

Mixed-Order Meshes through rp -adaptivity for Surface Fitting to Implicit Geometries*

Ketan Mittal^{†‡} Veselin A. Dobrev[†] Patrick Knupp[§] Tzanio Kolev[†]
Franck Ledoux[¶] Claire Roche^{||} Vladimir Z. Tomov[†]

Abstract

Computational analysis with the finite element method requires geometrically accurate meshes. It is well known that high-order meshes can accurately capture curved surfaces with fewer degrees of freedom in comparison to low-order meshes. Existing techniques for high-order mesh generation typically output meshes with same polynomial order for all elements. However, high order elements away from curvilinear boundaries or interfaces increase the computational cost of the simulation without increasing geometric accuracy. In prior work [5, 21], we have presented one such approach for generating body-fitted *uniform-order* meshes that takes a given mesh and morphs it to align with the surface of interest prescribed as the zero isocontour of a level-set function. We extend this method to generate mixed-order meshes such that curved surfaces of the domain are discretized with high-order elements, while low-order elements are used elsewhere. Numerical experiments demonstrate the robustness of the approach and show that it can be used to generate mixed-order meshes that are much more efficient than high uniform-order meshes. The proposed approach is purely algebraic, and extends to different types of elements (quadrilaterals/triangles/tetrahedron/hexahedra) in two- and three-dimensions.

1 Introduction

Meshes are an integral part of computational analysis using techniques such as the finite element method (FEM) and the spectral element method (SEM). From a mesh generation perspective, there are broadly two main requirements. First, the mesh must accurately

capture the domain boundaries and multimaterial interfaces. Second, the elements of the mesh must have good quality in terms of shape, size, and orientation to ensure accuracy in the numerical solution of the PDE of interest. This latter issue of mesh quality has been studied extensively, and there are various techniques based on r - and h -adaptivity for improving the quality of an existing mesh and increasing the accuracy of the solution [1, 4, 13, 10, 22, 25, 32, 43, 44]. The first issue of automatically generating a geometrically accurate mesh is a challenging open-question, and has led to the development of techniques that support meshes that are not aligned with the exact geometry [16, 23, 30, 38, 33]. These methods usually require complex numerical techniques to ensure robustness and accuracy, and classical approaches based on body-fitted/domain-conforming meshes are usually preferred in many situations.

In the context of body-fitted meshes, it is well known that a high-order mesh can accurately capture curved surfaces at a lower computational cost in comparison to its lower order counterpart. From a mesh generation point of view, it is convenient to generate a uniform-order mesh, i.e. a mesh in which all the elements are represented by the same number of degrees of freedom. Uniform-order meshes are also necessary when the FEM framework cannot support mixed-order/ p -refined/ p -adaptive meshes. For frameworks that support p -refined meshes, high-order elements can increase the computational cost without increasing accuracy when they are used in region where they are not necessary, e.g., around planar surfaces. Note that mesh curvature can also be dictated by a discrete simulation field, e.g., in Arbitrary Lagrangian Eulerian (ALE) simulations the mesh adaptivity is usually driven by the numerical solution [8, 9]. For the purposes of this work we focus on geometry-driven mesh curvature.

From a historical perspective, body-fitted high-order meshes are typically generated by starting with a linear body-fitted mesh, elevating the mesh to a

*Performed under the auspices of the U.S. Department of Energy under Contract DE-AC52-07NA27344 (LLNL-CONF-854941).

[†]Lawrence Livermore National Laboratory, Livermore, CA, U.S.A. {mittal3,dobrev1,kolev1,tomov2}@llnl.gov.

[‡]Corresponding Author.

[§]Dihedral LLC, Bozeman, MT, U.S.A. knupp.patrick@gmail.com.

[¶]CEA, DAM, DIF, F-91297, Arpajon, France. franck.ledoux@cea.fr.

^{||}CEA-CESTA, Le Barp, France. claire.roche@cea.fr.

higher polynomial order, and projecting the higher order nodes of the boundary elements to the domain boundary [42, 12, 31, 14, 41, 34, 36, 24]. Each of these approaches deals with uniform-order meshes. The literature for p -refined mesh generation is very sparse, except mainly the work by Karman et al. [17]. Therein, the authors present a method for taking a CAD geometry and a corresponding linear body-fitted mesh, and sequentially elevating the order of the elements at the boundary, blending the polynomial order increase in the interior, and smoothing the mesh to ensure mesh validity. The p -refinement/order-elevation for the boundary elements is based on the error between the discretized surface and the actual surface, while the order of interior elements is determined by measuring the difference in the shapes of adjacent elements with different polynomial orders.

We explore mesh p -adaptivity in the case when the target surface is described by the zero isocontour of a level-set function. Level-set based descriptions are commonly used for material interfaces in multimaterial configurations [40], and evolving geometries in shape and topology optimization [39, 2, 15], amongst other applications. In Prior work [21, 5], we have presented a technique for taking any mesh (e.g., uniform Cartesian-aligned mesh) and morphing it using node movement (r -adaptivity) to align with the target surface. This approach has proven to be robust and efficient at obtaining body-fitted high-order meshes for shape optimization applications in the FEM framework [5].

The main contribution of this work is to extend the r -adaptivity framework with p -refinement to produce mixed-order meshes. The proposed approach differs from Karman et al. in several ways, namely, [17] addresses p -adaptivity in the context of classical mesh generation, whereas we pose the problem as mesh adaptivity. The motivation for this approach is to be able to use existing FEM frameworks for obtaining body fitted meshes without having to couple with specialized mesh generation software. Consequently, our method assumes that the surface is prescribed by a discrete level-set function, instead of being given through CAD. Furthermore, our r -adaptation approach aligns the mesh with the zero level-set, while simultaneously ensuring good mesh quality through minimization of a global objective. In contrast, [17] uses a sequence of nodal projections (mesh alignment) followed by mesh smoothing to ensure mesh validity. Finally, instead of using the distance between the actual surface and the discretized surface for determining p -refinements, we use a level-set function based estimator.

The remainder of the paper is organized as follows. We introduce the main components of the rp -adaptivity framework in Section 2. Next, in Section 3 we go

into the deeper technical details of the p -adaptive mesh alignment approach. Finally, we present various numerical experiments to demonstrate the efficiency of mixed-order meshes in comparison to uniform-order meshes in Section 4. Summary and directions for future work are given in Section 5.

2 Preliminaries

In this section, we discuss the key components of our finite element framework that are essential for understanding the new rp -adaptivity method. We first introduce some mathematical notation relevant to the method. Next, we describe how p -adaptivity constraints are imposed for degrees of freedom between elements of different polynomial order, and finally discuss the target matrix optimization paradigm (TMOP)-based framework for mesh alignment using r -adaptivity.

2.1 Discrete Mesh Representation In our finite element based framework, the domain $\Omega \subset \mathbb{R}^d$ is discretized as a union of curved mesh elements, Ω^e , $e = 1 \dots N_E$, each of order p . To obtain a discrete representation of these elements, we select a set of scalar basis functions $\{\bar{w}_i\}_{i=1}^{N_p}$, on the reference element $\bar{\Omega}^e$. In the case of tensor-product elements (quadrilaterals in 2D and hexahedra in 3D), $N_p = (p+1)^d$, and the basis spans the space of all polynomials of degree at most p in each variable. These p th-order basis functions are typically chosen to be Lagrange interpolants at the Gauss-Lobatto nodes of the reference element. The position of an element Ω^e in the mesh \mathcal{M} is fully described by a matrix \mathbf{x}_e of size $d \times N_p$ whose columns represent the coordinates of the element *degrees of freedom* (DOFs). Given \mathbf{x}_e , we introduce the map between the reference and physical element, $\Phi_e : \bar{\Omega}^e \rightarrow \mathbb{R}^d$:

$$(2.1) \quad \mathbf{x}(\bar{\mathbf{x}}) = \Phi_e(\bar{\mathbf{x}}) \equiv \sum_{i=1}^{N_p} \mathbf{x}_{e,i} \bar{w}_i(\bar{\mathbf{x}}), \quad \bar{\mathbf{x}} \in \bar{\Omega}^e, \quad \mathbf{x} \in \Omega^e,$$

where $\mathbf{x}_{e,i}$ denotes the i -th column of \mathbf{x}_e , i.e., the i -th node of element Ω^e .

Throughout the manuscript, \mathbf{x} will denote the position function defined by (2.1), \mathbf{x}_e will denote the element-wise vector/matrix of nodal locations for element Ω^e , and $\mathbf{x}_E = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_E}]$ will denote the global vector of nodal locations for all elements.

