvers. Additionally, we have only presented the method for the case of closed, simply connected surfaces with one symmetry plane using a cut and the Euclidean DSRF. We consider extending the present work, using the theory of [40] and more-complicated seed meshes, to arbitrary spherical and hyperbolic surfaces as a major component of future work toward broad applicability of the method.

Overall, by leveraging the DSRF, we can provide a unified framework for generating low- or high-order surface meshes of arbitrary element type that integrates with any existing mesh reconstruction tool, to quickly remesh, refine, and optimize. Our DSRF-based technique facilitates the generation of high-quality discretizations, even for sub-optimal parent meshes, with demonstrations presented in this paper for the important quality measure of corner angle uniformity. The ability to automatically generate geometrically ultra-high order elements of high quality demonstrates significant advantages for practical application in CEM, both in reducing the number of unknowns and improving accuracy and robustness. Additionally, mesh refinement or full reconstruction (e.g., first-order triangle to ultra-high order quadrilateral and vice versa) is extremely inexpensive. A precomputed map from the parent surface and its parameterization enables this low-cost reconstruction and may assist many other common and desirable goals such as optimization. As such, DSRF meshes can be locally or globally refined efficiently motivated by geometric constraints, solution error constraints, or both.

References

[1] S. Kim, “Error estimation and adaptive refinement technique in the method of moments,” *Doctoral Thesis,* Georgia Institute of Technology, May 2017.

[2] J. Giesen and M. John, “Surface reconstruction based on a dynamical system,” *Computer Graphics Forum*, vol. 21, no. 3, pp. 363-371, 2002.

[3] R. Renka, “Two Simple Methods for Improving a Triangle Mesh Surface,” *Computer Graphics Forum*, vol. 35, no. 6, pp. 46-58, 2016.

[4] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, “Computational Geometry (2nd revised edition),” *Springer-Verlag*, 2000.

[5] S. M. Rao, D. R. Wilton, and A. W. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” *IEEE Transactions on Antennas Propagation*, vol. 30, no. 5, pp. 409–418, May 1982.

[6] J. Ikaheimo, K. Forsman, and L. Kettunen, “Adaptive Mesh Generation in 2D Magnetostatic Integral Formulations,” *IEEE Transactions on Magnetics,* vol. 33, no. 2, pp. 1736-1739, Mar 1997.

[7] Z. J. Cendes, D. Shenton, and H. Shahnasser, “Magnetic Field Computation Using Delaunary Triangulation and Complementary Finite Element Method,” *IEEE Transactions on Magnetics,* vol. mag-19, no. 6, pp. 2551-2554, Nov 1983.

[8] M. M. Sakamoto, J. R. Cardoso, J. M, Machado, and M. Salles, “A 2-D Delaunay Refinement Algorithm Using an Initial Prerefinement From the Boundary Mesh,” *IEEE Transactions on Magnetics,* vol. 44, no. 6, pp. 1418-1421, Jun 2008.

[9] H. Tsuboi, T. Asahara, F. Kobayashi, and T. Misaki, “Adaptive Triangular Mesh Generation for Boundary Element Method in Three-Dimensional Electrostatic Problems,” *IEEE Transactions on Magnetics,* vol. 34, no. 5, pp. 3379-3382, Sept 1998.

[10] L. Janicke and A. Kost, “Error Estimation and Adaptive Mesh Generation in the 2D and 3D Finite Element Method,” *IEEE Transactions on Magnetics,* vol. 32, no. 3, pp. 1332-1337, May 1996.

[11] S. Dafour, G. Vinsard, B. Laporte, and R. Moretti, “Mesh Improvement in 2-D Eddy-Current Problems,” *IEEE Transactions on Magnetics,* vol. 38, no. 2, pp. 377-380, Mar 2002.

[12] K. Virga and Y. Rahmat-Samii, “RCS Characterization of a Finite Ground Plane with Perforated Apertures: Simulations and Measurements,” *IEEE Transactions on Antennas and Propagation,* vol. 42, no. 11, pp. 1491-1501, Nov 1994.

[13] D. A. Lindholm, “Automatic Triangular Mesh Generation on Surfaces of Polyhedra,” *IEEE Transactions on Magnetics,* vol. mag-19, no. 6, pp. 2539-2542, Nov 1983.

[14] M. Kostic, B. M. Kolundzija, D. S. Sumic, and B. L. Mrdakovic, (2010). “Optimized quadrilateral mesh for higher order method of moment based on triangular mesh decimation,” *Proceedings of the 2010 Antennas and Propagation Symposium,* 2010.

