A Non-Uniform Refinement Approach for Solving Adjoint Problems in Functional Error Estimation and Mesh Adaptation

Brian N. Granzow[a,*], Assad A. Oberai[b], Mark S. Shephard[a]

[a] *Scientific Computation Research Center*
Rensselaer Polytechnic Institute
Troy, NY 12180
[b] *Department of Aerospace and Mechanical Engineering*
University of Southern California
Los Angeles, CA 90007

Abstract

Adjoint-based error estimation is used in finite element methods to effectively and accurately estimate discretization errors in physically meaningful output quantities. These estimates are obtained via the solution of an auxiliary adjoint problem. Obtaining an enriched representation of the solution to this adjoint problem is a necessary step in the process of adjoint-based error estimation. Solving the adjoint problem on a uniformly refined mesh is one possible way to obtain such an enriched adjoint representation. In this note, we propose a similar method for adjoint enrichment, whereby the adjoint problem is solved on a mesh obtained via non-uniform refinement. This leads to an adjoint problem with fewer degrees of freedom than the traditional uniform refinement approach. We propose two possible non-uniform refinement strategies and investigate resulting output error estimates for Poisson's equation in two dimensions and nonlinear elasticity in three dimensions.

Keywords: adjoint, a posteriori, functional, error estimation, refinement, finite element

1. Introduction

Adjoint-based error estimation [3, 9, 15, 19, 20, 21, 16, 17, 8, 6] is a tool used in numerical simulation to estimate the discretization error in physically meaningful output quantities. Combined with mesh adaptation, adjoint-based error estimation also provides the ability to control the discretization error. The process of adjoint-based error estimation relies on the introduction of an auxiliary *adjoint problem*, which is constructed using the solution to the original or *primal problem* of interest.

To obtain meaningful error estimates, the solution to the adjoint problem must be enriched in some manner. That is, it is necessary to obtain a representation of the adjoint solution in a richer space compared to the space used for the primal problem. Several strategies are commonly used to obtain an enriched adjoint representation. These approaches include solving the adjoint problem in a globally higher order polynomial space [7], solving the adjoint problem on a uniformly refined mesh [5], solving the adjoint problem in the same space as used for the primal problem and solving local patch-wise problems least squares problems [14] or performing patch-wise higher-order interpolation [3], and enriching the adjoint solution via variational multiscale methods [11].

Solving the adjoint problem in a globally higher order polynomial space or on a uniformly refined mesh is a computationally expensive proposition. On the other hand, solving the adjoint problem in the same space as used for the primal problem and enriching it via some local recovery operation may not be guaranteed

*Corresponding author, brian.granzow@gmail.com

to yield a more accurate adjoint solution. In this chapter, we propose a simple compromise and solve the adjoint problem on meshes obtained via non-uniform refinement.

The remainder of this chapter is structured as follows. First, we review adjoint-based error estimation for functional quantities using two discretization levels, a *coarse* space and a *fine* space. We then review three choices for the fine space obtained by refinement of the mesh used for the coarse space. The first choice is the standard uniform refinement method, while the other two approaches form the fine space via non-uniform refinement. In each of these sections, we discuss the algorithm utilized to generate the fine space. We then investigate these three adjoint enrichment approaches when applied to examples in Poisson's equation and conclude with a summary of the results.

2. Error Estimation with Two Levels

2.1. Error Estimates

Following Venditti and Darmofal [19, 20, 21], we review output-based error estimation using two discretization levels. Let $\mathcal{V}^h$ and $\mathcal{V}^H$ denote finite dimensional spaces such that $\mathcal{V}^H \subset \mathcal{V}^h$. We refer to $\mathcal{V}^h$ and V^H as the *fine* space and the *coarse* space, respectively. Let $\boldsymbol{R}^H : \mathbb{R}^N \to \mathbb{R}^N$ denote the system of (potentially nonlinear) algebraic equations arising from a finite element discretization of a PDE on the coarse space $\mathcal{V}^H$, such that the solution vector $\boldsymbol{u}^H \in \mathbb{R}^N$ satisfies

$$\boldsymbol{R}^H(\boldsymbol{u}^H) = 0. \tag{1}$$

Similarly, let $\boldsymbol{R}^h : \mathbb{R}^n \to \mathbb{R}^n$ denote the system of algebraic equations arising from a finite element discretization of the same PDE on the fine space $\mathcal{V}^h$, such that

$$\boldsymbol{R}^h(\boldsymbol{u}^h) = 0, \tag{2}$$

where $\boldsymbol{u}^h \in \mathbb{R}^n$ is the solution vector on the fine space and $n > N$.

Let $J^H : \mathbb{R}^N \to \mathbb{R}$ denote a discrete representation of a physically meaningful functional quantity on the coarse space $\mathcal{V}^H$, and similarly let $J^h : \mathbb{R}^n \to \mathbb{R}$ denote the functional approximated on the fine space $\mathcal{V}^h$. Let $\boldsymbol{u}_H^h := \boldsymbol{I}_H^h \boldsymbol{u}^H$ denote the prolongation of the coarse space solution $\boldsymbol{u}^H$ onto the fine space $\mathcal{V}^h$ via interpolation, where $\boldsymbol{I}_H^h : \mathcal{V}^H \to \mathcal{V}^h$.