2.2 TMOP for Mesh Quality Improvement via r -adaptivity For a given element with nodal coordinates \mathbf{x}_e , the Jacobian of the mapping Φ_e at any refer-

ence point $\bar{\mathbf{x}} \in \bar{\Omega}^e$ is
(2.2)

$$A_{ab}(\bar{\mathbf{x}}) = \frac{\partial x_a(\bar{\mathbf{x}})}{\partial \bar{x}_b} = \sum_{i=1}^{N_p} x_{i,a} \frac{\partial \bar{w}_i(\bar{\mathbf{x}})}{\partial \bar{x}_b}, \quad a, b = 1 \dots d$$

where x_a , represents the a th component of \mathbf{x} (2.1), and $x_{i,a}$ represents the a th component of $\mathbf{x}_{e,i}$, i.e., the i th DOF in element Ω^e . The Jacobian matrix A represents the local deformation of the physical element Ω^e with respect to the reference element $\bar{\Omega}^e$ at the reference point $\bar{\mathbf{x}}$. This matrix plays an important role in FEM as it is used to determine mesh validity ($\det(A)$ must be greater than 0 at every point in the mesh), and also used to compute derivatives and integrals. The Jacobian matrix can ultimately impact the accuracy and computational cost of the solution [29]. This lends to the central idea of TMOP of optimizing the mesh to control the local Jacobian $A_{d \times d}$ in the mesh.

The first step for mesh optimization with TMOP is to specify a target transformation matrix $W_{d \times d}$, analogous to $A_{d \times d}$, for each point in the mesh. Target construction is typically guided by the fact that any Jacobian matrix can be written as a composition of four geometric components [20], namely volume, rotation, skewness, and aspect-ratio:

$$(2.3) \quad W_{d \times d} = \underbrace{\zeta}_{[\text{volume}]} \circ \underbrace{R_{d \times d}}_{[\text{rotation}]} \circ \underbrace{Q_{d \times d}}_{[\text{skewness}]} \circ \underbrace{D_{d \times d}}_{[\text{aspect ratio}]} \cdot$$

In practice, the user may specify W as a combination of any of the four fundamental components that they are interested in optimizing the mesh for. For the purposes of this paper, we are mainly concerned with ensuring good element *shape* (skewness and aspect-ratio), so we set W to be that of an ideal element, i.e., square for quad elements, cube for hex elements, and equilateral simplex for triangles and tetrahedrons. Advanced techniques on how W can be constructed for optimizing different geometric parameters, or even for automatically adapting the mesh to the solution of the PDE are given in [8, 9, 18].

The next key component in the TMOP-based approach is a mesh quality metric that measures the deviation between the current Jacobian transformation A and the target transformation W . The mesh quality metric $\mu(T)$, $T = AW^{-1}$ in Figure 1, compares A and W in terms of some of the geometric parameters. For example, $\mu_{2,s} = \frac{|T|^2}{2\tau} - 1$ is a *shape* metric¹ that depends on the skewness and aspect ratio components, but is invariant to orientation/rotation and volume. Here, $|T|$ and τ are the Frobenius norm and determinant of

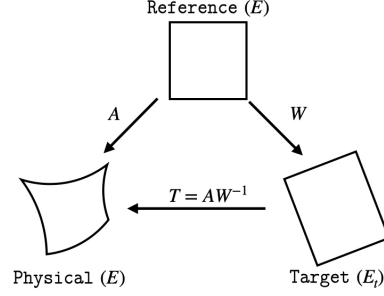


Figure 1: Schematic representation of the major TMOP matrices.

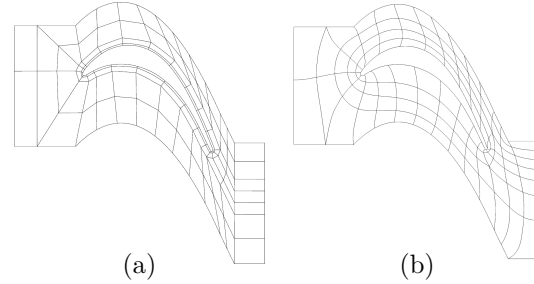


Figure 2: (a) Original and (b) optimized 4th order mesh for a turbine blade.

T , respectively. Similarly, $\mu_{77,v} = \frac{1}{2}(\tau - \tau^{-1})^2$ is a *size/volume* metric that depends only on the volume of the element. We also use *shape+size* metrics such as $\mu_{80,vs} = \gamma\mu_{2,s} + (1 - \gamma)\mu_{77,v}$, $0 \leq \gamma \leq 1$, that depend on volume, skewness and aspect ratio, but are invariant to rotation.

Using the mesh quality metric, the mesh optimization problem is minimizing the global objective:

$$(2.4) \quad F(\mathbf{x}) = \sum_{\Omega^e \in \mathcal{M}} \int_{\Omega^{et}} \mu(T(\mathbf{x})) d\mathbf{x}_t,$$

where F is a sum of the TMOP objective function for each element in the mesh, and Ω^{et} is the target element corresponding to the element Ω^e . Minimizing (2.4) results in node movement such that the local Jacobian transformation A resembles the target transformation W as close as possible at each point, in terms of the geometric parameters enforced by the mesh quality metric. Figure 2 shows an example of high-order mesh optimization for a turbine blade using $W = I$ with a shape metric. The resulting optimized mesh has elements closer to unity aspect ratio and skewness closer to $\pi/2$ radians in comparison to the original mesh, as prescribed by the target $W = I$.

2.3 r -adaptivity for Mesh Quality Improvement and Surface Alignment

In the framework of

¹The metric subscript follows the numbering in [19, 20].

mesh morphing to obtain body-fitted meshes, it is assumed that the target surface is described as the isocontour of a smooth discrete level-set function, $\sigma(\mathbf{x})$. Figures 3(a) and (b) show a simple example of a *squircle* interface represented by the zero isocontour of the level set function, $\sigma(\mathbf{x}) = (x - 0.5)^4 + (y - 0.5)^4 - 0.24^4$, on a third-order multmaterial quadrilateral mesh. To effect alignment of the multmaterial interface with $\sigma(\mathbf{x}) = 0$, the TMOP objective function (2.4) is augmented as follows:

$$(2.5) \quad F(\mathbf{x}) = \underbrace{\sum_{E \in \mathcal{M}} \int_{E_t} \mu(T(\mathbf{x})) d\mathbf{x}_t}_{F_\mu} + w_\sigma \underbrace{\sum_{s \in \mathcal{S}} \sigma^2(\mathbf{x}_s)}_{F_\sigma}.$$

Here, F_σ is a penalty-type term that depends on the penalization weight w_σ , the set of nodes \mathcal{S} discretizing the set \mathcal{F} of mesh faces/edges to be aligned to the level set, and the level set function $\sigma(\mathbf{x})$, evaluated at the positions \mathbf{x}_s of all nodes $s \in \mathcal{S} \in \mathcal{F}$. In Figure 3(b), \mathcal{F} is thus the union of all faces between the two materials (colored blue and orange), and \mathcal{S} is the set of all the nodes that are located on these faces.

Minimizing F_σ represents weak enforcement of $\sigma(\mathbf{x}_s) = 0$, only for the nodes in \mathcal{S} , while ignoring the values of σ for the nodes outside \mathcal{S} . Minimizing the full nonlinear objective function, $F = F_\mu + F_\sigma$, produces a balance between mesh quality and surface fitting.

The objective function (2.5) is usually minimized by the Newton's method. This requires the first- and second-derivative of the objective (F_μ and F_σ) with respect to the node position. The Newton iterations are typically done until the maximum fitting error $|\sigma|_{\mathcal{S}, \infty}$ is below a user-specified threshold:

$$(2.6) \quad |\sigma|_{\mathcal{S}, \infty} := \max_{s \in \mathcal{S}} |\sigma(x_s)|.$$

Here, the maximum fitting error is the maximum value of the level-set function evaluated at the nodes $s \in \mathcal{S} \in \mathcal{F}$. Further implementation details are provided in [5, 21]. Figures 3(c) and (d) show a first- and third-order optimized 4×4 mesh, respectively, obtained by minimizing (2.5). Figures 3(e) and (f) show optimized cubic 8×8 and 16×16 meshes, respectively.

The level-set in Figure 3 is representative of why p -refined meshes are important. Here, the squircle level-set has regions of both high curvature (resembling a circle) and low curvature (resembling a square). Using a linear mesh, as in Figure 3(c), results in an interface aligned mesh where the element vertices are located exactly on the zero level-set. This is due to the node-wise formulation (2.5). But in a continuous sense, it is clear that the linear mesh does not align well. This can be addressed by using a high-order mesh

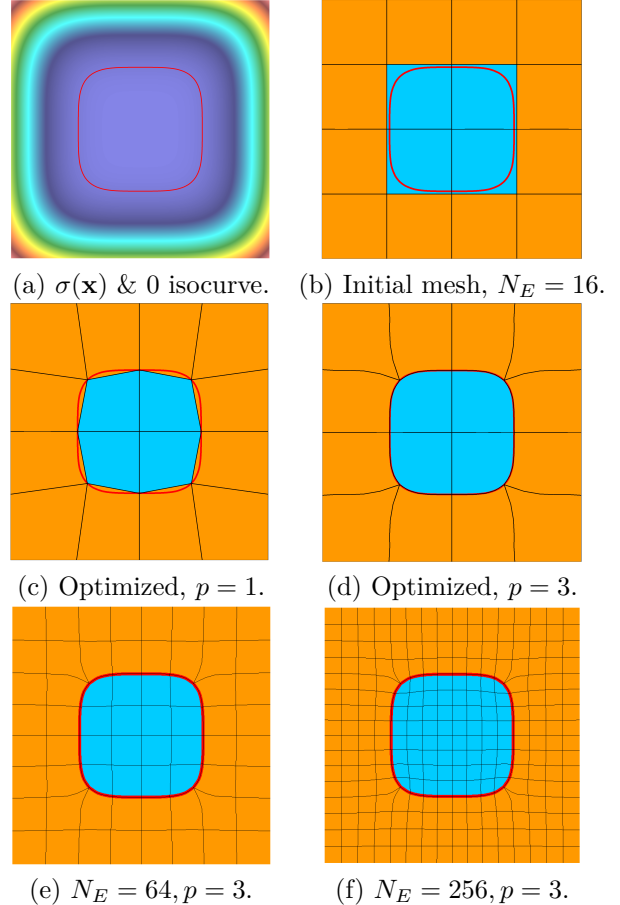


Figure 3: (a) Level set function $\sigma(\mathbf{x})$ with zero isocontour in red, (b) a Cartesian mesh with material interface nodes to be aligned to the zero level set of $\sigma(\mathbf{x})$. (c) Linear mesh and (d) cubic mesh optimized to align with the level-set. Uniformly refined and optimized cubic meshes are also shown with (e) $N_E = 64$ and (f) $N_E = 256$ elements.

