SANDIA REPORT

SAND2014-17921
Unlimited Release
Printed September, 2014

Electromagnetic Extended Finite Elements for High-Fidelity Multimaterial Problems LDRD Final Report

Christopher M. Siefert, Pavel B. Bochev, Richard M. J. Kramer, Thomas E. Voth and James V. Cox

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-17921
Unlimited Release
Printed September, 2014

Electromagnetic Extended Finite Elements for High-Fidelity Multimaterial Problems LDRD Final Report

Christopher M. Siefert, Pavel B. Bochev, Richard M. J. Kramer,
Thomas E. Voth and James V. Cox

Abstract

Surface effects are critical to the accurate simulation of electromagnetics (EM) as current tends to concentrate near material surfaces. Sandia EM applications, which include exploding bridge wires for detonator design, electromagnetic launch of flyer plates for material testing and gun design, lightning blast-through for weapon safety, electromagnetic armor, and magnetic flux compression generators, all require accurate resolution of surface effects. These applications operate in a large deformation regime, where body-fitted meshes are impractical and multimaterial elements are the only feasible option. State-of-the-art methods use various mixture models to approximate the multi-physics of these elements. The empirical nature of these models can significantly compromise the accuracy of the simulation in this very important surface region. We propose to substantially improve the predictive capability of electromagnetic simulations by removing the need for empirical mixture models at material surfaces. We do this by developing an eXtended Finite Element Method (XFEM) and an associated Conformal Decomposition Finite Element Method (CDFEM) which satisfy the physically required compatibility conditions at material interfaces. We demonstrate the effectiveness of these methods for diffusion and diffusion-like problems on node, edge and face elements in 2D and 3D. We also present preliminary work on h-hierarchical elements and remap algorithms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many important physics problems have solutions which are strongly influenced by the effects of material interfaces. Electromagnetics, the simulation of which requires edge and face element discretizations, provided our primary motivating examples. Problems such as exploding bridge wires, electromagnetic flyer plates, lightning blast-through, electromagnetic armor and magnetic flux compression generators are good example of problems where surface effects matter. Solutions of these problems can and do develop strong and/or weak discontinuities in the solution field or its spatial derivatives. This creates well-documented problems for the convergence of finite element methods, unless one can ensure that the mesh is body fitted, i.e., the elements align with the interface [1]. However, the applications mentioned have large deformations, where body-fitted meshes are impractical, leaving multimaterial elements (and the associated degraded convergence) as the only feasible option.

The eXtended Finite Element Method (XFEM) provides one means by which these discontinuities can be dealt with, allowing the recovery of optimal finite element convergence [11, 3]. Typically a Heaviside or indicator function is used to "enrich" the basis, which works very nicely for strong discontinuities. Unfortunately, weak discontinuities require the addition of some type of constraint, or "tying." This adds the requirement of inf-sup stability of the discrete Lagrange multiplier space, or else the resulting field can suffer from instabilities and oscillations [16, 15]. This can be difficult to accomplish, but several approaches, including the method of vital vertices, have been able to successfully applied in practice [10, 4].

We have developed our own variant of the XFEM, XFEM with algebraic constraints (XFEM-AC) which has allowed us to side-step the inf-sup stability issue. We do this by proving the formal equivalence of XFEM-AC to a conformal decomposition finite element method (CDFEM) [12]. CDFEM enriches cut elements by decomposing them into interface-conforming sub-elements. Our XFEM-AC approach is applied to the full de Rahm complex (node, edges and faces) in 2D and 3D, and we develop a corresponding CDFEM method for cases where it has not previously been developed (edges and faces) [6, 7, 8]. We summarize these already published theoretical results in Chapter 2.

We have also considered an alternative means of discretization, an interface-conforming h-hierarchical element. These are different from the more typical p-hierarchical element, as the h-hierarchical element enriches the element with a more spatially refined element of the same order, while the p-hierarchical element adds higher order basis functions on

top of the same spatial extent. A slight modification to these elements allow us to make them interface-conforming, which suits the purpose of this study. This work is detailed in Chapter 3.

We also consider the issue of remap, or solution transfer, i.e., the mapping of fields from one mesh to a second mesh. This is necessary for the application of our methods in operator-split Lagrange-remap codes such as ALEGRA [13]. Here we generalize the approach of Bochev and Shashkov [2] to the remap problem for CDFEM discretizations, in 2-D for both nodal and edge-based fields. We detail this work in Chapter 4. Finally, we summarize the conclusions of this work and outline a roadmap for the future in Chapter 5.

Chapter 2

XFEM-AC and CDFEM for Nodes, Edges and Faces

Rather than reproducing our previous articles, [6], [7], and [8], in their entirety here, we instead summarize their major results, both theoretical and computational. Two key theorems will be stated, but their proofs will be omitted since they have been published elsewhere.

2.1 Governing Equations

We consider three sets of partial differential equations. These are intended to provide motivating examples of nodal, edge and face element discretizations.

For the nodal case, we consider

$$\tau \frac{\partial u}{\partial t} - \nabla \cdot \sigma \nabla u = 0, \tag{2.1}$$

where τ and σ are material parameters and u is the nodal solution field. This corresponds to the heat equation, or the Laplace equation (when $\tau = 0$).

For the edge case, we consider

$$\sigma \frac{\partial \mathbf{E}}{\partial t} + \nabla \times \nu \nabla \times \mathbf{E} = 0, \tag{2.2}$$

where σ and ν are the material parameters and $\mathbf{E}$ is the edge solution field. This corresponds to the $\mathbf{E}$-form of the eddy current Maxwell's equations.

In the face case, we consider

$$\nu \frac{\partial \mathbf{F}}{\partial t} - \nabla \lambda \nabla \cdot \mathbf{F} = 0, \tag{2.3}$$

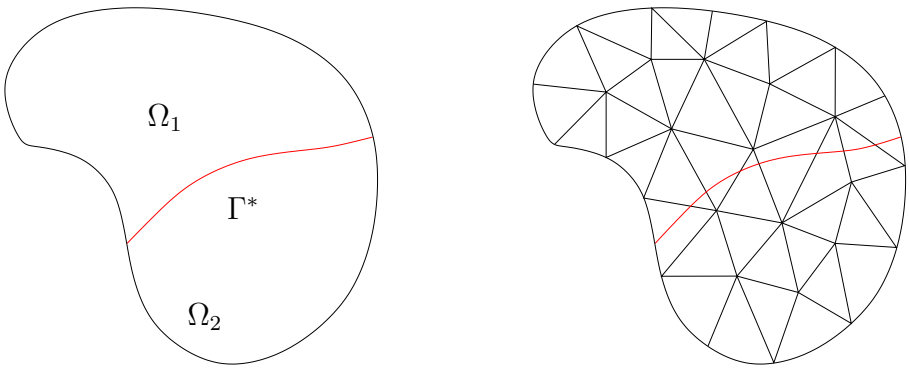
where ν and λ are the material parameters and $\mathbf{F}$ is the face solution field. This corresponds to the heat equation written in terms of heat fluxes (rather than temperatures).