[15] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” *Computer-aided Design*, vol. 10, no. 6, pp. 350–355, 1978.

[16] L. Velho and D. Zorin, “4-8 subdivision,” *Computer Aided Geometric Design*, vol. 18, no. 5, pp. 397–427, 2001.

[17] J. Xia, I. Garcia, Y. He, S. Xin, and G. Patow, “Editable polycube map for gpu-based subdivision surfaces,” *Proceedings of the Symposium on*

Interactive 3D Graphics and Games, 2011, New York, NY, USA, pp. 151–158.

[18] I. Boier-Martin, H. Rushmeier, and J. Jin, "Parameterization of triangle meshes over quadrilateral domains," *Proceedings of the Eurographics Symposium on Geometry Processing*, 2004, Nice, France, pp. 197–208.

[19] B. Lévy, and Y. Liu, "LP Centroidal Voronoi Tesselation and its applications," *ACM Transactions on Graphics,* vol. 29, no. 4, pp. 101-A119, 2010.

[20] F. Kälberer, M. Nieser, and K. Polthier, "Quadcover - surface parameterization using branched coverings," *Computer Graphics Forum,* vol. 26, no. 3, pp. 375–384, 2007.

[21] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez, "Periodic global parameterization," *ACM Transactions on Graphics*, vol. 25, pp. 1460–1485, Oct. 2006.

[22] D. Bommes, H. Zimmer, and L. Kobbelt, "Mixed-integer quadrangulation," *ACM Transactions on Graphics*, vol. 28, pp. 77-78, Jul. 2009.

[23] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin, "Quad-Mesh Generation and Processing: A Survey," *Computer Graphics Forum*, vol. 32, no. 6, pp. 51-76, 2013.

[24] G. Xiao and Y. Hou, "Intuitive Formulation of Discontinuous Galerkin Surface Integral Equations for Electromagnetic Scattering Problems," *IEEE Transactions on Antennas and Propagation,* vol. 65, no. 1, pp. 287-294, Jan 2017.

[25] S. L. Ho, Y. Zhao, and W. N. Fu, "An Efficient Parameterized Mesh Method for Large Shape Variation in Optimal Designs of Electromagnetic Devices," *IEEE Transactions on Magnetics,* vol. 48, no. 11, pp. 4507-4510, Nov 2012.

[26] B. M. Kolundzija, "Automatic Mesh Generation Using Single- and Double-Node Segmentation Techniques," *IEEE Antennas and Propagation Magazine,* vol. 40, no. 4, pp. 30-38, Aug 1998.

[27] E. H. Newman and P. Tulyathan, "A Surface Patch Model for Polygonal Plates," *IEEE Transactions on Antennas and Propagation,* vol. ap-30, no. 4, pp. 588-593, Jul 1982.

[28] T. A. Linkowski and P.M. Slobodzian, "Contour- and Grid-Based Algorithm for Mixed Triangular-Rectangular Planar Mesh Generation," *Progress in Electromagnetics Research B,* vol. 40, pp. 201-220, 2012.

[29] J. Moreno, M. J. Algar, I Gonzalez Diego, and F. Catedra, "A New Mesh Generator Optimized for Electromagnetic Analysis," *Proceedings of the 5th European Conference on Antennas and Propagation,* pp. 1734-1738, 2011.

[30] J. Moreno, M. J. Algar, I. Gonzalez, F. Catedra, "Redesign and Optimization of the Paving Algorithm Applied to Electromagnetic Tools," *Progress in Electromagnetics Research B,* vol. 29, pp. 409-429, 2011.

[31] H. Borouchaki and P. Frey, "Adaptive triangular-quadrilateral mesh generation," *International Journal For Numerical Methods In Engineering*, vol. 41 no. 5, pp. 915-934, 1998.

[32] B. M. Notaros, "Higher Order Frequency-Domain Computational Electromagnetics," *Special Issue on Large and Multiscale Computational Electromagnetics, IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2251-2276, Aug 2008.

[33] R. D. Graglia, D. R.Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 329–342, Mar. 1997.

[34] M. Djordjevic and B. M. Notaros, "Double Higher Order Method of Moments for Surface Integral Equation Modeling of Metallic and Dielectric Antennas and Scatterers," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 8, pp. 2118-2129, Aug 2004.

[35] Z.Q. Xie, R. Sevilla, O. Hassan, and K. Morgan, "The generation of arbitrary order curved meshes for 3d finite element analysis," *Computational Mechanics,* vol. 51, no. 3, pp. 361-374, 2013.