as shown in Figure 3(d), which aligns more closely with the zero-isocontour, even in the regions of high curvature. In Figures 3(d)-(f), we also see that as the mesh is uniformly refined, there are a lot more high-order elements than needed, both in the regions of low curvature on the interface and away from the interface. Such extraneous high-order elements could be avoided by using p -refined meshes, which we will address in the next section.

While it is assumed that the prescribed level-set function $\sigma(\mathbf{x})$ is smooth, this is often not the case for real-world applications. The node-wise formulation in (2.5) has proven to be more robust in comparison to an integral-based formulation when the level-set function is not smooth. For smooth functions, an integral-based formulation is expected to be better suited for

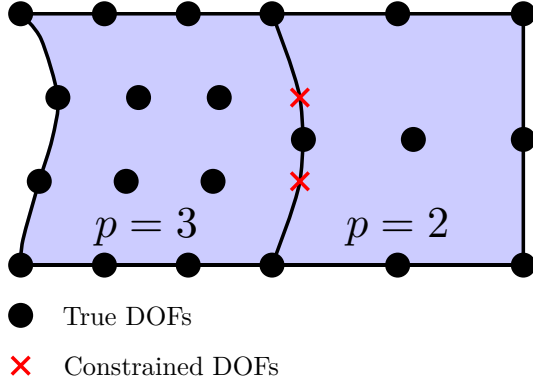


Figure 4: Schematic showing the DOFs for a p -refined mesh.

p -adaptivity, and we will explore this in future work.

2.4 p -Adaptivity Constraints p -refinement introduces hanging/non-conforming nodes between elements of different polynomial orders. In our framework, the hanging nodes on a shared edge/face from higher-order element interpolate from the nodes of the lower-order element. Thus, the accuracy of a discrete finite element function along a shared edge/face is limited by the lowest polynomial order of adjacent elements. Consequently, when we p -refine for mesh fitting at a certain mesh face, the polynomial order of the elements on both sides of the face is elevated. Figure 4 shows a simple example of a two element p -refined mesh with $p = 2$ and 3 depicting the *constrained* and *true* DOFs. A detailed description of how general finite element assembly is handled with hanging nodes in our framework is provided in [3, 7].

3 Methodology

In this section, we describe our approach for p -adaptivity.

3.1 Level-Set Function Representation and Error Computation for Mesh Fitting In our framework, we allow the level-set function to be represented on a different background mesh \mathcal{M}_B than the mesh to be morphed, i.e., $\sigma_B(\mathbf{x}_B)$, where \mathbf{x}_B represents the positions of the background mesh. This decoupling is especially critical when the current mesh \mathcal{M} does not have enough resolution to represent the level-set function and its gradients with sufficient accuracy near the zero level set. Since the discretization error in representing a finite element function depends on the element size h and the polynomial order p , we typically use a background mesh that is adaptively refined around the zero isocontour of the level-set function and has sufficiently high polynomial order.

The example in Figure 3 demonstrates that a node-wise approach is not sufficient for measuring the accuracy of mesh alignment to the level-set. We instead measure the alignment error on each face $f \in \mathcal{F}$ as the squared \mathcal{L}^2 norm of the level-set function on that face:

$$(3.7) \quad e_{f \in \mathcal{F}} := \|\sigma_B\|_{\mathcal{L}^2, f}^2 = \int_f \sigma_B^2(\mathbf{x}).$$

Here we compute the integrated fitting error for a face/edge of the mesh to be morphed ($f \in \mathcal{F} \in \mathcal{M}$) with respect to the level-set function σ_B defined on the \mathcal{M}_B . This integration requires interpolation of σ_B at the quadrature points associated with f .

3.2 Level-Set Interpolation from \mathcal{M}_B to \mathcal{M}

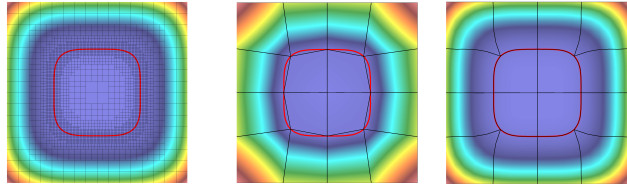
Since we use a background mesh \mathcal{M}_B for $\sigma_B(\mathbf{x}_B)$, it is required to transfer the level set function and its derivatives from \mathcal{M}_B to the nodes $\mathcal{S} \in \mathcal{M}$ prior to each Newton iteration, and at the quadrature points associated with $f \in \mathcal{F}$ prior to computing the integrated error (3.7). This transfer between the background mesh and the current mesh is done using *findpts*, a high-order interpolation library [11]. The *findpts* library enables high-order interpolation at arbitrary points using a sequence of three functions. First, in a pre-processing step (*findpts_setup*), *findpts* constructs some internal data structures based on the input mesh that allow it to do a fast parallel search for arbitrary points in the second step. Next, the *findpts* function takes as an input a set of points $\mathbf{x}^* = (\mathbf{x}_1^*, \mathbf{x}_2^* \dots \mathbf{x}_b^*)$, where b is the number of points to be found, and determines the *computational* coordinates of each point. These computational coordinates are $\mathbf{q}_j^* = (e^*, p^*, \bar{x}^*, \bar{y}^*, \bar{z}^*)_j$ the element e^* , processor p^* , and the corresponding reference space coordinates $\bar{\mathbf{x}}^* = (\bar{x}^*, \bar{y}^*, \bar{z}^*)$. Finally, *findpts_eval* interpolates any given finite element function u at \mathbf{x}_j^* using the computational coordinates returned by *findpts* and a form similar to (2.1). The user is referred to Section 2.3 of [28] and Section 3.2 of [27] for more details. For brevity, we will refer to interpolation of any scalar function $u_B(\mathbf{x}_B)$ at arbitrary points (\mathbf{x}^*) in physical space using *findpts* as:

$$(3.8) \quad u(\mathbf{x}^*) = \mathcal{I}(\mathbf{x}^*, \mathbf{x}_B, u_B(\mathbf{x}_B)).$$

Using (3.8), we compute the integrated fitting error on each face as

$$(3.9) \quad e_{f \in \mathcal{F}} = \sum_{q=1}^{N_q} w_q (\mathcal{I}(\mathbf{x}_q, \mathbf{x}_B, \sigma_B(\mathbf{x}_B)))^2 d\mathbf{x}_q,$$

where N_q is the number of quadrature points on face f and \mathbf{x}_q are the physical space coordinates of the q th



(a) $\sigma_B(\mathbf{x}_B), p_{\sigma_B} = 4$ (b) $\sigma(\mathbf{x}), p = 1$ (c) $\sigma(\mathbf{x}), p = 3$

Figure 5: Interpolating the level-set function from (a) adaptively refined background mesh with $p_{\sigma_B} = 4$ to (b) an optimized linear mesh and (c) an optimized cubic mesh.

quadrature point. From an implementation perspective, we query $findpts$ to get the interpolated value at all the quadrature points simultaneously, and use the interpolated values for integration as needed.

Figure 5 shows an example of high-order interpolation through $findpts$. A quartic representation of the squircle level-set function is interpolated at the nodal positions of the optimized linear- and cubic-mesh. The level-set function is of the same order as the mesh, i.e. $p = 1$ in Figure 5(b) and $p = 3$ in Figure 5(c). As expected, the $p = 3$ case resembles the source function more accurately than the $p = 1$ case.

3.3 p -Refinement Criterion for Interface Elements To motivate our approach, we present Figure 6 which shows how the total fitting error, computed as the sum of integrated fitting error (3.7) over all the faces $f \in \mathcal{F}$ marked for fitting, varies with the number of degrees of freedom for different uniform-order optimized meshes. For each p considered, the coarse 4×4 mesh shown in Figure 3(b) is also refined up to 3 times, each of which are indicated by different data points on the corresponding curve. As we can see in Figure 3(d)-(f), with uniform mesh refinement, we get a lot more elements away from the interface that are essentially linear in shape, and using $p = 3$ for these elements is increasing the computational cost of the system without providing any gain in accuracy for the mesh alignment problem. Also note that for a given number of DOFs, higher polynomial order results in higher accuracy, which is why high-order meshes are preferred over uniformly refined low-order meshes.