The functional evaluated on the fine space $J(\boldsymbol{u}^h)$ can be expanded in a Taylor series approximation about the prolonged coarse space solution $\boldsymbol{u}_H^h$ as

$$J^h(\boldsymbol{u}^h) = J^h(\boldsymbol{u}_H^h) + \left[\left.\frac{\partial J^h}{\partial \boldsymbol{u}^h}\right|_{\boldsymbol{u}_H^h}\right](\boldsymbol{u}^h - \boldsymbol{u}_H^h) + \dots \tag{3}$$

Similarly, the residual system of equations evaluated on the fine space $\boldsymbol{R}^h(\boldsymbol{u}^h)$ can be expanded about the prolonged coarse space solution $\boldsymbol{u}_H^h$ as

$$\boldsymbol{R}^h(\boldsymbol{u}^h) = \boldsymbol{R}^h(\boldsymbol{u}_H^h) + \left[\left.\frac{\partial \boldsymbol{R}^h}{\partial \boldsymbol{u}^h}\right|_{\boldsymbol{u}_H^h}\right](\boldsymbol{u}^h - \boldsymbol{u}_H^h) + \dots \tag{4}$$

Using the governing relation (2) in the residual Taylor expansion (4) suggests a first order approximation for the discretization error between the spaces:

$$(\boldsymbol{u}^h - \boldsymbol{u}_H^h) \approx -\left[\left.\frac{\partial \boldsymbol{R}^h}{\partial \boldsymbol{u}^h}\right|_{\boldsymbol{u}_H^h}\right]^{-1} \boldsymbol{R}^h(\boldsymbol{u}_H^h). \tag{5}$$

Inserting the error approximation (5) into the functional Taylor expansion (3) suggests the error estimate:

$$J^h(\boldsymbol{u}^h) - J^h(\boldsymbol{u}_H^h) \approx -\left[\left.\frac{\partial J^h}{\partial \boldsymbol{u}^h}\right|_{\boldsymbol{u}_H^h}\right]\left[\left.\frac{\partial \boldsymbol{R}^h}{\partial \boldsymbol{u}^h}\right|_{\boldsymbol{u}_H^h}\right]^{-1} \boldsymbol{R}^h(\boldsymbol{u}_H^h), \tag{6}$$

which can be re-written in terms of an *adjoint* variable $\boldsymbol{z}^h$ as

$$J^h(\boldsymbol{u}^h) - J^h(\boldsymbol{u}_H^h) \approx -\boldsymbol{z}^h \cdot \boldsymbol{R}^h(\boldsymbol{u}_H^h), \tag{7}$$

where $\boldsymbol{z}^h \in \mathbb{R}^n$ is the solution to the so-called *adjoint problem* given by

$$\left[\frac{\partial \boldsymbol{R}^h}{\partial \boldsymbol{u}^h}\bigg|_{\boldsymbol{u}_H^h}\right]^T \boldsymbol{z}^h = -\left[\frac{\partial J^h}{\partial \boldsymbol{u}^h}\bigg|_{\boldsymbol{u}_H^h}\right]^T. \tag{8}$$

2.2. A Simple A-Priori Analysis

Consider that the functional of interest converges at the rate k, such that $J - J^h(\boldsymbol{u}_H^h) = cH^k$ and $J - J^h(\boldsymbol{u}^h) = ch^k$, where J is the exact value of the functional quantity of interest. Assume that the fine space is obtained via refinement of the coarse space. Consider the ratio

$$\frac{J^h(\boldsymbol{u}^h) - J^h(\boldsymbol{u}_H^h)}{J - J^h(\boldsymbol{u}_H^h)} \approx \frac{-\boldsymbol{z}^h \cdot \boldsymbol{R}^h(\boldsymbol{u}_H^h)}{J - J^h(\boldsymbol{u}_H^h)} \tag{9}$$

which, as $H \to 0$, will tend towards [8]

$$\alpha := 1 - \left(\frac{h}{H}\right)^k. \tag{10}$$

Let η denote our approximation to the functional error $J - J^h(\boldsymbol{u}_H^h)$. Let $\mathcal{I}$ denote the effectivity index given by

$$\mathcal{I} = \frac{\eta}{J - J^h(\boldsymbol{u}_H^h)}. \tag{11}$$

We would like error estimates $\mathcal{E}$ that lead to effectivity indices of $\mathcal{I} = 1$ as $H \to 0$. To achieve this, we scale the two-level adjoint weighted residual estimate (7) by the inverse of the factor α, such that

$$\eta = -\frac{1}{\alpha}\boldsymbol{z}^h \cdot \boldsymbol{R}^h(\boldsymbol{u}_{\boldsymbol{H}}^{\boldsymbol{h}}). \tag{12}$$

3. Choices for the Fine Space

3.1. Uniform Refinement

We first consider the traditional approach of using a uniformly refined mesh to solve the adjoint problem. We refer to this approach as the UNIF refinement approach. To perform uniform refinement, every edge in the mesh is marked for refinement. The algorithm for uniform refinement is given in Algorithm 1. Figure 1 demonstrates an example of the UNIF refinement approach applied to a base mesh. For the uniform refinement approach, we naturally choose the ration $\frac{h}{H} = \frac{1}{2}$, leading to the scaling parameter $\alpha = 1 - \left(\frac{1}{2}\right)^k$.

Algorithm 1 Uniform refinement algorithm

for each edge e in mesh M **do**
  mark edge e for refinement.
end for

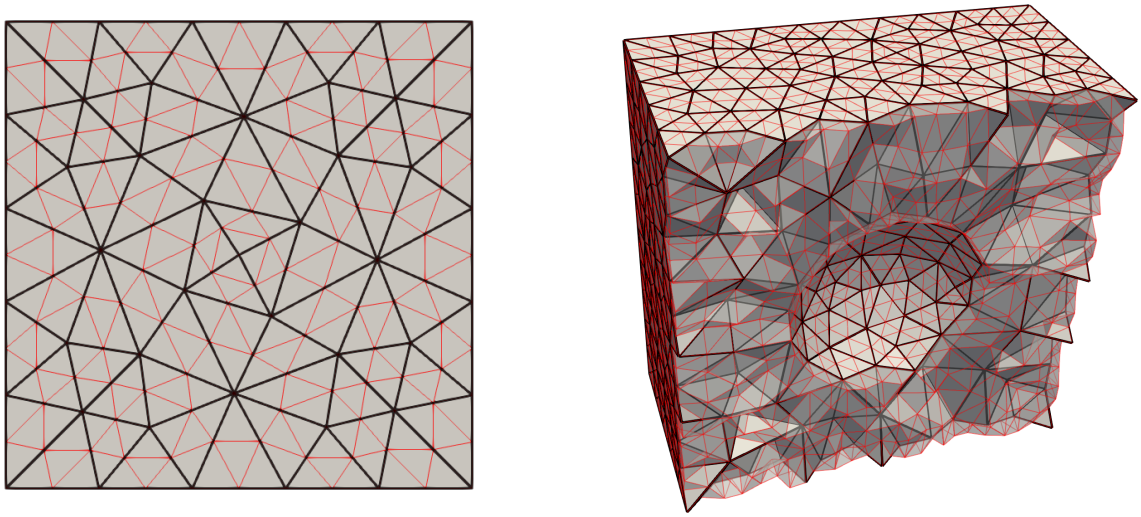

Figure 1: Edges of a base mesh (black) and a nested mesh refined with the Unif scheme (red) in two dimensions.