2.2 Discretizations

We consider only lowest order discretizations on simplicies (e.g. triangles and tetrahedra) for the PDEs in Section 2.1. There are two primary motivations for this choice. First, lowest order simplicies have constant Jacobians, which means that a material interface which is a line (in 2D) or planar (in 3D) in physical space is also linear or planar in reference space. This means that the reference space can be used for constraints in our proposed XFEM method. The second reason for choosing lowest order simplicies is that edge and face bases functions on these elements have very special properties. On a simplex, edge basis functions have a *constant* tangential component on any line [7, 8] and face basis functions have a *constant* normal component on any plane [8]. These results are key to proving the equivalence of XFEM and CDFEM, which is summarized in Section 2.4.

2.3 The Interface Problem

We consider the case where the mesh is not fitted to a material boundary. Thus it must cut through one or more elements, as shown in Figure 2.1, which we reproduce from [6]. This interface Γ^* is assumed to be piecewise smooth and divide the domain Ω into two parts Ω_i, $i = 1, 2$. Different continuity conditions must be satisfied on Γ^* depending on the PDE. For (2.1), C^0 continuity must be maintained. For (2.2), $H(curl)$ or tangential continuity must be maintained. Finally, for (2.3), $H(div)$ or normal continuity must be maintained. Standard basis functions require more continuity than this on the immersed boundary, which leads to poor convergence and motivates our approach in the next section.



(a) Domain Ω (b) Discretization of Ω

Figure 2.1. (a) The embedded interface Γ^* divides Ω into subdomains Ω_1 and Ω_2. (b) The finite element partition $T(\Omega)$ of Ω, which does not conform to Γ^* [6].

2.4 Theoretical Justification for XFEM-AC and CDFEM

The key theoretical results of our work require some amount of notation. First, let $T \in T(\Omega)$ represent the set of (not necessarily interface conforming) elements in the domain Ω. Let $V(T)$ be the set of vertices on element T, $E(T)$ be the set of edges on element T and (in 3D only) $F(T)$ be the set of faces on element T. This allows us to define the standard finite element spaces

$$G_T^h := G^h\big|_T = \text{span}\{N_i \mid \boldsymbol{v}_i \in V(T)\}, \tag{2.4}$$
$$\mathbf{C}_T^h := \mathbf{C}^h\big|_T = \text{span}\{\mathbf{W}_i \mid \mathbf{e}_i \in E(T)\}, \tag{2.5}$$
$$\mathbf{D}_T^h := \mathbf{D}^h\big|_T = \text{span}\{\mathbf{V}_i \mid \mathbf{f}_i \in F(T)\}, \tag{2.6}$$

where the N_i's represent nodal basis functions, $\mathbf{W}_i$'s represents edge basis functions and $\mathbf{V}_i$'s represent face basis functions (in 3D only).

2.4.1 CDFEM conforming spaces

We can define the CDFEM, interface-conforming spaces by partitioning a cut element T into subelements T_ρ. We can then use these sub-elements to define conforming subspaces $\widehat{G}_T^h, \widehat{\mathbf{C}}_T^h$ and $\widehat{\mathbf{D}}_T^h$ much like the standard spaces above, but equal to the span of the basis functions on each sub-element.

2.4.2 XFEM-AC enriched spaces

For cut XFEM elements, we define the enriched spaces

$$\widetilde{G}_T^h := \bigcup_{T_\rho \in T} \text{span}\{\mathcal{H}_\rho N_i \mid \boldsymbol{v}_i \in V(T)\}, \tag{2.7}$$
$$\widetilde{\mathbf{C}}_T^h := \bigcup_{T_\rho \in T} \text{span}\{\mathcal{H}_\rho \mathbf{W}_i \mid \mathbf{e}_i \in E(T)\}, \tag{2.8}$$
$$\widetilde{\mathbf{D}}_T^h := \bigcup_{T_\rho \in T} \text{span}\{\mathcal{H}_\rho \mathbf{V}_i \mid \mathbf{f}_i \in F(T)\}, \tag{2.9}$$

where $\mathcal{H}_\rho$ is a Heaviside function that is one on subelement ρ and zero on all of the other subelements. These spaces do not enforce continuity across the cut.

2.4.3 Equivalence results

The first major result is that the CDFEM spaces are subspaces of the corresponding XFEM-AC spaces. Two and three dimensional versions of this theorem have been developed [6, 7, 8].

Theorem 2.4.1. *Let $\widehat{G}^h, \widehat{\mathbf{C}}^h, \widehat{\mathbf{D}}^h$ be the conforming spaces defined in Section 2.4.1. Let $\widetilde{G}^h, \widetilde{\mathbf{C}}^h, \widetilde{\mathbf{D}}^h$ be the enriched spaces defined in* (2.7)–(2.9)*. Then,*

$$\widehat{X}_T^h \subset \widetilde{X}_T^h, \quad \textit{for } X = G, \mathbf{C}, \mathbf{D}. \tag{2.10}$$

With this result in hand we develop a system of algebraic constraints which guarantees the equivalence of XFEM-AC and CDFEM. These constraints exploit the special properties of lowest order edge and face discretizations described in Section 2.2, though due to the tedious nature of their construction we will not detail them here. We will suffice it to say that constraints can be applied algebraically to generate systems of the form

$$\begin{bmatrix} \widetilde{A}_X & B_X^T \\ B_X & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_X^h \\ \lambda_X^h \end{bmatrix} = \begin{bmatrix} \mathbf{f}_X^h \\ 0 \end{bmatrix}, \tag{2.11}$$

where $X = \{G, \mathbf{C}, \mathbf{D}\}$, A_X is the mass-plus-stiffness matrix, and B_X is a constraint matrix. We are then able to define matrices Π_X and P_X s.t. the following is satisfied:

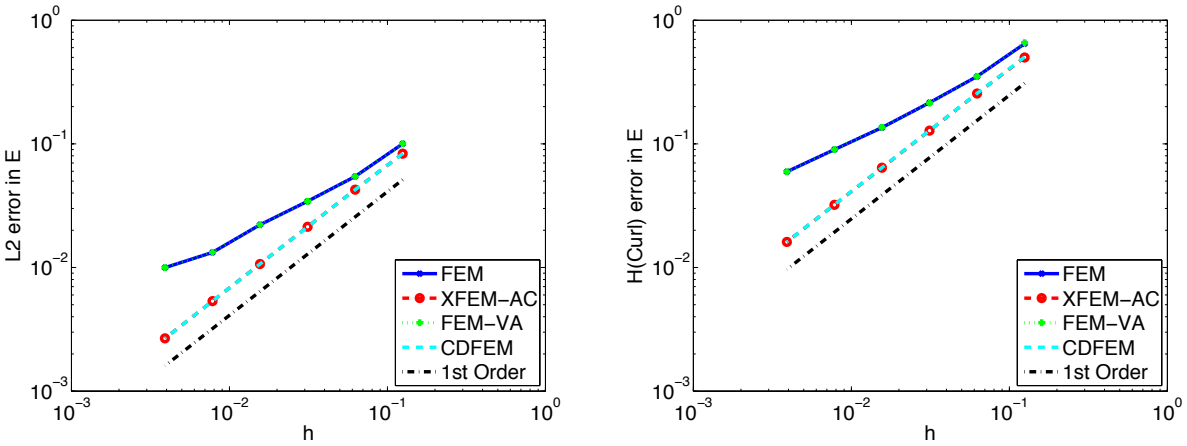
$$\Pi_X \widetilde{u} = P_X \widehat{u}, \tag{2.12}$$

where $\widetilde{u}$ is the XFEM-AC solution, $\widehat{u}$ is the CDFEM solution, and the matrices $P_X^T P_X$ and Π_X are invertible. This allows us to show the equivalence between the two methods for nodes, edges and faces.

2.5 Computational Examples

We have published numerous computational examples of the accuracy of our methods [6, 7, 8]. For the purpose of this report, we reproduce one representative plot for the 2D edge case [7]. Here we show convergence for the XFEM-AC, CDFEM and two variants of the standard finite element method (FEM and FEM-VA) for a 2D transient two material verification problem with solution

$$\begin{aligned} E_x(y,t) &= \begin{cases} \frac{x_c}{\pi}\cos(\pi y/x_c)e^{-t}, & x \le x_c \\ 4\frac{x_c}{\pi}\cos(\pi y/x_c)e^{-t}, & x > x_c \end{cases}, \\ E_y(x,t) &= \begin{cases} -\frac{2x_c}{\pi}\cos(\pi x/x_c)e^{-t}, & x \le x_c \\ -\frac{2x_c}{\pi}\cos(\pi(x+x_c)/2x_c)e^{-t}, & x > x_c \end{cases}. \end{aligned} \tag{2.13}$$



(a) $\|\mathbf{E} - \mathbf{E}_h^N\|_{L_2}$ (b) $\left|\mathbf{E} - \mathbf{E}_h^N\right|_{H(curl)}$

Figure 2.2. Convergence plots for the transient two material problem (2.13) [7].

We consider convergence in the $L^2(\Omega)$ norm as well as the $H(curl, \Omega)$ semi-norm. Lowest order edge elements are first order accurate for problems without material discontinuities, and we would like to recover that accuracy in the two material case. Figure 2.2 shows the accuracy of these four methods with respect to mesh convergence. The problem is solved to a time $t = 0.02$ over a number of timesteps. We see that the XFEM-AC and CDFEM methods convergence identically at the expected first order, while the FEM and FEM-VA methods converge at reduced accuracy. Results of this sort are typical in our work.

Chapter 3

Hierarchical Basis Functions

3.1 Introduction

This work presents a limited study seeking to overcome the ill-conditioning that can occur when a material interface is modeled explicitly, and a small sliver element (for a conforming FEM approach) or sub-element (for non-conforming approaches like XFEM or hierarchical bases) results. The initial emphasis of the work addressed scaling for a hierarchical basis approach. We sought to determine scaling factors for the added hierarchical basis functions to eliminate the ill-conditioning that can occur when a material interface results in a small sub-element. We examined an algebraic approach to this for one-dimensional and two-dimensional problems. The two-dimensional case raised several questions about the applicability of the approach in higher dimensions, so we decided to examine the problem in the context of a conforming FEM approach; we show how the condition number for the sliver element alone varied with geometry, analytically derive the perfect diagonal scaling for particular geometry case, attempt to use some schemes for combining diagonal scaling of elements to obtain a diagonal preconditioner for the global system.

3.2 One-dimensional nodal

The equation of interest is 1D static conduction,

$$\frac{\partial}{\partial x}\kappa\frac{\partial T}{\partial x} = 0 \tag{3.1}$$

$$T(x=0)=0,\ T(x=1)=1 \tag{3.2}$$

which after conversion to Galerkin weak form and subsequent discretization yields,

$$\mathbf{Ku}=\mathbf{f} \tag{3.3}$$

where linear basis functions,

$$N_1^e(\eta)=(1-\eta)/2,\text{ and } N_2^e(\eta)=(1+\eta)/2 \tag{3.4}$$

are assumed. With this discretization the element stiffness matrices are,

$$\mathbf{k}^e = \frac{\kappa}{2\Delta x}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.5}$$

from which the global $\mathbf{K}$ of (3.3) is assembled as usual. Note that (3.5) assumes κ is constant in the element. If we assume that κ is piecewise constant in an element with left and right-hand-sides having values κ_l and κ_r respectively, then (3.5) becomes,

$$\mathbf{k}^e = \frac{\overline{\kappa}}{2\Delta x}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.6}$$

where,

$$\overline{\kappa} = (1-\eta_c)\kappa_r + (1+\eta_c)\kappa_l. \tag{3.7}$$

Hierarchical Enrichment: A hierarchical basis may be introduced into the cut element to better resolve the solution discontinuity in the cut element. This basis is given by,

$$M(\eta) = \begin{cases} 1-\eta & \text{where} \quad -1 \le \eta < \eta_c \\ 1+\eta & \text{where} \quad \eta_c \le \eta < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Incorporating this basis with the standard linear basis yields the following "enriched" element stiffness matrix,

$$\mathbf{k}^e_{hb} = \frac{1}{\Delta x}\begin{bmatrix} \overline{\kappa} & -\overline{\kappa} & s(\kappa_r - \kappa_l) \\ -\overline{\kappa} & \overline{\kappa} & s(\kappa_l - \kappa_r) \\ s(\kappa_r - \kappa_l) & s(\kappa_l - \kappa_r) & 2s^2\frac{(1+\eta_c)\kappa_r + (1-\eta_c)\kappa_l}{1-\eta_c^2} \end{bmatrix}. \tag{3.9}$$