The above observations are used to guide our approach for mesh p -refinement. We typically start with the mesh at given polynomial order, e.g., $p_{init} = 1$. We then morph the mesh using the TMOP-based formulation (2.5). Next, we compute the integrated fitting error e_f (3.7) on each face f marked for fitting, and refine the elements adjacent to that face if the error is greater than a user prescribed threshold. From a practical point of

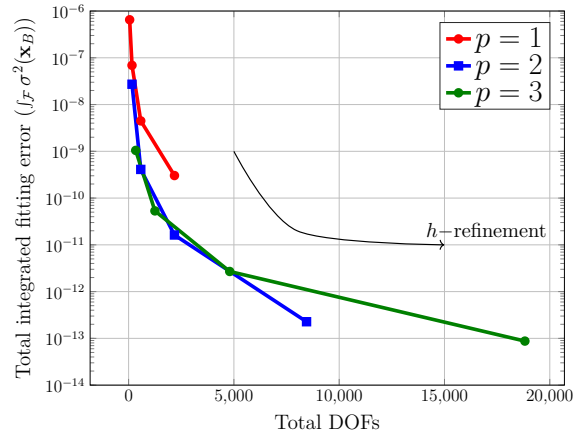


Figure 6: Comparison of total DOFs versus total integrated error for meshes with uniform polynomial orders (p). For each p considered, the coarse 4×4 mesh is uniformly refined up to 3 times, and aligned with the squircle level-set.

view, there are two choices for the refinement threshold (e_{ref}). First is an absolute threshold (γ_1) that is typically guided by the user's knowledge of the level-set function. In this case, elements adjacent to a face are refined if

$$(3.10) \quad \text{Criterion 1: } e_f > e_{ref} = \gamma_1.$$

The second is a relative threshold (determined by γ_2) that depends on the maximum of the integrated fitting error for all the faces in the initial mesh ($e_{\mathcal{F},\infty}$). In this case, adjacent elements are p -refined if

$$(3.11) \quad \text{Criterion 2: } e_f \geq \gamma_2 \cdot e_{\mathcal{F},\infty}.$$

Once the integrated fitting error has been computed for each face and the adjacent elements have been marked for p -refinement, the polynomial order of these elements is increased by a user-prescribed parameter (Δp_{ref}). Since not all the high-order nodes of this p -refined mesh are located on the zero isocontour, the mesh is optimized again by minimizing (2.5). This process of morphing followed by p -refinement is repeated until all the faces have their integrated fitting error below the user-prescribed threshold or the elements have been elevated to maximum allowed polynomial order p_{max} .

In terms of increase in polynomial order p , there are two choices of interest. The first straightforward choice is $\Delta p_{ref} = 1$. A drawback of $\Delta p_{ref} = 1$ is that it can require multiple Newton minimization steps for (2.5) (up to $p_{max} - p_{init}$ times) until the mesh aligns with the target isocontour with the desired accuracy (e_{ref}). Each of these Newton minimization steps has a non-trivial computational cost that increases with p . The second choice is to use $\Delta p_{ref} = p_{max} - p_{init}$ such that

all the interface elements to be refined are elevated to the maximum polynomial order. The mesh can then be morphed to align with the level-set, and no more Newton iterations would be needed afterwards.

Irrespective of what Δp_{ref} is, we first optimize the mesh at p_{init} , because the cost associated with it can be significantly lower in comparison to higher polynomial orders if p_{max} is much greater than p_{init} . This choice is driven by the fact that the storage, assembly, and evaluation of FEM operators scales as $\mathcal{O}(p^{2d})$, $\mathcal{O}(p^{3d})$, and $\mathcal{O}(p^{2d})$, respectively, for a traditional full assembly-based approach [3]. Furthermore, optimizing the mesh at a lower polynomial order, e.g., $p_{\text{init}} = 1$, provides a good initial condition for solving the high-order mesh optimization problem as demonstrated by Ruiz-Gironés et al. [37].

The approach outlined in this section is general in the sense that it gives the user flexibility on a case-by-case basis. Our preliminary experiments show that a robust choice is to start with $p_{\text{init}} = 1$, set $\Delta p_{\text{ref}} = p_{\text{max}} - p_{\text{init}}$, and simply set $e_{\text{ref}} = \gamma_1 = 0$ to refine all elements at the interface once the linear mesh has been aligned with the level-set function.

3.4 p -Derefinement Criterion After we p -refine the mesh as described in the previous section, the morphed mesh could have some interface elements at higher polynomial order than needed, and these elements will unnecessarily increase the computational cost of the actual simulation. To ensure that interface elements are of minimum p required to achieve the desired accuracy, we take the optimized p -refined mesh and check for possible derefinements. This is done by taking the finite element corresponding to each face f , and projecting it to lower order spaces between the current polynomial order and p_{init} sequentially, and finding the lowest polynomial order that maintains the desired accuracy. Unlike the refinement case, the TMOP objective is not minimized again by solving the mesh alignment problem after derefinement, i.e., there is no r -adaptivity step after derefinement. Here, we consider three derefinement criteria. Our first criterion is a threshold relative to the refinement threshold:

$$(3.12) \quad \text{Criterion 1: } e_{f,\hat{p}} < e_{\text{deref}} = \beta_1 e_{\text{ref}},$$

where $e_{f,\hat{p}}$ represents the integrated error of the face f evaluated at polynomial order $p_{\text{init}} < \hat{p} < p_{\text{max}}$, and β_1 is a scalar that controls the desired accuracy relative to the refinement threshold. The second criterion is based on maximum allowed change in the integrated fitting error relative to the current error at order p , i.e.,

$$(3.13) \quad \text{Criterion 2: } e_{f,\hat{p}} < (1 + \beta_2) e_{f,p}.$$

For example, β_2 can be 0.05 if the user wants to allow up to 5% increase for an acceptable derefinement. The third and final criterion is based on maximum decrease in the size l of the edge/face f relative to its current size:

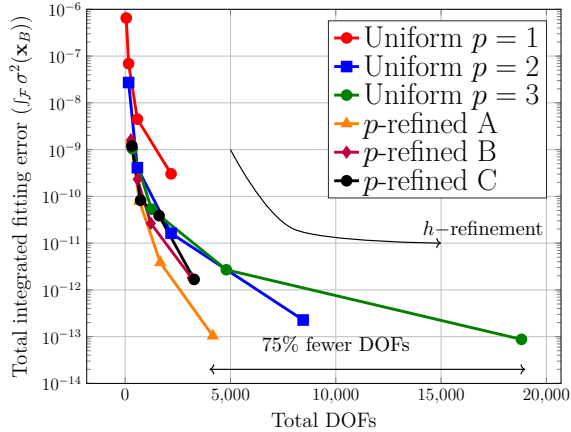
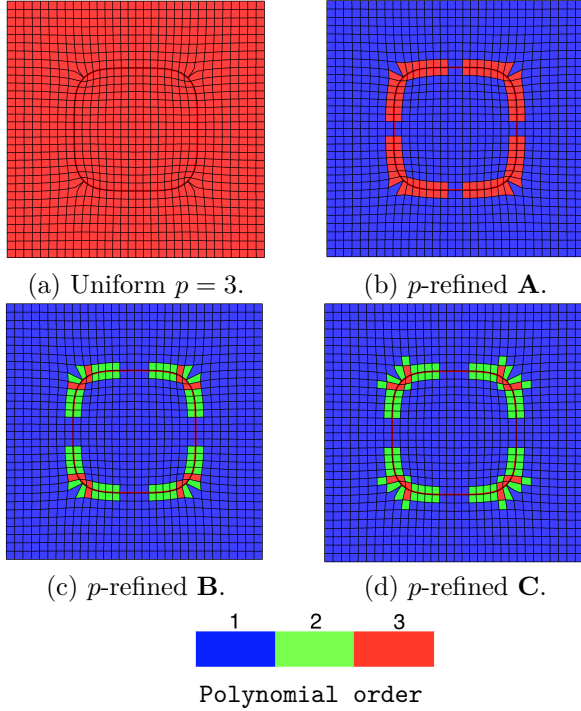
$$(3.14) \quad \text{Criterion 3: } l_{f,\hat{p}} > (1 - \beta_3) l_{f,p}.$$

This criterion is motivated by the fact that a derefinement step can change the shape quality of the element. A small decrease of l suggests that the shape quality around the face remains mostly the same, which is the desired configuration.

Note that when interface elements are derefined, we must also make sure that none of these elements or their neighbors become inverted. This is especially important in high-curvature regions. Hence if a derefinement step leads to an inverted element, the step is rejected.

Figure 7 shows a comparison of the p -refined meshes with uniform-order meshes for the squircle level-set. In this example, $p_{\text{init}} = 1$, and we set $p_{\text{max}} = 3$, $\Delta p_{\text{ref}} = 2$, $\gamma_1 = 10^{-14}$, and $\beta_3 = 10^{-5}$. In each case, we first optimize the linear mesh, then p -refine the mesh using the absolute error-based threshold, and optimize the mesh again. At this point, we show the total DOFs vs integrated error for the aligned mesh as approach **A** in Figure 7(b). As we can see, approach **A** results in the mesh that has essentially the same accuracy as a uniform-order $p = 3$ mesh but at a much lower computational cost. For approach **B** in Figure 7(c), we also derefine the mesh at the end using $\beta_3 = 10^{-5}$. This slightly reduces the total DOFs and accuracy of the alignment as expected. Notice that the optimized mesh has highest order elements of $p = 3$ at the high-curvature region, and $p = 2$ and $p = 1$ at regions of low and no curvature, respectively. Figure 7(b) also demonstrates an example of p -derefinement being rejected to prevent inverting an element. At the high-curvature regions of the squircle, the elements on the outside of the level-set derefine from $p = 3$ to 2, but the element on the inside stays at $p = 3$.