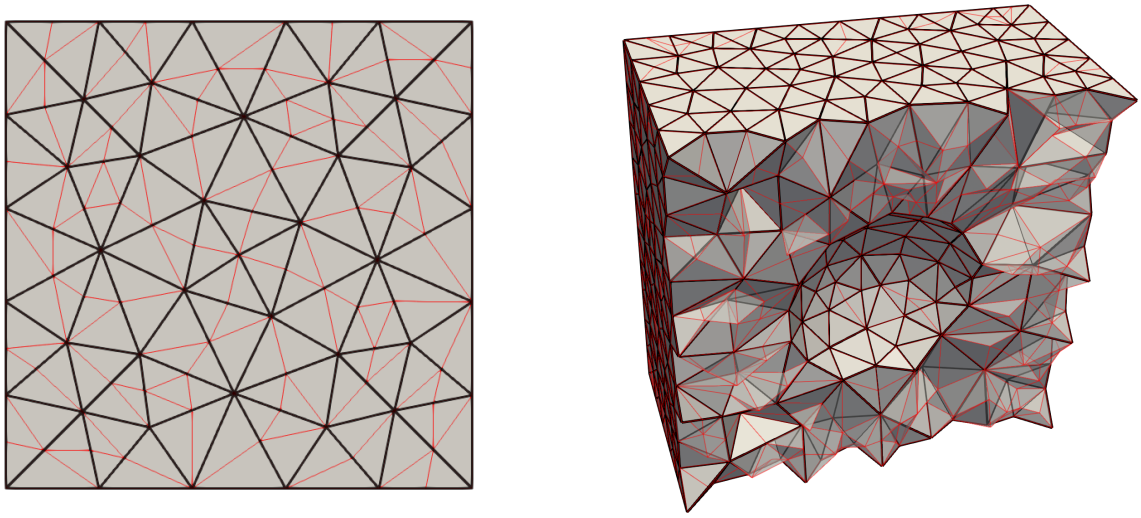

Figure 2: Edges of a base mesh (black) and a nested mesh refined with the Long scheme (red) in two dimensions.

3.2. Long Edge Refinement

Next, we consider an adaptive scheme that marks the longest edge in each element for refinement. We refer to this scheme as the Long edge refinement scheme. The LONG edge refinement algorithm is outlined in Algorithm 2. Figure 2 illustrates the LONG edge refinement algorithm applied to a base mesh.

Note that, for the LONG edge refinement approach, some elements are split once while others are split multiple times. It follows then that there is no single global ratio $\frac{h}{H}$ of the fine mesh size to the coarse mesh size. Presently, we approximate this ratio by taking the average of all ratios of nested element sizes to their parent element size, given by

$$\frac{h}{H} \approx \frac{1}{n_{el}} \sum_{e=1}^{n_{el}} \frac{h_e}{H_e}, \tag{13}$$

where n_{el} is the total number of elements in the nested mesh.

Algorithm 2 Long edge refinement algorithm

```
for each element el in mesh M do
    for each edge e in element el do
        if e is longest edge in el then
            mark edge e for refinement.
        end if
    end for
end for
```

3.3. Single Edge Refinement

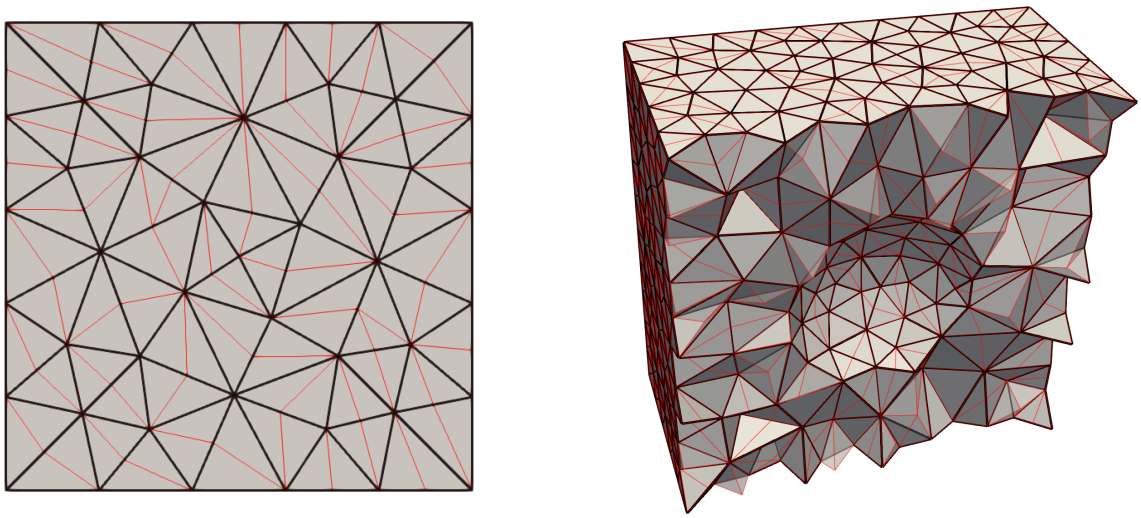

Figure 3: Edges of a base mesh (black) and a nested mesh refined with the Single scheme (red) in two dimensions.