(3.9) implies an additional degree-of-freedom (DOF 3 on the element) where the field is reconstructed as,

$$u^h(x) = \sum_{I=1}^{2} N_I(\eta)u^0_I + sM(\eta)u^1. \tag{3.10}$$

and s is a scaling parameter as introduced by Soghrati et al. [17, 18].

In [17] and later in [18] the scaling parameter is defined as,

$$s = 4\zeta^2 \tag{3.11}$$

where,

$$\zeta = \frac{min(||x_1 - x_c||, ||x_2 - x_c||)}{||x_2 - x_1||}. \tag{3.12}$$

Note that (3.12) may be rewritten more compactly as,

$$\zeta = min((1 + \eta_c)), (1 - \eta_c))/2 \tag{3.13}$$

where η_c is the location of the cut in master space.

(3.11) is shown in Figure 3.1. Also shown is a scaling parameter which is suggested by (3.9), that is,

$$s = \sqrt{1 - \eta_c^2} \tag{3.14}$$

Note that this form of s is suggested in Figure (5) of [17] but not their equation. It is unclear which form was used in [17] and so we test the impact of both on condition number. Interestingly, [18] restates (3.11) without modification.

Plots of estimated condition number for the fully assembled $\mathbf{K}$ are shown in Figure 3.2 for $s = 1$ (e.g. unscaled). For comparison, the condition number of the unenriched FEM solution is also plotted (red line). It is clear from this figure that regardless of whether the hierarchical basis is included, the condition number scales as the square of $1/\Delta x$. It is also apparent that the impact of decreasing cut size (corresponding to larger η_c) increases the condition number for a given overall mesh-resolution.

Estimated condition number plots for the scaling of (3.11) are shown in Figure 3.3 and demonstrate the poor performance of this approach. Again, it is somewhat unclear as to whether this is actually the form of s Soghrati et al. used. Estimated condition number plots with scaling of (3.14) are shown in Figure 3.4. Excellent results are seen with the cut location, η_c, having negligible impact on stiffness condition number.

3.3 Two-dimensional nodal triangles

For the above one-dimensional case, a single enriched node is contained within the element (*i.e.*, not shared with adjacent elements), and as such the algebra is fairly straightforward.

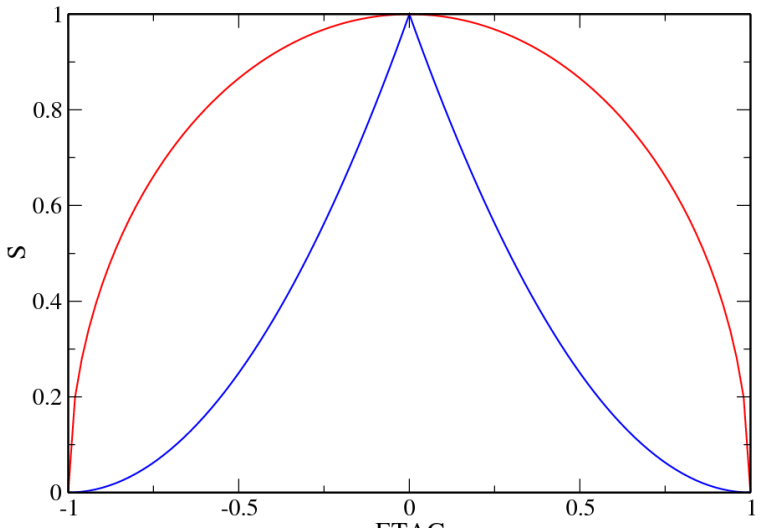


Figure 3.1. Scaling parameter of Eqns. (3.11) and (3.14) in blue and red respectively.

To determine the scaling for the enriched node, we simply set the scaling such that the "hierarchical-hierarchical partition" (a single term in this case) of the stiffness matrix remains bounded as the internal node approaches either of the nodes of the element. As shown this approach eliminated the ill-conditioning that can occur with a material cut occurring close to an element node.

For higher dimension problems the "plot thickens," because the material interface requires multiple enriched nodes to represent the discontinuity and these nodes are generally shared between elements. As the next step in complexity, we consider a two-dimensional problem. This work continues where the previous effort left off in terms of considering a scaling for the two-dimensional static conduction equation given by

$$\nabla \cdot \kappa \nabla T = 0 \tag{3.15}$$

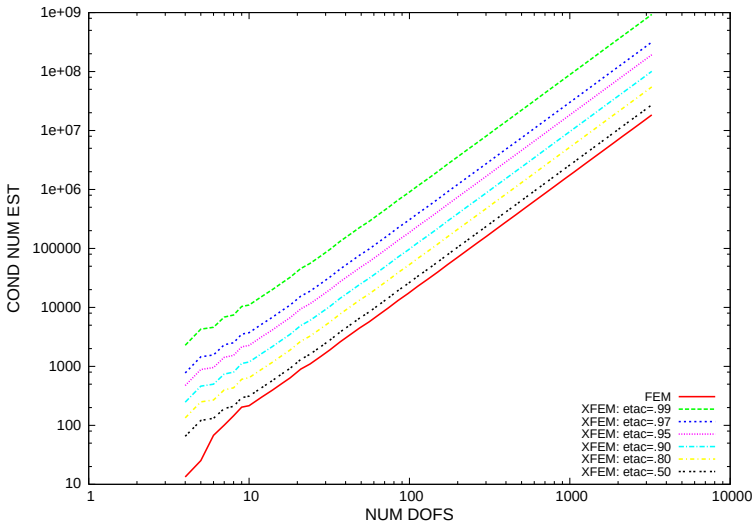


Figure 3.2. Stiffness condition number as a function of mesh resolution for a range of cut locations, η_c.