3.5 Propagating p -refinement to Interior Elements As pointed out in [17], for cases like viscous flow simulations where elements near the boundary have high aspect-ratio, mesh curving of only the boundary elements can tangle the mesh. To support mesh curving for such cases, we allow the user to specify the maximum allowable difference (Δp) between the polynomial orders of neighboring elements. This parameter can vary as one moves away from the boundary or simply be held constant. For example, for thin boundary layer elements, a user may want to impose $\Delta p = 0$ for the first few layers adjacent to the boundary and $\Delta p = 1$ away from



(e) Error versus total DOFs for different approaches.

Figure 7: Comparison of uniform-order and p -refined meshes. For uniform polynomial order, a coarse 4×4 mesh is uniformly h -refined up to 3 times, and aligned with the squircle level-set. For p -refined meshes, three different approaches are considered. We start with a linear mesh and align it with the squircle level-set. For approach **A**, we then p -refine the mesh around the interface using $\gamma_1 = 10^{-14}$ (3.10), and then re-align the mesh with the level-set. For approach **B**, we take this mesh and derefine it using the size-based criterion (3.14) with $\beta_3 = 10^{-5}$. For approach **C**, we do the same as approach **B**, but also make sure that the maximum difference in order of adjacent elements is at-most 1 when the mesh is p -refined/derefin.

the boundary. On the other hand, for the case like the squircle, Δp can be simply set to p_{max} so that only the elements at the surface of interest are curved. For lack of a better choice, Δp can also be held constant at 1 so that the element polynomial orders go smoothly from p_{max} to p_{init} as we go away from the surface marked for fitting. From a p -adaptivity perspective, this entails propagating the polynomial orders of the elements adjacent to \mathcal{F} after p -refinement, but before realigning with the level-set, and after p -derefinement. We show the final optimized mesh for the squircle case with $\Delta p=1$ in Figure 7(d) as approach **C**. As evident, using this approach does not increase the total number of DOFs significantly, maintains the fitting accuracy, but can potentially circumvent deterioration in mesh quality. Note from an implementation perspective, the polynomial orders are propagated in our framework through edge-based connections in 2D and face-based connections in 3D.

3.6 Summary of the p -adaptivity Algorithm

Algorithm 1 summarizes our p -adaptivity methodology for mesh alignment. There are three sets of inputs to our method. The first set consists of the parameters that affect the mesh quality, i.e. the target matrix W , the mesh quality metric $\mu(T)$, and the mesh \mathcal{M} with nodal coordinates \mathbf{x} that is to be optimized. The second set impacts the mesh alignment problem from a uniform-order mesh's perspective. These inputs are the background mesh \mathcal{M}_B with nodal coordinates \mathbf{x}_B and the level-set function $\sigma_B(\mathbf{x}_B)$ defined on it. The first two sets together define the r -adaptivity method for mesh morphing to align with a level-set function [5]. The third and final set is the set of parameters that controls the p -adaptivity algorithm. This consists of the maximum polynomial order allowed in the mesh p_{max} , the change in element order upon p -refinement Δp_{ref} , maximum allowed difference in polynomial orders of adjacent elements Δp , and refinement- (γ_1 or γ_2) and derefinement-threshold criterion (β_1 or β_2 or β_3).

Algorithm 1: rp -adaptivity for Mesh Alignment

Input: $\mu, W; \mathcal{M}(\mathbf{x}), \mathcal{M}_B(\mathbf{x}_B), \sigma_B(\mathbf{x}_B);$
 $p_{max}, \Delta p_{ref}, \Delta p, (e_{ref} : \gamma_1|\gamma_2), (e_{deref} : \beta_1|\beta_2|\beta_3)$
Output: Variable order mesh aligned with level-set.

- 1 Determine the set of faces \mathcal{F} that have to be aligned with the level-set function.
 - 2 Minimize (2.5) to align the input mesh (\mathbf{x}) to level-set function ($\sigma_B(\mathbf{x}_B)$) while ensuring good mesh quality as prescribed by μ and W . See Algorithm 1 of [5].
 - 3 **while true do**
 - 4 Compute $e_f \forall f \in \mathcal{F}$, and set polynomial order of faces with $e_f > e_{ref}$ to $\min(p_{max}, p_f + \Delta p_{ref})$, Section 3.2 and 3.3.
 - 5 If no faces are p -refined, **go to** 14.
 - 6 Propagate polynomial orders to interior elements, Section 3.5.
 - 7 Minimize (2.5) for r -adaptivity.
 - 8 **if** $\Delta p_{ref} > 1$ **then**
 - 9 Compute $e_f \forall f \in \mathcal{F}$ and decrease polynomial order of face to minimum order that meets derefinement criterion, Section 3.4.
 - 10 Propagate polynomial orders to interior elements, Section 3.5.
 - 11 **end**
 - 12 If elements were refined up to p_{max} at Step 4, **go to** 14.
 - 13 **end**
 - 14 Return variable order mesh aligned with $\sigma_B(\mathbf{x}_B)$.
-

4 Results

In this section, we demonstrate the impact of our rp -refinement approach using various numerical experiments and show that it extends to different element types in 2D and 3D. Our implementation uses the general finite element infrastructure provided by the MFEM finite element library [3, 26].

4.1 p -Refinement in 3D For our first example, we consider the 3D variant of the squircle level-set. The domain $\Omega \in [0, 1]^3$ is modeled using a $16 \times 16 \times 16$ uniform Cartesian mesh, and the target surface is prescribed through the level-set function:

$$\sigma(\mathbf{x}) = (x - 0.5)^4 + (y - 0.5)^4 + (z - 0.5)^4 - 0.3^4,$$

which is discretized on an adaptively refined background mesh with $p_{\sigma_B} = 4$. A slice-view of the background mesh is shown with the zero isosurface of the level-set in Figure 8(a), and a slice-view of the uniform linear mesh to be optimized is shown in Figure 8(b).

For mixed-order curving, we first take the linear uniform mesh, and morph it to align with the target

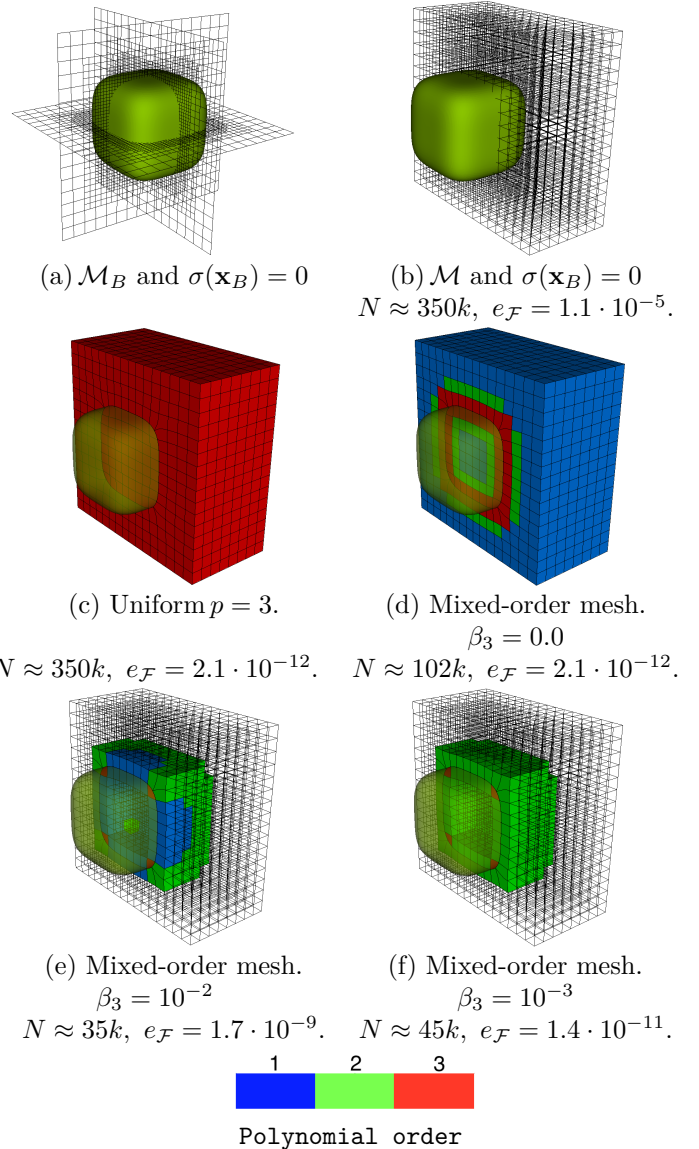


Figure 8: Mixed order mesh curving for 3D. The zero isosurface of the level-set function $\sigma(\mathbf{x}_B) = 0$ defined on the background mesh (\mathcal{M}_B) is shown for each case, and only the cut-view of \mathcal{M} is shown for clarity. For the p -refined meshes, we start with $p_{init} = 1$ and use $p_{max} = 3, \Delta p = 1, \gamma_1 = 0$. The final mixed-order meshes are shown for $\beta_3 = 0$ (no p -derefinement), 10^{-2} and 10^{-3} . The number of DOFs (N) and fitting accuracy $e_{\mathcal{F}}$ are indicated for the initial and optimized meshes.

surface. Then, we p -refine the mesh around the interface uniformly (i.e. $\gamma_1 = 0.0$) with $p_{max} = 3, \Delta p = 1$, and re-align it with the level-set. Figure 8(d) shows the p -refined mesh with $p = 3$ around \mathcal{F} . Here, $\Delta p = 1$, and as a result the difference in polynomial order of face neighbors is 1. Finally, we take this optimized mesh and

derefine it using the size-based criterion (3.14). Figure 8(e) and (f) shows the mixed-order mesh with $\beta_3 = 10^{-2}$ and 10^{-3} , respectively.