Finally, we consider a cheap refinement alternative to uniform refinement that attempts to only mark a single edge in each element for refinement. We refer to this approach as the SINGLE edge refinement approach. To perform single edge refinement, a traversal of all edges in the mesh is performed. During this traversal, the first edge encountered is marked for refinement and the elements adjacent to that edge are tagged as 'visited'. As the edges in the mesh are traversed, each element adjacent to the edge is checked to see if it has already been encountered. If all adjacent elements have not been encountered, then the edge is marked for refinement. After this process has completed, some elements may be *isolated*, in that they have still not been marked as 'visited'. Thus, for each element remaining that has not been marked as 'visited', we mark the first edge adjacent to the element for refinement. The single edge refinement algorithm is illustrated in Algorithm 3. Figure 3 demonstrates a mesh resulting from the application of the single edge refinement scheme. For the SINGLE scheme, we again approximate the ratio $\frac{h}{H}$ with equation (13).

4. Mesh Adaptation

4.1. Error Localization

It is necessary to localize contributions to the total error η to mesh entity level *correction indicators* to drive mesh adaptation. For finite volume and discontinuous Galerkin finite element methods, it is common

Algorithm 3 Single edge refinement algorithm

```
initialize all elements to be 'not visited'.
for each edge e in mesh M do
    let S be the set elements adjacent to edge e.
    if each element in S is 'not visited' then
        mark edge e for refinement.
        for each element el in S do
            mark element el as 'visited'.
        end for
    end if
end for
for each element el in mesh M do
    if el is marked as 'not visited' then
        let S be the edges adjacent to element el
        mark the first edge e in S for refinement
    end if
end for
```

to consider the discrete element-level adjoint weighted residuals of the form $\boldsymbol{z}_e^h \cdot \boldsymbol{R}_e^h$, where the subscript e denotes evaluations over elements. However, for continuous finite elements, this approach does not account for systematic inter-element cancellation [8], which could lead to a sub-optimal adaptive strategy.

Traditonally, for continuous Galerkin finite element methods, the error is localized by integrating the residual by parts to recover strong form volumetric and jump contributions to the error over element interiors and boundaries, respectively. Presently, we utilize a localization strategy introduced by Richter and Wick [18] that proceeds by introducing a partition of unity ϕ_i, such that $\sum_i \phi_i = 1$, into the variational residual. In this localization, adjoint-weighted residual error information from neighboring elements is gathered to mesh vertices, leading to vertex-based correction indicators η_i, for $i = 1, 2, \ldots, n_{vtx}$. Here n_{vtx} denotes the number of vertices in the fine mesh. To obtain element-level correction indicators η_e, where $e = 1, 2, \ldots, n_{el}$, for the n_{el} elements in the space $\mathcal{V}^H$, we interpolate the vertex-based indicators η_i to element centers in the coarse mesh. While a full discussion of this localization procedure is outside of the scope of the present work, we refer readers to [18, 22] to demonstrate how this approach is utilized for Galerkin finite element methods and [10] to demonstrate how this approach is utilized for stabilized finite element methods.

4.2. Mesh Size Field

Once element-level correction indicators η_e have been obtained, we drive conforming mesh adaptation by specifying a *mesh size field*. For isotropic mesh adaptation, which we presently consider, this mesh size field defines the desired lengths of edges over the mesh. We utilize a mesh size field as described by Boussetta et al. [4] that attempts to equidistribute the error in an output adapted mesh with N target elements. From a high level, this size field will refine the mesh in areas of the domain that contribute strongly to the error in the functional and coarsen the mesh in areas of the domain that weakly contribute to the error in the functional.

Let p be the polynomial interpolant order for the chosen finite element method. In the subsequent results section, we consider only $p = 1$. We first define the global quantity G as

$$G = \sum_{e=1}^{n_{el}} (\eta_e)^{\frac{2d}{2p+d}}. \tag{14}$$

From this global quantity, we compute new element size H_e^{new} by scaling previous element sizes H_e according to the formula

$$H_e^{\text{new}} = \left(\frac{G}{N}\right)^{\frac{1}{d}} (\eta_e)^{\frac{-2}{2p+d}} H_e. \tag{15}$$

To ensure that mesh adaptation is being driven by accurate correction indicators and to prevent excessive coarsening and refinement in a single adaptive step, we additionally clamp new element sizes such that they are no smaller than one quarter and no greater than twice the previous element size,

$$\frac{1}{4} \leq \frac{H_e^{\text{new}}}{H_e} \leq 2. \tag{16}$$

Presently, we make use of the PUMI [12] software for mesh adaptation purposes. This software uses a sequence of edge splits, swaps, and collapses [13, 1] to locally modify the mesh to satisfy the input mesh size field.

5. Results

5.1. Effectivity Indices for Poisson's Equation

As a first example, we investigate the effectivity of the error estimate (12) for the model problem:

$$\begin{cases} -\nabla^2 u = f & \boldsymbol{x} \in \Omega, \\ u = 0 & \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{17}$$

when using the Unif, Long, and Single approaches to solve the adjoint problem (8). The model problem leads to the Galerkin finite element method: find $u^H \in \mathcal{V}^H$ such that

$$(\nabla w^H, \nabla u^H) = (w, f) \quad \forall w^H \in \mathcal{V}^H, \tag{18}$$

where $(w, u) := \int_\Omega wu \, \mathrm{d}\Omega$ denotes the L^2 inner product over the space $\mathcal{V}^H$, defined as

$$\mathcal{V}^H := \{u^h \in H^1(\Omega) : u^H = 0 \text{ on } \partial\Omega \,,\, u^H|_{\Omega_e} \in \mathbb{P}^1\}. \tag{19}$$

Here Ω_e denotes an element in a decomposition of the domain Ω into n_{el} non-overlapping elements such that $\cup_{e=1}^{n_{el}} \Omega_e = \Omega$ and $\Omega_i \cap \Omega_j = \varnothing$ if $i \neq j$. Additionally, $\mathbb{P}^1$ denotes the space of piecewise linear polynomials.