assuming appropriate boundary conditions. Following standard discretization techniques yields the Bubnov-Galerkin form,

$$\mathbf{Ku} = \mathbf{f} \tag{3.16}$$

where linear triangle element basis functions,

$$N_1^e(r,s) = 1 - r - s,\ N_2^e(r,s) = r, \text{ and } N_3^e(r,s) = s \tag{3.17}$$

are assumed and r and s are the master-coordinates as in Figure (3.5). With this discretization the element stiffness matrix is,

$$\mathbf{K}^e = \frac{\kappa}{2|J|}\begin{bmatrix} y_{32}^2 + x_{32}^2 & -(y_{31}y_{32} + x_{31}x_{32}) & y_{21}y_{32} + x_{21}x_{32} \\ -(y_{31}y_{32} + x_{31}x_{32}) & y_{31}^2 + x_{31}^2 & -(x_{21}x_{31} + y_{21}y_{31}) \\ y_{21}y_{32} + x_{21}x_{32} & -(x_{21}x_{31} + y_{21}y_{31}) & x_{21}^2 + y_{21}^2 \end{bmatrix} \tag{3.18}$$

where $|J| = x_{21}y_{31} - x_{31}y_{21}$ and $x_{IJ} = x_I - x_J$ and we have assumed that κ is a scalar (isotropic conduction) and constant over the element. In the above, $x_{IJ} = x_I - x_J$ where I and J are the coarse-level local node numbers shown in Fig. (3.5).

3.3.1 Material Interface

Here we assume that the coarse-scale triangle element contains two materials that are separated by a discrete interface (i.e. they are not mixed). The three possible configurations

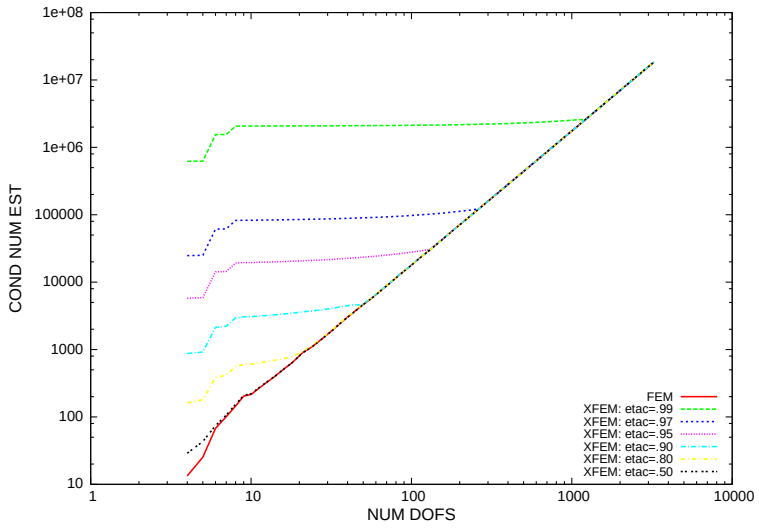


Figure 3.3. Stiffness condition number as a function of mesh resolution for a range of cut locations, η_c. Element stiffness matrices are scaled using (3.11).

(configurations 1, 2 and 4) are shown in Figure 3.5. Configuration 1 consists of subtriangle A_1 having $\kappa = \kappa_2$ and the remaining $\kappa = \kappa_1$. Configurations 2 and 4 follow in the obvious fashion.

The element stiffness for configurations 1, 2 and 4 are given by (3.18) but with the uniform κ replaced by $\overline{\kappa}_C$ computed as,

$$\overline{\kappa}_1 = ((\kappa_2 - \kappa_1)(1-\alpha)\gamma + \kappa_1 \tag{3.19}$$

$$\overline{\kappa}_2 = ((\kappa_2 - \kappa_1)(1-\beta)\alpha + \kappa_1 \tag{3.20}$$

or

$$\overline{\kappa}_4 = ((\kappa_2 - \kappa_1)(1-\gamma)\beta + \kappa_1 \tag{3.21}$$

and $C = 1$, 2 or 4 is the cut configuration number corresponding to Figure 3.5.

3.3.2 Hierarchical Enrichment

Here we add hierarchical bases to the one material element of Section 3.3. The enrichment basis are the local assembly of the mid-side nodes associated with a fine-scale tri discretization of the coarse-scale tri. In the case of Figure 3.5 the fine-scale bases are associated with

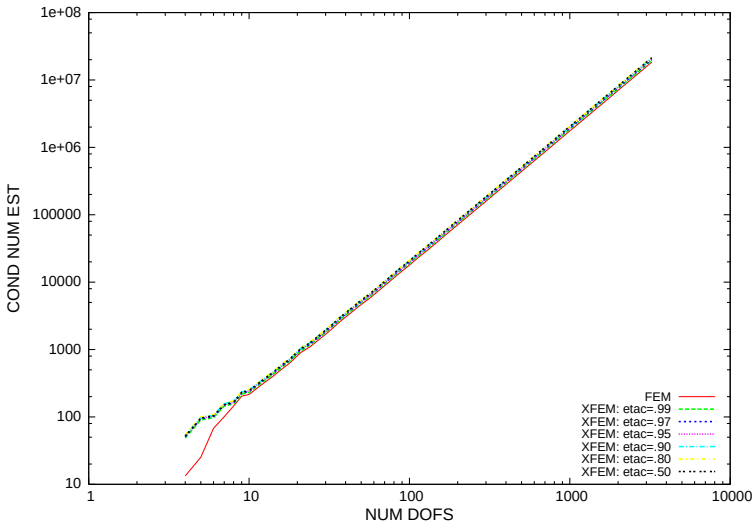


Figure 3.4. Stiffness condition number as a function of mesh resolution for a range of cut locations, η_c. Element stiffness matrices are scaled using (3.14).