Figure 9 shows a comparison in the accuracy of alignment of the uniform-order meshes at $p=1, 2,$ and $3,$ with the p -refined meshes. For the uniform-order meshes, increasing p increases the computational cost significantly without much increase in the geometric accuracy. This is similar to what we had observed in the 2D example. For p -refined meshes, elevating the polynomial order of elements at the interface to $p_{max} = 3$ with $\Delta p = 1$ gives us the same geometric accuracy as a uniform $p = 3$ mesh but at a 71% lower computational cost in terms of DOFs ($N = 102435$ versus 352947). The p -refined mesh also has fewer DOFs than a uniform-order quadratic mesh, and provides higher accuracy. Figure 9 also shows that derefining the mesh using the size-based criterion decreases the number of DOFs and the fitting accuracy as expected. The number of linear, quadratic, and cubic elements in each of the mixed-order meshes are summarized in Table 1.

	p=1	p=2	p=3
$\beta_3 = 0$	2784	536	776
$\beta_3 = 10^{-3}$	3320	696	80
$\beta_3 = 10^{-2}$	3704	312	80

Table 1: Number of linear, quadratic, and cubic elements for each of the mixed-order meshes aligned with the 3D level-set.

Recall that in our approach, we elevate polynomial order of elements at the interface based on the integrated error on the faces that are aligned to the level-set function (Section 3.3). Since the lowest-order entity constraints the solution, it is essential that in 3D, elements having only edges on the interface are also p -refined when necessary (Section 2.4). We handle such elements by setting their polynomial order as the maximum of the polynomial orders of their neighbors that have a face on the interface. This is essential for obtaining accurate alignment with the level-set in 3D. Figure 8(d) shows there are 4 such corner elements in region of high curvature that have an edge on the interface, which are elevated to $p = 3$ due to their neighbors.

4.2 Domain Prescribed by Geometric Primitives Curvilinear domains are commonly prescribed as a combination of geometric primitives in Constructive Solid Geometry (CSG) [35]. Figure 10 shows one such example adapted from [5] where the domain is prescribed as an intersection of geometric primitive for a circle, parabola, and a trapezium. Each geometric prim-

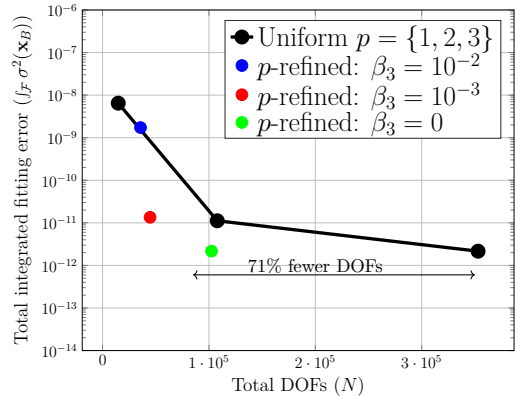


Figure 9: Comparison of total DOFs versus total integrated error for uniform-order and p -refined meshes in 3D. We consider $p=1, 2,$ and 3 for uniform-order meshes. For p -refinement, we use $p_{init} = 1, p_{max} = 3, \Delta p = 1, \gamma_1 = 0,$ and show the mesh for $\beta_3 = 0, \beta_3 = 10^{-2},$ and $10^{-3}.$

itive is prescribed as a step function $\mathcal{G}(x_B)$ that is 1 inside one material and -1 inside the other material. For such cases, we start with a coarse background mesh and use adaptive mesh refinement around the zero level set of $\mathcal{G}(x_B)$. When the zero isocontour interface intersects with the boundary of the domain, as in the present case, we configure the background mesh to completely encompass the spatial domain of the morphed mesh $\mathcal{M},$ and extend slightly beyond it, i.e. $\Omega_B \supseteq \Omega.$ This is essential for accurate computation of derivatives near the boundary of $\mathcal{M}.$ Figure 10(a) shows the resulting background mesh $\mathcal{M}_B.$ Next, we compute a discrete distance function $\sigma_B(\mathbf{x}_B)$ using the p-Laplacian solver of [6], Section 7, from the zero level set of $\mathcal{G}(x_B).$ Figure 10(b) shows the mesh to be morphed $\mathcal{M},$ the level-set function $\sigma_B(\mathbf{x}_B),$ and its zero isocontour.

With the level-set function defined on $\mathcal{M}_B,$ we assign the fictitious material indicators to elements in \mathcal{M} to determine element faces that will be aligned with $\sigma_B(\mathbf{x}_B).$ The uniform linear mesh with two materials is shown in Figure 10(c). To minimize the TMOP problem, we use a shape metric with target transformation W set to be that for an equilateral triangle everywhere in the domain. The optimized linear mesh aligned with the zero isocontour of $\sigma_B(\mathbf{x}_B)$ is shown in Figure 10(d). Note that due to the prescribed mesh quality metric and target, the elements away from the interface have optimized to as close to equilateral triangle as possible. As expected, the optimized linear mesh aligns well with the zero isocontour in regions of low curvature, but cannot capture the surface accurately in high curvature regions due to lack of degrees of freedom.

Next, using $e_{ref}=0.,$ all the elements at the interface

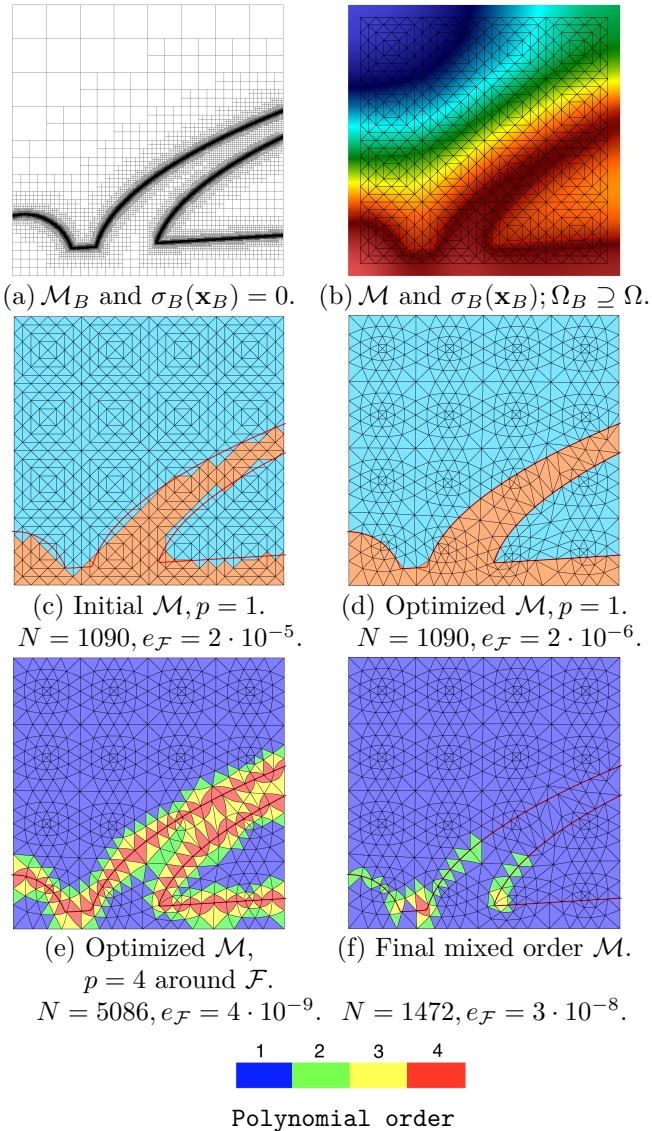


Figure 10: Mixed order mesh curving for a Fischer-Tropsch reactor like domain. For the p -refined meshes, the maximum element order is $p_{max} = 4$, $\Delta p = 1$, and the derefinement is done using size-based criterion (3.14) using 10^{-4} . The zero isosurface of the level-set function $\sigma_B(\mathbf{x}_B) = 0$ defined on the background mesh \mathcal{M}_B is shown for each case. The number of DOFs (N) and fitting accuracy $e_{\mathcal{F}}$ are indicated for the initial and optimized meshes.

are refined to $p_{max} = 4$ with $\Delta p = 1$. This p -refined mesh is then morphed again to align with the level-set function by minimizing the TMOP objective, as shown in Figure 10(e). Finally, we derefine the elements based on the relative change in element size criterion ($\beta_3 = 10^{-4}$ in (3.14)), to obtain the mixed order mesh shown in Figure 10(f).