We choose the domain $\Omega = [0, 1] \times [0, 1]$ and the data to be $f = 2\pi^2 \sin(\pi x) \sin(\pi y)$ such that the exact solution is $u(x, y) = \sin(\pi x) \sin(\pi y)$. We choose the functional quantity to be $J(u) = \int_\Omega u \, \mathrm{d}\Omega$, which has the exact value $J(u) = \frac{4}{\pi^2}$. With the proposed finite element method, we expect the functional to converge at the rate $k = 2$, which we use to determine the scaling parameter α as given by equation (10).

The model problem was solved with mesh sizes $H = \{\frac{1}{5}, \frac{1}{10}, \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}\}$. For each chosen mesh size, the discrete adjoint problem (8) was solved on fine spaces $\mathcal{V}^h$ generated by the Unif, Long, and Single refinement schemes. An error estimate for the three schemes is then computed according to equation (12).

Figure 4 plots the effectivity index (11) for each of the three schemes at each chosen mesh size. Effectivity indices for the baseline Unif method approach 1 in the limit as $H \to 0$ as expected. The effectivity indices obtained using the two non-uniform refinement approaches, are less accurate and do not appear to be asymptotically correct. This is perhaps not surprising, as we have considered a bulk average for the ratio $\frac{h}{H}$ for these two schemes, as shown in Table 1.

H	Long : $\frac{h}{H}$	Single : $\frac{h}{H}$
$\frac{1}{5}$	0.6344	0.8198
$\frac{1}{10}$	0.6323	0.8212
$\frac{1}{20}$	0.6325	0.8204
$\frac{1}{40}$	0.6490	0.8183
$\frac{1}{80}$	0.6477	0.8202
$\frac{1}{160}$	0.6467	0.8195

Table 1: Approximated mesh size ratios for the Long and Single schemes for the first Poisson's equation example.

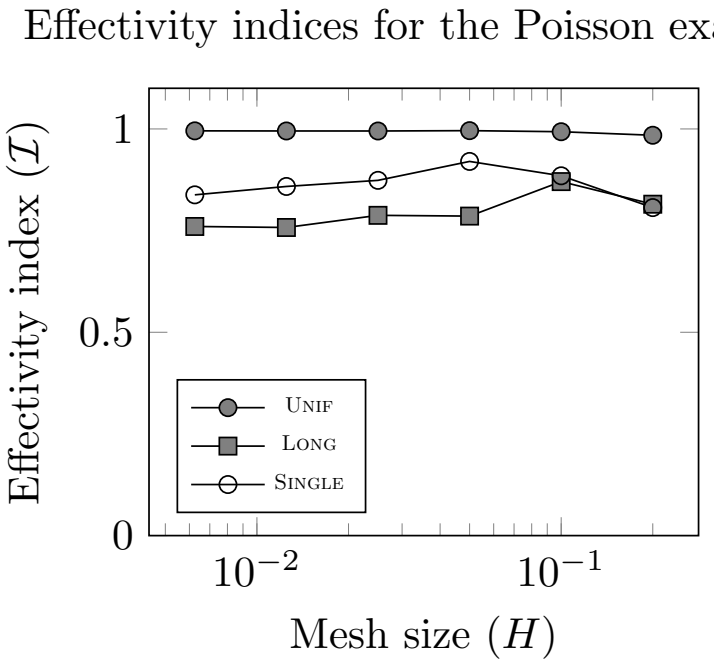

Figure 4: Effectivity indices using the Unif, Long, and Single refinement schemes for the Poisson example problem.

However, even though the error estimates themselves obtained by the Long and Single schemes may not be suitable for application purposes, these schemes may still be suitable to drive mesh adaptation at a cheaper cost than the full Unif approach. Figure 5 demonstrates the decrease in the total number of degrees of freedom for the adjoint problem for the Long and Single schemes as compared to the Unif scheme. This motivates us to consider an adaptive example for Poisson's equation in the next section.

5.2. Mesh Adaptation for Poisson's Equation

In this example, we again consider the governing equations for Poisson's equation, as given in the previous section. However, we now choose the forcing function f to be $f = 1$ and the domain $\Omega := [-1, 1] \times [-1, 1] \setminus [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$. Further, we consider the point-wise quantity of interest $J(u) = \int_\Omega \delta(\boldsymbol{x} - \boldsymbol{x}_0) u \, \mathrm{d}\Omega$, where the point of interest is chosen to be $\boldsymbol{x}_0 = (0.75, 0.75)$. We again expect the functional to converge at the rate $k = 2$. The domain and point-wise QoI location are show in Figure 6. The value of the quantity of interest was determined to have a value of $J(u) = 0.0334473 \pm 1e\text{-}7$ in the reference [2].

We performed the steps:

Solve Primal → Solve Adjoint → Estimate Error → Adapt Mesh

7 times, starting from the initial mesh shown in Figure 6. We solve the adjoint problem with three different methods on nested meshes obtained with the Unif, Long, and Single refinement schemes. At each adaptive step, the mesh size field was set according to equation (15), such that the target number of elements N is twice that of the current mesh.

Figure 7 illustrates the convergence history for the error $J(u) - J(u^H)$ for the three adaptive schemes obtained with the Unif (Goal Uniform), the Long (Goal Long), and the Single strategies, along with the error obtained by solving the primal problem with successively uniformly refined meshes (Uniform). The