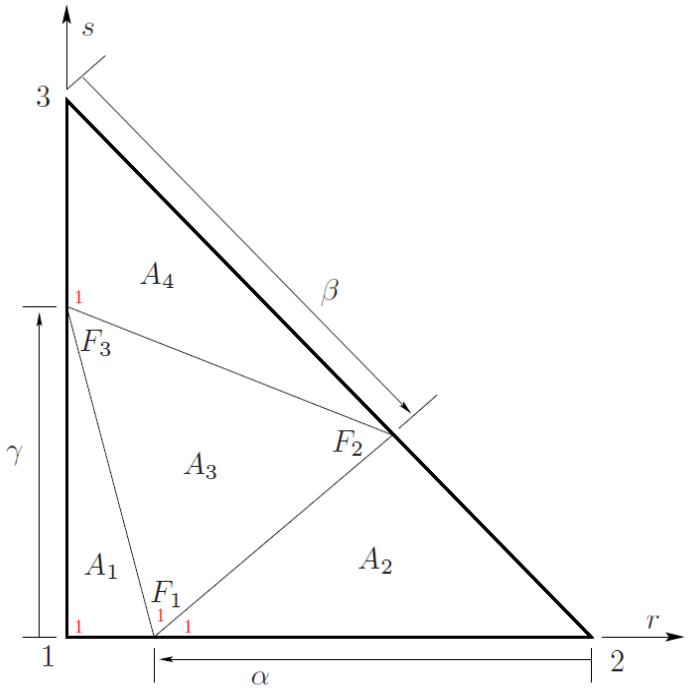


Figure 3.5. Three possible configurations for two material element. Cut location is indicated as 1, 2 or 4 where A_1, A_2, or A_4 (respectively) contains material 2. The remainder of the element is filled with material 1.

"nodes" F_1, F_2 and F_3 and are computed as,

$$M_1(u,v) = N_2^{e_1}(u,v) + N_1^{e_2}(u,v) + N_1^{e_3}(u,v), \tag{3.22}$$

$$M_2(u,v) = N_3^{e_2}(u,v) + N_2^{e_3}(u,v) + N_2^{e_4}(u,v), \tag{3.23}$$

and

$$M_3(u,v) = N_3^{e_1}(u,v) + N_3^{e_3}(u,v) + N_3^{e_4}(u,v) \tag{3.24}$$

respectively. In the above u and v are the local coordinates for the fine-scale bases which are in fact assemblages of the associated fine-scale tri local bases. Local basis number is indicated in Figure 3.5 with local bases beginning at 1 (indicated in red) and numbered consecutively in counter-clockwise order. Fine-scale node coordinates are parameterized as a convex combination of their coarse-scale face end-points as illustrated in the figure.

With the addition of the fine-scale bases the standard bases for the coarse tri we have that the scalar field may be approximated as,

$$u^h(x,y) = \sum_{I=1}^{3} N_I(r,s)u_I^0 + \sum_{J=1}^{3} s_J M_J(u,v)u_J^1. \tag{3.25}$$

where s_J is a scaling-parameter associated with basis M_J. Substituting (3.25) into the Galerkin weak form yields the element stiffness matrix,

$$\mathbf{K}^{e,hb} = \begin{bmatrix} \mathbf{K}^{e,00} & \mathbf{K}^{e,01} \\ \mathbf{K}^{e,10} & \mathbf{K}^{e,11} \end{bmatrix}. \tag{3.26}$$

In the above, $\mathbf{K}^{e,00}$ is identically $\mathbf{K}^e$ of (3.18) for constant κ over the coarse element. When κ is non-constant, as when configurations 1, 2 or 4 of Figure 3.5 are encountered, (3.18) is still valid but with κ replaced by the appropriate $\overline{\kappa}_C$ of (3.20)—(3.21) as in Section 3.3.1. Hence, for the hierarchical tri, adding the fine-scale bases does not change the coarse scale stiffness.

Remark 3.3.1. *Note that when non-uniform κ are encountered in the hierarchical formulation, both the fine and coarse stiffness matrices must be evaluated with varying κ. Specifically, you are not allowed to choose a constant κ for the coarse scale and expect the fine-scale to pick-up the appropriate variability of the material property.*

3.3.3 Algebraic Scaling

As for the one-dimensional case, we start by trying to define the scaling to reduce the effect of the "hierarchical-hierarchical partition" of the stiffness matrix ($\mathbf{K}^{e,11}$) upon the ill-conditioning. Again we initially take an algebraic approach, determining the scaling factors so that they eliminate singular stiffness terms in $\mathbf{K}^{e,11}$. As for the simpler one-dimensional case, the approach is rational in that we expect the condition number of the matrix to increase if a stiffness term can be unbounded. Unfortunately in this case, the complexity of the approach increases.

The expression for $\mathbf{K}^{e,11}$ is complex, but the general form of i-j^{th} term is as follows:

$$K_{ij}^{e,11} = s_i s_j \frac{\kappa \; f_{ij}\left(\mathbf{x},\mathbf{y},\alpha,\beta,\gamma\right)}{|J| \;\; g_{ij}\left(\alpha,\beta,\gamma\right)} \tag{3.27}$$

where (**x**,**y**) denotes the vectors of coarse-scale nodal coordinates. We seek to define the scaling factors such that they eliminate the singularities that can occur in terms when *edge intersect coordinates* (α, β, γ) approach 0 or 1; for these cases, the material intersect with the element produces a *sliver* or *chip* sub-element. As implied in Figure 3.5, for the sliver case (*e.g.*, letting $\alpha \to 0$ while $\beta = 1/2$) the sub-element has extreme aspect ratios, and for the chip case (*e.g.*, letting $\alpha = (1 - \beta) \to 0$) the sub-element is simply much smaller than the parent element.

Symbolic manipulation was used in this study to examine the form of the stiffness terms that result from the hierarchical element approach. As a preliminary means of determining the needed scaling, we considered the form of scaling factors required to cancel zeros in the $g_{ij,}$ *denominator factors*. This is neither a necessary nor sufficient condition for the scaling factors but useful in determining possible scaling forms. It is not necessary because sometimes in the limit terms from f_{ij} can cancel those in g_{ij}, and it is not sufficient because in some cases the scaling (in the limit) can zero rows and columns in $\mathbf{K}^{e,11}$. While this later case might produce a scaled stiffness that is less sensitive, in terms of ill-conditioning, it is still preferable for $\mathbf{K}^{e,11}$ to not lose rank (*i.e.*, not trade one problem for another). After determining possible forms for the scaling factors, they can be tested initially to determine their effects upon the terms in $\mathbf{K}^{e,11}$ for limiting cases, and ultimately by determining their effects upon the condition number of $\mathbf{K}^{e}$ for limiting cases.