[36] J. C. Young, "Higher-Order Mesh Generation Using Linear Meshes [EM Programmer's Notebook]," *IEEE Antennas and Propagation Magazine,* vol. 61, no. 2, pp. 120-126, Apr 2019.

[37] J. Harmon, C. Key, and B. M. Notaros, "Geometrically Conformal Quadrilateral Surface-Reconstruction for MoM-SIE Simulations," *Proceedings of the 2019 International Applied Computational Electromagnetics Society (ACES) Symposium – ACES2019*, April 15–19, 2019, Miami, Florida, USA.

[38] C. Key and B. M. Notaros, "Automatic Generalized Quadrilateral Surface Meshing in Computational Electromagnetics by Discrete Surface Ricci Flow," accepted for the *2019 IEEE International Symposium on Antennas and Propagation*, July 7–12, 2019, Atlanta, GA.

[39] J. Harmon, C. Key, S. B. Manic, and B. M. Notaros, "Construction and Application of Geometrically Optimal Curvilinear Surface Elements for Double Higher-Order MoM-SIE Modeling," accepted for the *2019 USNC-URSI Radio Science Meeting* (joint with the IEEE AP-S International Symposium), July 7–12, 2019, Atlanta, GA.

[40] G. Zhang, L. Zeng, and G. Yau, "The unified discrete surface Ricci flow," *Graphical Models*, vol. 76 no. 5, pp. 321-339, 2014.

[41] H. D. Cao and X. P. Zhu, "A Complete Proof of the Poincaré and Geometrization Conjectures - Application of the Hamilton-Perelman Theory of the Ricci Flow," *Asian Journal of Mathematics*, vol. 10, no. 2, pp. 165-492, June 2006.

[42] M. Jin, J. Kim, F. Luo, and X. Gu, "Discrete Surface Ricci Flow," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, Oct 2008.

[43] X. Gu, F. Luo, J. Sun, and T. Wu, "A Discrete Uniformization Theorem for Polyhedral Surfaces," *Journal of Differential Geometry,* vol. 109, no. 2, 2013.

[44] X. Gu, R. Guo, F. Luo, J. Sun, and T. Wu, "A Discrete Uniformization Theorem for Polyhedral Surfaces II," *Journal of Differential Geometry*, vol. 109, no. 3, 2014.

[45] R. Garimella, "Conformal Refinement of Unstructured Quadrilateral Meshes," *Proceedings of the 18th International Meshing Roundtable*, pp. 30-44, 2009.

Cam Key (S'16) was born in Fort Collins, CO in 1996. He received his B.S. (2018) and is currently pursuing his Ph.D. in Electrical and Computer Engineering from Colorado State University. His current research interests include uncertainty quantification, error prediction, and optimization for computational science and engineering; computational geometry, meshing, data science, machine learning, artificial intelligence, remote sensing and GIS, and novel applications of numerical methods across disciplines.

Jake Harmon (S'19) was born in Fort Collins, CO in 1996. He received his B.S. (*summa cum laude*) in 2019 and is currently pursuing his Ph.D. in Electrical Engineering from Colorado State University. His current research interests include adaptive numerical methods, uncertainty quantification, computational geometry, and higher order modeling in the finite element method and surface integral equation method of moments.

Branislav M. Notaroš (M'00-SM'03-F'16) received the Dipl.Ing. (B.S.), M.S., and Ph.D. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1988, 1992, and 1995, respectively.

From 1996 to 1999, he was Assistant Professor in the School of Electrical Engineering at the University of Belgrade. He was Assistant and Associate Professor from 1999 to 2006 in the Department of Electrical and Computer Engineering at the University of Massachusetts Dartmouth. He is currently Professor of Electrical and Computer Engineering, University Distinguished Teaching Scholar, and Director of Electromagnetics Laboratory at Colorado State University.

Dr. Notaroš serves as General Chair of the 2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI National Radio Science Meeting and is Associate Editor for the IEEE Transactions on Antennas and Propagation. He serves as Vice President of Applied Computational Electromagnetics Society (ACES) and as Vice-Chair of USNC-URSI Commission B. He was the recipient of the 2005 IEEE MTT-S Microwave Prize, 1999 IEE Marconi Premium, 2019 ACES Technical Achievement Award, 2015 ASEE ECE Distinguished Educator Award, 2015 IEEE Undergraduate Teaching Award, and many other research and teaching international and national awards.