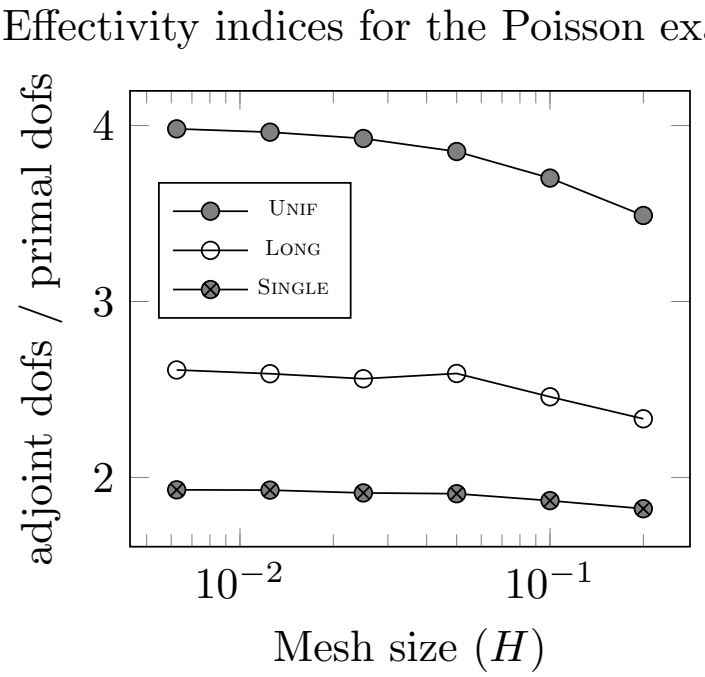

Figure 5: Ratio of adjoint problem degrees of freedom to primal problem degrees of freedom using the Unif, Long, and Single refinement schemes for the Poisson example problem.

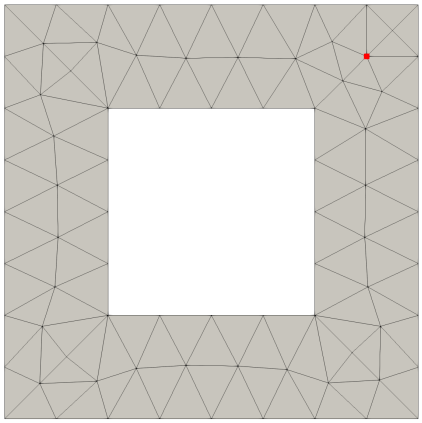

Figure 6: Geometry and initial mesh used for the second Poisson's equation example with the point of interest shown in red.

rate of convergence for the Unif scheme agrees with the reference [2]. Additionally, the error for both the Long and Single schemes converges at a rate almost near the Unif scheme.

Figure 8 illustrates the final adapted mesh obtained using the Unif strategy to solve the adjoint problem. The distribution of degrees of freedom in this mesh closely resemembles the results obtained in reference [2]. However, using the Long and Single to solve the adjoint problem results in final adapted meshes that appear to be largely unsuitable for application analysis, as shown in Figures 9 and 10, even thought these

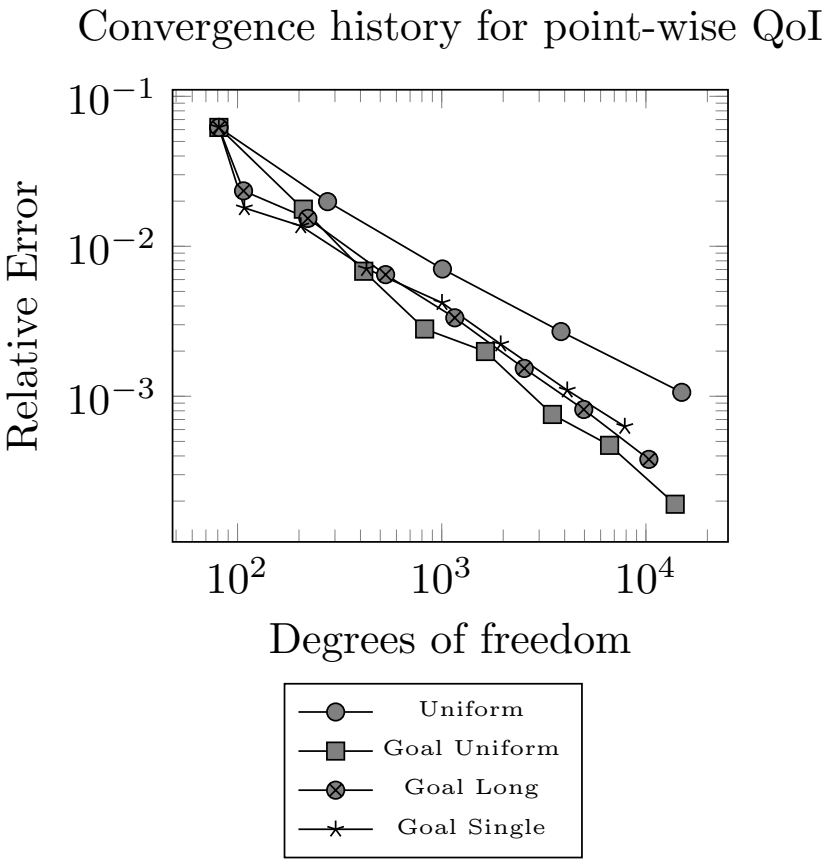

Figure 7: Error evolution for adaptive schemes for the second Poisson's equation example.

meshes result in more accurate functional evaluations as compared to uniform refinement.

6. Conclusions and Outlook

We have developed two alternative approaches to uniform refinement for performing enriched adjoint solves in adjoint-based error estimation with two discretization levels. We have applied this approach to Poisson's equation. While the number of degrees of freedom for the adjoint solve for these two alternative approaches decreases significantly when compared to the more traditional approach of solving the adjoint problem on a uniformly refined mesh, the present outlook indicates that these approaches are not yet suitable for practical applications. That is, when performing adjoint-based error estimation with the two novel approaches, effectivity indices are not asymptotically correct. Additionally, the meshes obtained with adaptive adjoint-based analysis display qualitatively different features when compared to the uniform refinement approach.