To determine the forms of the scaling factors we first examine cases when one edge intersect coordinate is approaching 0 or 1 (*i.e.*, one edge node is approaching a coarse element node). For the non-critical values of the other edge intersect coordinates, let their values be one half for illustration. For one of these non-critical values, we assume that it is defined by the other intersect of the material interface but sufficiently far from the nodes (at 0 or 1). For the second non-critical value, the edge coordinate is arbitrary and is simply defined to partition the remaining quadrilateral (from the material cut) into triangles. Consider the forms of the denominator factors (g) of specific stiffness terms for three cases:

1. α=β=1/2, $K_{33}^{e,11}$ denominator (all other denominator factors nonzero)
   $g_{33}(\alpha, \beta, \gamma) = \gamma(\gamma - 1)$
2. α=γ=1/2, $K_{22}^{e,11}$denominator (all other denominator factors nonzero)
   $g_{22}(\alpha, \beta, \gamma) = \beta(\beta - 1)$
3. β=γ=1/2, $K_{11}^{e,11}$denominator (all other denominator factors nonzero)
   $g_{11}(\alpha, \beta, \gamma) = \alpha(\alpha - 1)$

For these three respective cases, the scaling factors that would cancel the denominator factors are:

$$(1) \quad s_3 = \sqrt{\gamma(1-\gamma)} \tag{3.28}$$

$$(2) \quad s_2 = \sqrt{\beta(1-\beta)} \tag{3.29}$$

$$(3) \quad s_1 = \sqrt{\alpha(1-\alpha)} \tag{3.30}$$

The remaining two scale factors for each case can be taken as unity.

Now consider the cases when two edge intersect coordinates are approaching 0 or 1 (*i.e.*, the material interface is close to one or two nodes). Consider the forms of the denominator factors (g) of specific stiffness terms for the three cases:

1. α=1/2, denominator factors for each term in $\mathbf{K}^{e,11}$
$$[g_{ij}(\alpha,\beta,\gamma)] = [1-\beta-\gamma+2\beta\gamma]\begin{bmatrix} 1 & 1 & 1 \\ 1 & \beta(\beta-1) & 1 \\ 1 & 1 & \gamma(\gamma-1) \end{bmatrix}$$

2. β=1/2, denominator factors for each term in $\mathbf{K}^{e,11}$
$$[g_{ij}(\alpha,\beta,\gamma)] = [1-\alpha-\gamma+2\alpha\gamma]\begin{bmatrix} \alpha(\alpha-1) & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & \gamma(\gamma-1) \end{bmatrix}$$

3. γ=1/2, denominator factors for each term in $\mathbf{K}^{e,11}$
$$[g_{ij}(\alpha,\beta,\gamma)] = [1-\alpha-\beta+2\alpha\beta]\begin{bmatrix} \alpha(\alpha-1) & 1 & 1 \\ 1 & \beta(\beta-1) & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Note that alternative forms to the multiplicative terms of the denominator factors are helpful in determining the limiting cases. For example,

$$1-\beta-\gamma+2\beta\gamma = (1-\beta)+\gamma(2\beta-1) = (1-\gamma)+\beta(2\gamma-1)$$

makes it clear that this factor will go to zero for two relevant limit cases (1) $\beta \to 1$ *and* $\gamma \to 0$, and (2) $\beta \to 0$ *and* $\gamma \to 1$.

A preliminary set of scale factors, which we will refer to as "scaling set 1," were defined as:

$$\begin{aligned} s_1 &= \sqrt{\alpha(1-\alpha)(1-\beta)} \\ s_2 &= \sqrt{\alpha\beta(1-\beta)} \\ s_3 &= \sqrt{\beta\gamma(\gamma-1)} \end{aligned} \tag{3.31}$$

Consider the results obtained with this set of scaling factors (Table 3.1) for each of the cases listed above where two edge intersect nodes approach coarse element nodes. These results are simply observations about the nature of $\mathbf{K}^{e,11}$ once scaled.

This set of scaling factors eliminates all unbounded stiffness terms. These factors do not solve the condition of canceling the zeros of the denominators in all cases, but when they do not the limiting form of the numerator helped prevent singular stiffness terms. One third (highlighted in green) of the double limit cases listed above do not produce a rank deficient $\mathbf{K}^{e,11}$, and thus scaling set 1 is a strong candidate for scaling these cases. For all cases where

Table 3.1. Effects of scaling set 1, for "double limiting cases."

1. α=1/2

Limit case	$\beta \to 0,\ \gamma \to 0$	$\beta \to 0,\ \gamma \to 1$	$\beta \to 1,\ \gamma \to 0$	$\beta \to 1,\ \gamma \to 1$
subelement	tri-sliver	chip	quad-sliver	tri-sliver
Result	rank deficient	rank deficient	finite terms	rank deficient

2. β=1/2

Limit case	$\alpha \to 0,\ \gamma \to 0$	$\alpha \to 0,\ \gamma \to 1$	$\alpha \to 1,\ \gamma \to 0$	$\alpha \to 1,\ \gamma \to 1$
subelement	tri-sliver	quad-sliver	chip	tri-sliver
Result	rank deficient	finite terms	rank deficient	diagonal

3. γ=1/2

Limit case	$\alpha \to 0,\ \beta \to 0$	$\alpha \to 0,\ \beta \to 1$	$\alpha \to 1,\ \beta \to 0$	$\alpha \to 1,\ \beta \to 1$
subelement	tri-sliver	chip	quad-sliver	tri-sliver
Result	rank deficient	rank deficient	finite terms	rank deficient

Results Key for scaled $\mathbf{K}^{e,11}$:

infinite terms: as denoted some terms become unbounded

finite terms: all terms finite and nonzero

finite & 0 terms: all terms finite but scaling made some zero

diagonal: all terms finite and all off-diagonal terms zero

rank deficient: all terms finite but scaling made it obviously rank deficient (zero row and column)

the scaled stiffness terms were finite, the Eigenvalues of $\mathbf{K}^{e,11}$ were analytically determined to verify a lack of rank deficiency

While not all "rank deficient" cases have been checked, for those that have been checked, a rank deficient $\mathbf{K}^{e,11}$ has produced additional "rank deficiency" in $\mathbf{K}^{e}$; the normal rank deficiency of $\mathbf{K}^{e}$ of 1, increases to 2 or 3. The normal rank deficiency of the element matrix is a constant temperature mode (analogous to a rigid body mode in solid mechanics), which does not pose a problem given boundary conditions that yield a unique solution. The increase of the rank deficiency (or equivalently dimension of the null-space, analogous to a spurious hourglass mode in solid mechanics) is a potential problem. Obviously in the limit it is a problem, but for edge nodes that are close to the (coarse element) corner nodes the resulting ill-conditioning could be less than what the unbounded stiffness terms would produce. Further analysis would be needed to verify this assertion, and it is possible that scaling factors that do not increase the rank deficiency can be determined. We should also note that while we are evaluating limits to determine candidate scaling factors, a typical implementation would move the material interface to coincide with a node or edge when it is within some tolerance.