In terms of accuracy, the initial linear mesh has 1024 elements with 1090 DOFs and a total integrated fitting error of $1.5 \cdot 10^{-5}$. This fitting error reduces to $1.5 \cdot 10^{-6}$ in the morphed linear mesh. The p -refined mesh with quartic elements at the interface has 5086 DOFs and $e_{\mathcal{F}} = 3.8 \cdot 10^{-9}$. Finally, the mixed order mesh after derefinement has 967 linear elements, 45 quadratic elements, 10 cubic elements, and 2 quartic elements (Figure 10(f)), with a total of 1472 DOFs and a fitting accuracy of $e_{\mathcal{F}} = 2.6 \cdot 10^{-8}$. In contrast, the total number of DOFs would have been 16642 had a uniform fourth order mesh been used. Thus, this final mixed order mesh with has 91% fewer DOFs in comparison to a uniform-order mesh. Note that in this example, the primary source of geometrical error are the *sharp* corners in the surface of interest. Ideally, such regions should be handled with higher mesh resolution, and we will explore augmenting our framework with h -refinement in future work.

4.3 Coarse Mixed-Order Mesh Starting From a Dense Low-Order Mesh While high-order meshes have become popular in recent times, a lot of classical mesh generation tools still support only low-order meshes. In this example, we show use of our framework to obtain a coarse mixed-order mesh starting from a dense low-order mesh.

Figure 11(a) shows a linear mesh (\mathcal{M}_{lin}) generated for the 2D Apollo capsule. Using this linear mesh we construct a background coarse quad mesh \mathcal{M}_B and adaptively refine it based on the boundary of \mathcal{M}_{lin} . We use findpts (Section 3.2) to determine which elements of \mathcal{M}_B intersect with the boundary of \mathcal{M}_{lin} . Figure 11(b) shows the overlap between \mathcal{M}_B and \mathcal{M}_{lin} . Next, we generate a discrete grid function $\phi_B(\mathbf{x}_B)$ on \mathcal{M}_B that indicates whether a given node of \mathcal{M}_B is inside ($\phi_B = +1$) or outside ($\phi_B = -1$) the domain of \mathcal{M}_{lin} . This discrete function is similar to the non-smooth representations that we get for geometric primitives in our CSG-based approach, as described in the previous section. The level-set function σ_B is then extracted as the distance function to the zero-isocontour of ϕ_B . The error in the zero isocontour of this level-set function σ_B , with respect to the true boundary of \mathcal{M}_{lin} , is proportional to the element size of \mathcal{M}_B , and we use 12 levels of AMR for \mathcal{M}_B to decrease this error. Figure 11(c) shows the level-set function $\sigma_B(\mathbf{x}_B)$, its zero isocontour, and the original linear mesh \mathcal{M}_{lin} .

With the level-set function σ_B , we obtain a mixed-order mesh using the approach presented in this work. Starting from a uniform linear triangular mesh, we mark the elements based on whether they are located inside or outside the domain, see Figure 11(d). The elements

determined to be completely outside the domain are removed from the mesh, and the boundary of the new mesh is aligned to $\sigma_B(\mathbf{x}_B) = 0$. The optimized linear mesh is shown in Figure 11(e). Next, we elevate all elements adjacent to the interface to $p_{max} = 4$ and use $\Delta p = 1$ to ensure that the maximum different between polynomial order of edge-neighbors is 1. The optimized mesh is shown in Figure 11(f) along with the polynomial order of each element. The final mixed order mesh is shown in Figure 11(g) is obtained by using $\beta_3 = 2 \cdot 10^{-4}$.

The total integrated error for the final mixed-order mesh is 10^{-8} . The original mesh in Figure 11(d) has 526 elements, which is trimmed to 390 elements shown in Figure 11(e). The final optimized mixed-order mesh has 359 linear, 19 quadratic, 9 cubic, and 4 quartic element. We note that the maximum local error in the final mixed-order mesh is at the element with $p = 4$ at the left corner. This is due to the high-curvature of the zero isocontour in that region, combined with the overall coarse resolution in the mesh. This is similar to what was observed in the example in the previous section. In future work, we will seek to address this issue using h -refinement such that local mesh resolution can be enhanced on demand when p_{max} is not high enough to obtain the desired accuracy.

5 Summary and Future Work

We have presented a novel approach for generating mixed-order meshes through rp -adaptivity in the context of surface fitting. This approach uses TMOP for ensuring good mesh quality as a mesh is morphed to align with the surface prescribed as the zero isocontour of a level-set function. The proposed approach is purely algebraic, and extends to different element types in 2D and 3D. We have proposed different possibilities for setting up the mesh p -adaptivity problem, and demonstrated how it can be used to generate mixed order meshes for problems of practical interest. We have also identified ways to further improve the effectiveness of the method, for example, using h -refinement to augment local resolution when needed. In future work, we will present an automated algorithm based on hrp -adaptivity to robustly obtain accurate mixed order meshes for different problem types with minimal user input.

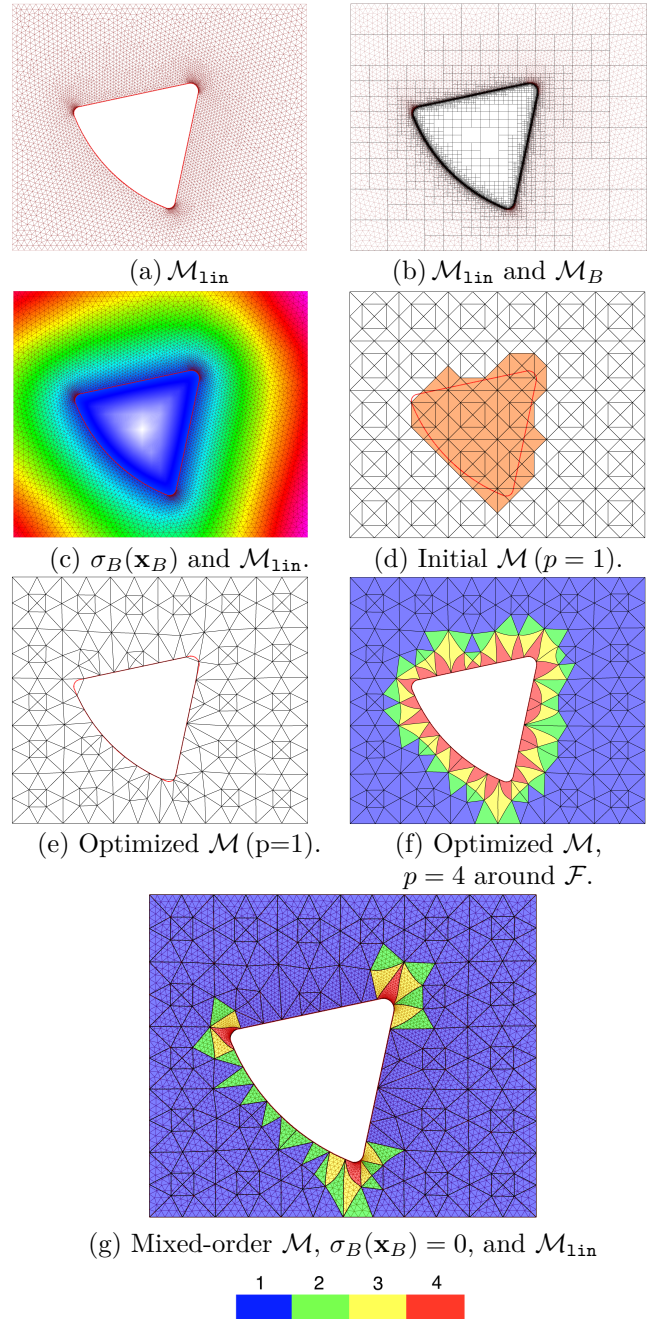


Figure 11: Generating a mixed order mesh for the Apollo capsule starting from a dense linear mesh. The input linear mesh \mathcal{M}_{lin} is used to define a level-set function $\sigma_B(\mathbf{x}_B)$ on an adaptively refined background mesh \mathcal{M}_B . A coarse linear mesh \mathcal{M} is then aligned with $\sigma_B(\mathbf{x}_B) = 0$, and p -refined using $p_{max} = 4$, $\Delta p = 1$, and $\gamma_1 = 0$. The p -refined mesh is then optimized to align with $\sigma_B(\mathbf{x}_B) = 0$, and derefined using (3.14) with $\beta_3 = 2 \cdot 10^{-4}$ to obtain the final mixed-order mesh.