It is possible that more accurate error estimates could be obtained by considering the total functional error as the sum of element-level contributions

$$J^h(u^h) - J^h(u_H^h) \approx \sum_{e=1}^{n_{el}} -\frac{1}{\alpha_e} \boldsymbol{z}_e^h \cdot \boldsymbol{R}_e^h(\boldsymbol{u}_H^h), \tag{20}$$

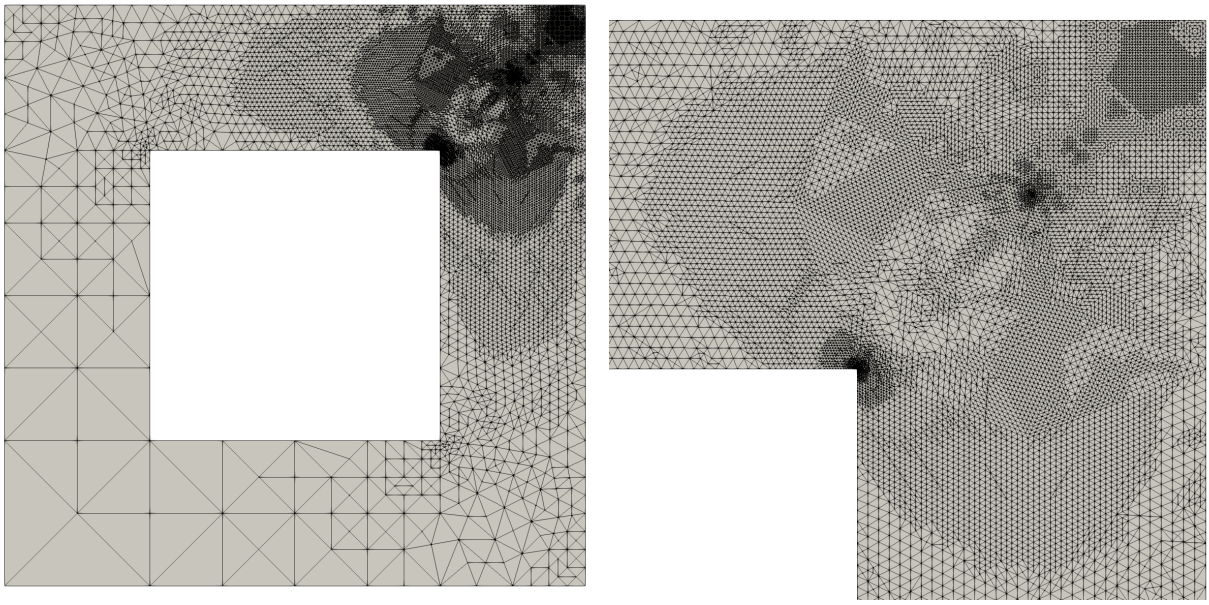

Figure 8: The final adapted mesh using the Unif strategy to solve the adjoint problem (left) and a close-up of the upper right-hand corner of this mesh (right).

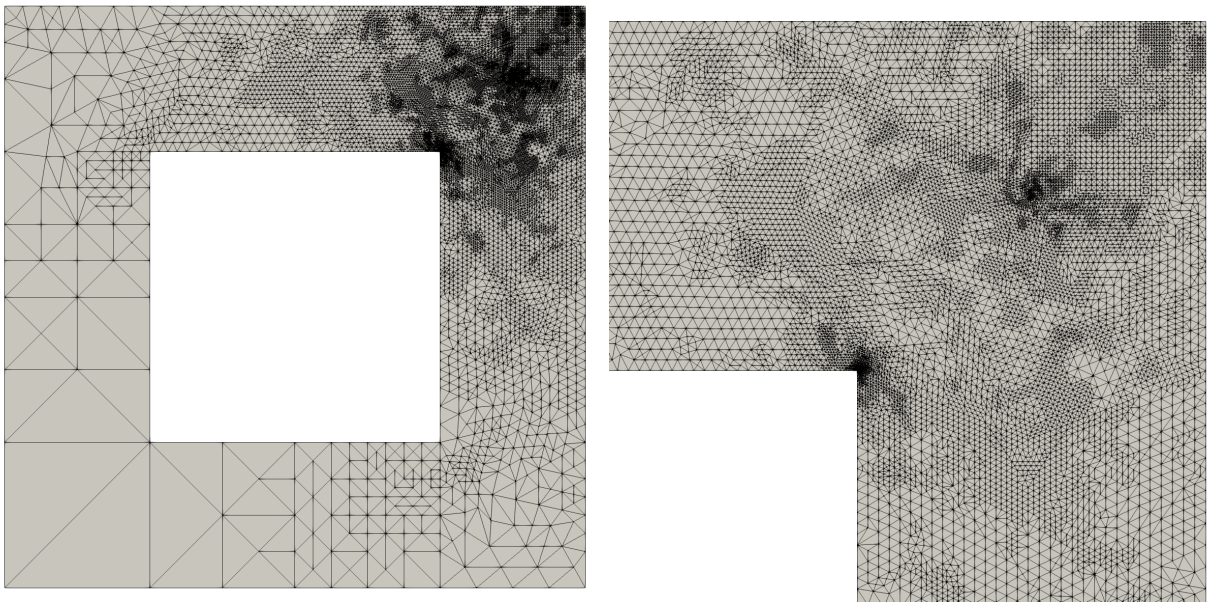

Figure 9: The final adapted mesh using the Long strategy to solve the adjoint problem (left) and a close-up of the upper right-hand corner of this mesh (right).

where we have replaced the approximated ratio (13) with the exact element-level ratio, $\alpha_e = 1 - \left(\frac{h_e}{H_e}\right)^k$. Here, the subscript e denotes the element-level contributions to the corresponding global quantity. Additionally, it is possible that more suitable meshes may be obtained during the adaptive process if a size field smoothing algorithm is utilized. We leave investigation into these areas as a suggestion for future work.

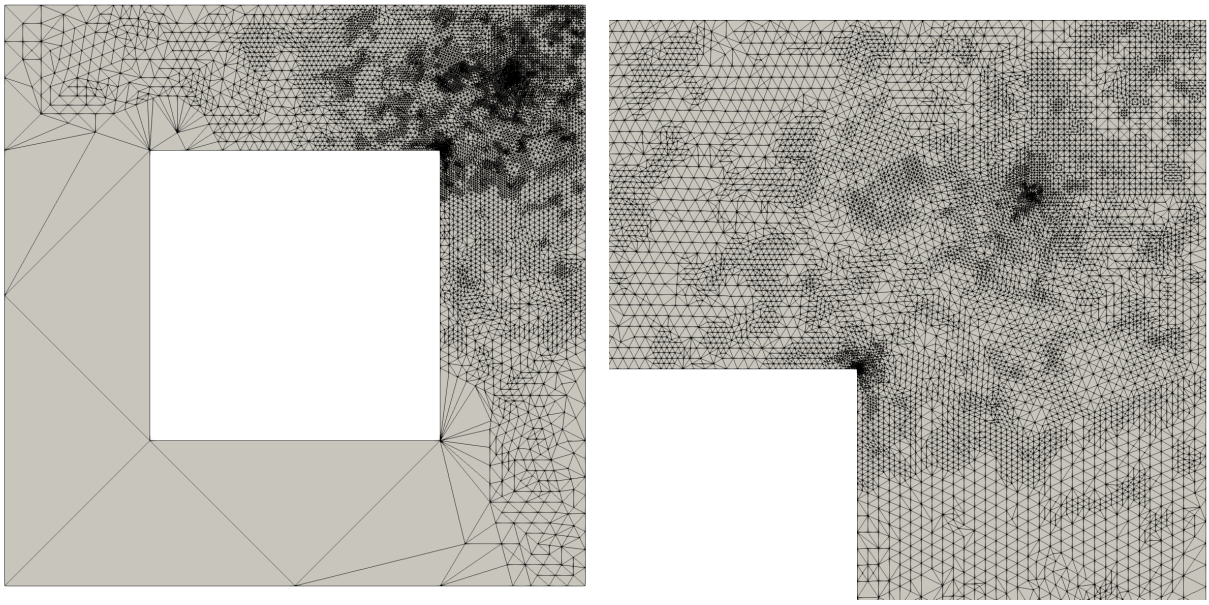

Figure 10: The final adapted mesh using the Single strategy to solve the adjoint problem (left) and a close-up of the upper right-hand corner of this mesh (right).

References

[1] Frédéric Alauzet, Xiangrong Li, E Seegyoung Seol, and Mark S Shephard. Parallel anisotropic 3D mesh adaptation by mesh modification. *Eng. with Comp.*, 21(3):247–258, Jan. 2006.

[2] Wolfgang Bangerth. Deal ii step 14, 2017.

[3] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, May. 2001.

[4] Ramzy Boussetta, Thierry Coupez, and Lionel Fourment. Adaptive remeshing based on a posteriori error estimation for forging simulation. *Comput. Methods in Appl. Mechanics and Eng.*, 195(48):6626–6645, Oct. 2006.

[5] Carsten Burstedde, Omar Ghattas, Georg Stadler, Tiankai Tu, and Lucas C Wilcox. Parallel scalable adjoint-based adaptive solution of variable-viscosity stokes flow problems. *Comput. Methods in Appl. Mechanics and Eng.*, 198(21):1691–1700, May. 2009.

[6] Jeffrey M Connors, Jeffrey W Banks, Jeffrey A Hittinger, and Carol S Woodward. A method to calculate numerical errors using adjoint error estimation for linear advection. *SIAM J. on Numerical Anal.*, 51(2):894–926, Mar. 2013.

[7] Krzysztof J Fidkowski. Output error estimation strategies for discontinuous galerkin discretizations of unsteady convection-dominated flows. *Int. J. for Numerical Methods in Eng.*, 88(12):1297–1322, May. 2011.

[8] Krzysztof J Fidkowski and David L Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA J.*, 49(4):673–694, Apr. 2011.

[9] Michael B. Giles and Niles A. Pierce. *Chapter 2 - Adjoint Error Correction for Integral Outputs*, pages 47–95. Springer, Berlin, Germany, 2016.

[10] Brian N Granzow, Assad A Oberai, and Mark S Shephard. Adjoint-based error estimation and mesh adaptation for stabilized finite deformation elasticity. submitted for publication.

[11] Brian N Granzow, Mark S Shephard, and Assad A Oberai. Output-based error estimation and mesh adaptation for variational multiscale methods. *Comput. Methods in Appl. Mechanics and Eng.*, 322:441–459, Aug. 2017.

[12] Daniel A Ibanez, E Seegyoung Seol, Cameron W Smith, and Mark S Shephard. PUMI: Parallel unstructured mesh infrastructure. *ACM Trans. on Math. Software*, 42(3):17–45, Jun. 2016.

[13] Xiangrong Li, Mark S Shephard, and Mark W Beall. 3D anisotropic mesh adaptation by mesh modification. *Comput. Methods in Appl. Mechanics and Eng.*, 194(48):4915–4950, Nov. 2005.

[14] Marian Nemec and Michael J Aftosmis. Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes. In *18th AIAA Computational Fluid Dynamics Conf., Miami, FL, USA*, Miami, FL, USA, Jun. 2007.

[15] Niles A Pierce and Michael B Giles. Adjoint and defect error bounding and correction for functional estimates. *J. of Comput. Physics*, 200(2):769–794, Nov. 2004.

[16] Serge Prudhomme and J Tinsley Oden. On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors. *Comput. Methods in Appl. Mechanics and Eng.*, 176(1-4):313–331, Jul. 1999.

[17] Serge Prudhomme, J Tinsley Oden, Tim Westermann, Jon Bass, and Mark E Botkin. Practical methods for a posteriori error estimation in engineering applications. *Int. J. for Numerical Methods. in Eng.*, 56(8):1193–1224, Jan. 2003.

[18] Thomas Richter and Thomas Wick. Variational localizations of the dual weighted residual estimator. *J. of Comput. and Appl. Math.*, 279:192–208, May. 2015.

[19] David A Venditti and David L Darmofal. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *J. of Comput. Physics*, 164(1):204–227, Oct. 2000.

[20] David A. Venditti and David L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *J. of Comput. Physics*, 176(1):40–69, Feb. 2002.

[21] David A. Venditti and David L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. *J. Comput. Phys.*, 187(1):22–46, May. 2003.

[22] Thomas Wick. Goal functional evaluations for phase-field fracture using PU-based DWR mesh adaptivity. *Comput. Mechanics*, 57(6):1017–1035, Mar. 2016.