References

- [1] F. ALAUZET AND A. LOSEILLE, *A decade of progress on anisotropic mesh adaptation for computational fluid dynamics*, *Computer-Aided Design*, 72 (2016), pp. 13–39.
- [2] G. ALLAIRE, C. DAPOGNY, AND P. FREY, *Shape optimization with a level set based mesh evolution method*, *Computer Methods in Applied Mechanics and Engineering*, 282 (2014), pp. 22–53.
- [3] R. ANDERSON, J. ANDREJ, A. BARKER, J. BRAMWELL, J.-S. CAMIER, J. CERVENY, V. A. DOBREV, Y. DUDOUIT, A. FISHER, T. V. KOLEV, W. PAZNER, M. STOWELL, V. Z. TOMOV, I. AKKERMAN, J. DAHM, D. MEDINA, AND S. ZAMPINI, *MFEM: a modular finite elements methods library*, *Computers & Mathematics with Applications*, 81 (2021), pp. 42–74.
- [4] G. APARICIO-ESTREMS, A. GARGALLO-PEIRÓ, AND X. ROCA, *Combining high-order metric interpolation and geometry implicitization for curved r-adaption*, *Computer-Aided Design*, 157 (2023), p. 103478.
- [5] J.-L. BARRERA, T. KOLEV, K. MITTAL, AND V. TOMOV, *High-order mesh morphing for boundary and interface fitting to implicit geometries*, *Computer-Aided Design*, 158 (2023), p. 103499.
- [6] A. G. BELYAEV AND P.-A. FAYOLLE, *On variational and PDE-based distance function approximations*, in *Computer Graphics Forum*, vol. 34, Wiley Online Library, 2015, pp. 104–118.
- [7] J. CERVENY, V. DOBREV, AND T. KOLEV, *Nonconforming mesh refinement for high-order finite elements*, *SIAM Journal on Scientific Computing*, 41 (2019), pp. C367–C392.
- [8] V. A. DOBREV, P. KNUPP, T. V. KOLEV, K. MITTAL, R. N. RIEBEN, AND V. Z. TOMOV, *Simulation-driven optimization of high-order meshes in ALE hydrodynamics*, *Computers & Fluids*, (2020).
- [9] V. A. DOBREV, P. KNUPP, T. V. KOLEV, K. MITTAL, AND V. Z. TOMOV, *hr-adaptivity for nonconforming high-order meshes with the Target-Matrix Optimization Paradigm*, *Engineering with Computers*, (2021).
- [10] K. J. FIDKOWSKI AND D. L. DARMOFAL, *Review of output-based error estimation and mesh adaptation in computational fluid dynamics*, *AIAA journal*, 49 (2011), pp. 673–694.
- [11] P. FISCHER, *GSLIB: sparse communication library [Software]*. <https://github.com/gslib/gslib>, 2017.
- [12] M. FORTUNATO AND P.-O. PERSSON, *High-order unstructured curved mesh generation using the Winslow equations*, *Journal of Computational Physics*, 307 (2016), pp. 1–14.
- [13] P.-J. FREY AND F. ALAUZET, *Anisotropic mesh adaptation for cfd computations*, *Computer methods in applied mechanics and engineering*, 194 (2005), pp. 5068–5082.
- [14] A. GARGALLO-PEIRÓ, X. ROCA, J. PERAIRE, AND J. SARRATE, *A distortion measure to validate and generate curved high-order meshes on CAD surfaces with independence of parameterization*, *International Journal for Numerical Methods in Engineering*, 106 (2016), pp. 1100–1130.
- [15] M. HOJJAT, E. STAVROPOULOU, AND K.-U. BLETZINGER, *The vertex morphing method for node-based shape optimization*, *Computer Methods in Applied Mechanics and Engineering*, 268 (2014), pp. 494–513.
- [16] D. M. INGRAM, D. M. CAUSON, AND C. G. MINGHAM, *Developments in cartesian cut cell methods*, *Mathematics and Computers in Simulation*, 61 (2003), pp. 561–572.
- [17] S. L. KARMAN, K. KARMAN-SHOEMAKE, AND C. D. WOEBER, *Mixed order mesh curving*, in *Mesh Generation and Adaptation: Cutting-Edge Techniques*, Springer, 2022, pp. 1–21.
- [18] P. KNUPP, *Target formulation and construction in mesh quality improvement*, Tech. Rep. LLNL-TR-795097, Lawrence Livermore National Lab.(LLNL), Livermore, CA, 2019.
- [19] P. KNUPP, *Metric type in the target-matrix mesh optimization paradigm*, Tech. Rep. LLNL-TR-817490, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2020.
- [20] P. KNUPP, *Geometric parameters in the target matrix mesh optimization paradigm*, *Partial Differential Equations in Applied Mathematics*, (2022), p. 100390.
- [21] P. KNUPP, T. KOLEV, K. MITTAL, AND V. Z. TOMOV, *Adaptive surface fitting and tangential relaxation for high-order mesh optimization*, arXiv preprint arXiv:2105.12165, (2021).
- [22] A. LOSEILLE, A. DERVIEUX, AND F. ALAUZET, *Fully anisotropic goal-oriented mesh adaptation for 3d steady euler equations*, *Journal of computational physics*, 229 (2010), pp. 2866–2897.
- [23] A. MAIN AND G. SCOVAZZI, *The shifted boundary method for embedded domain computations. part i: Poisson and stokes problems*, *Journal of Computational Physics*, 372 (2018), pp. 972–995.
- [24] J. MARCON, G. CASTIGLIONI, D. MOXEY, S. J. SHERWIN, AND J. PEIRÓ, *rp-adaptation for compressible flows*, *International Journal for Numerical Methods in Engineering*, 121 (2020), pp. 5405–5425.
- [25] J. MARCON, M. TURNER, D. MOXEY, S. J. SHERWIN, AND J. PEIRÓ, *A variational approach to high-order r-adaptation*, arXiv preprint arXiv:1901.00990, (2019).
- [26] *MFEM: Modular finite element methods [Software]*. mfem.org.
- [27] K. MITTAL, *Highly scalable solution of incompressible navier-stokes equations using the spectral element method with overlapping grids*, (2019).
- [28] K. MITTAL, S. DUTTA, AND P. FISCHER, *Nonconforming Schwarz-spectral element methods for incompressible flow*, *Computers & Fluids*, 191 (2019), p. 104237.
- [29] K. MITTAL AND P. FISCHER, *Mesh smoothing for the spectral element method*, *Journal of Scientific Computing*, 78 (2019), pp. 1152–1173.
- [30] R. MITTAL AND G. IACCARINO, *Immersed boundary*

- methods*, Annu. Rev. Fluid Mech., 37 (2005), pp. 239–261.
- [31] D. MOXEY, M. GREEN, S. SHERWIN, AND J. PEIRÓ, *An isoparametric approach to high-order curvilinear boundary-layer meshing*, Computer Methods in Applied Mechanics and Engineering, 283 (2015), pp. 636–650.
- [32] N. ODIER, A. THACKER, M. HARNIEH, G. STAFFELBACH, L. GICQUEL, F. DUCHAINE, N. G. ROSA, AND J.-D. MÜLLER, *A mesh adaptation strategy for complex wall-modeled turbomachinery LES*, Computers & Fluids, 214 (2021), p. 104766.
- [33] C. S. PESKIN, *The immersed boundary method*, Acta numerica, 11 (2002), pp. 479–517.
- [34] R. POYA, R. SEVILLA, AND A. J. GIL, *A unified approach for a posteriori high-order curved mesh generation using solid mechanics*, Computational Mechanics, 58 (2016), pp. 457–490.
- [35] A. A. REQUICHA AND H. B. VOELCKER, *Constructive solid geometry*, Production Automation Project, University of Rochester, (1977).
- [36] E. RUIZ-GIRONÉS AND X. ROCA, *Automatic penalty and degree continuation for parallel pre-conditioned mesh curving on virtual geometry*, Computer-Aided Design, 146 (2022), p. 103208.
- [37] E. RUIZ-GIRONÉS AND X. ROCA, *Automatic penalty and degree continuation for parallel pre-conditioned mesh curving on virtual geometry*, Computer-Aided Design, 146 (2022), p. 103208.
- [38] L. SCHNEIDERS, D. HARTMANN, M. MEINKE, AND W. SCHRÖDER, *An accurate moving boundary formulation in cut-cell methods*, Journal of Computational Physics, 235 (2013), pp. 786–809.
- [39] J. SOKOLOWSKI AND J.-P. ZOLÉSIO, *Introduction to shape optimization*, in Introduction to shape optimization, Springer, 1992, pp. 5–12.
- [40] M. SUSSMAN, P. SMEREKA, AND S. OSHER, *A level set approach for computing solutions to incompressible two-phase flow*, Journal of Computational Physics, 114 (1994), pp. 146–159.
- [41] T. TOULORGE, J. LAMBRECHTS, AND J.-F. REMACLE, *Optimizing the geometrical accuracy of curvilinear meshes*, Journal of Computational Physics, 310 (2016), pp. 361–380.
- [42] Z. Q. XIE, R. SEVILLA, O. HASSAN, AND K. MORGAN, *The generation of arbitrary order curved meshes for 3D finite element analysis*, Computational Mechanics, 51 (2013), pp. 361–374.
- [43] M. YANO AND D. L. DARMOFAL, *An optimization-based framework for anisotropic simplex mesh adaptation*, Journal of Computational Physics, 231 (2012), pp. 7626–7649.
- [44] R. ZHANG, A. JOHNEN, AND J.-F. REMACLE, *Curvilinear mesh adaptation*, in International meshing roundtable, Springer, 2018, pp. 57–69.