Let us now consider other sets of scaling factors and the results they yield for the double limit cases. These results will demonstrate that the scaling factors that yield finite stiffness terms for a given case are not unique and suggest that one could construct sets that are applicable to each case. We did not investigate it long enough to determine if a single set could be determined that would be effective for all cases or determine sets for all limit cases, as there are other issues that merit investigation before they could be successfully applied in higher dimension.

Scaling set 2 is obtained by noting that scaling set 1 lacks symmetry in the edge intersect coordinates. As such let's define the second set as follows:

$$\begin{aligned} s_1 &= \sqrt{\gamma\alpha\,(1-\alpha)} \\ s_2 &= \sqrt{\alpha\beta\,(1-\beta)} \\ s_3 &= \sqrt{\beta\gamma\,(\gamma-1)} \end{aligned} \tag{3.32}$$

Consider the results obtained with this set of scaling factors (Table 3.2) for each of the cases listed above where two edge intersect nodes approach coarse element nodes. These results, again, are simply observations about the nature of $\mathbf{K}^{e,11}$ once scaled.

This set of scaling factors eliminates all unbounded stiffness terms. One half (highlighted in green) of the double limit cases listed above do not produce a rank deficient $\mathbf{K}^{e,11}$, and thus scaling set 2 is a strong candidate for scaling these cases.

Scaling set 3 is obtained by replacing each of the interface edge coordinates with one minus the coordinates as follows:

$$s_1 = \sqrt{(1-\gamma)\,\alpha\,(1-\alpha)}$$

Table 3.2. Effects of scaling set 2, for "double limit cases."

1. α=1/2

Limit case	$\beta \to 0,\ \gamma \to 0$	$\beta \to 0,\ \gamma \to 1$	$\beta \to 1,\ \gamma \to 0$	$\beta \to 1,\ \gamma \to 1$
subelement	tri-sliver	chip	quad-sliver	tri-sliver
Result	rank deficient	rank deficient	finite terms	diagonal

2. β=1/2

Limit case	$\alpha \to 0,\ \gamma \to 0$	$\alpha \to 0,\ \gamma \to 1$	$\alpha \to 1,\ \gamma \to 0$	$\alpha \to 1,\ \gamma \to 1$
subelement	tri-sliver	quad-sliver	chip	tri-sliver
Result	rank deficient	finite terms	rank deficient	diagonal

3. γ=1/2

Limit case	$\alpha \to 0,\ \beta \to 0$	$\alpha \to 0,\ \beta \to 1$	$\alpha \to 1,\ \beta \to 0$	$\alpha \to 1,\ \beta \to 1$
subelement	tri-sliver	chip	quad-sliver	tri-sliver
Result	rank deficient	rank deficient	finite terms	diagonal

Results Key for scaled $\mathbf{K}^{e,11}$:

infinite terms: as denoted some terms become unbounded

finite terms: all terms finite and nonzero

finite & 0 terms: all terms finite but scaling made some zero

diagonal: all terms finite and all off-diagonal terms zero

rank deficient: all terms finite but scaling made it obviously rank deficient (zero row and column)

Table 3.3. Effects of scaling set 3, for "double limit cases."

1. α=1/2

Limit case	$\beta \to 0,\ \gamma \to 0$	$\beta \to 0,\ \gamma \to 1$	$\beta \to 1,\ \gamma \to 0$	$\beta \to 1,\ \gamma \to 1$
subelement	tri-sliver	chip	quad-sliver	tri-sliver
Result	diagonal	rank deficient	infinite terms	rank deficient

2. β=1/2

Limit case	$\alpha \to 0,\ \gamma \to 0$	$\alpha \to 0,\ \gamma \to 1$	$\alpha \to 1,\ \gamma \to 0$	$\alpha \to 1,\ \gamma \to 1$
subelement	tri-sliver	quad-sliver	chip	tri-sliver
Result	diagonal	infinite terms	rank deficient	rank deficient

3. γ=1/2

Limit case	$\alpha \to 0,\ \beta \to 0$	$\alpha \to 0,\ \beta \to 1$	$\alpha \to 1,\ \beta \to 0$	$\alpha \to 1,\ \beta \to 1$
subelement	tri-sliver	chip	quad-sliver	tri-sliver
Result	diagonal	rank deficient	infinite terms	rank deficient

Results Key for scaled $\mathbf{K}^{e,11}$:

infinite terms: as denoted some terms become unbounded

finite terms: all terms finite and nonzero

finite & 0 terms: all terms finite but scaling made some zero

diagonal: all terms finite and all off-diagonal terms zero

rank deficient: all terms finite but scaling made it obviously rank deficient (zero row and column)

$$s_2 = \sqrt{(1-\alpha)\,\beta\,(1-\beta)} \qquad (3.33)$$
$$s_3 = \sqrt{(1-\beta)\,\gamma\,(\gamma-1)}$$

Consider the results obtained with this set of scaling factors (Table 3.3) for each of the cases listed above where two edge intersect nodes approach coarse element nodes.

This set of scaling factors does not eliminate all unbounded stiffness terms, but it does provide candidate scaling factors for the remaining sliver cases.

The only cases we do not have candidate scaling sets for are those with chip sub-elements. Based upon the form of the denominator factors, several scaling sets were examined, but the best result obtained so far is "rank deficient." Since determining the scale factors is not a purely analytical process, let's postpone further examination of those cases. As previously noted, there is a hurdle to using these scaling factors that must also be resolved, specifically adjacent elements containing a common material interface will have different scaling factors for the common edge node.

A possible next step could have been to examine a simple problem where the material

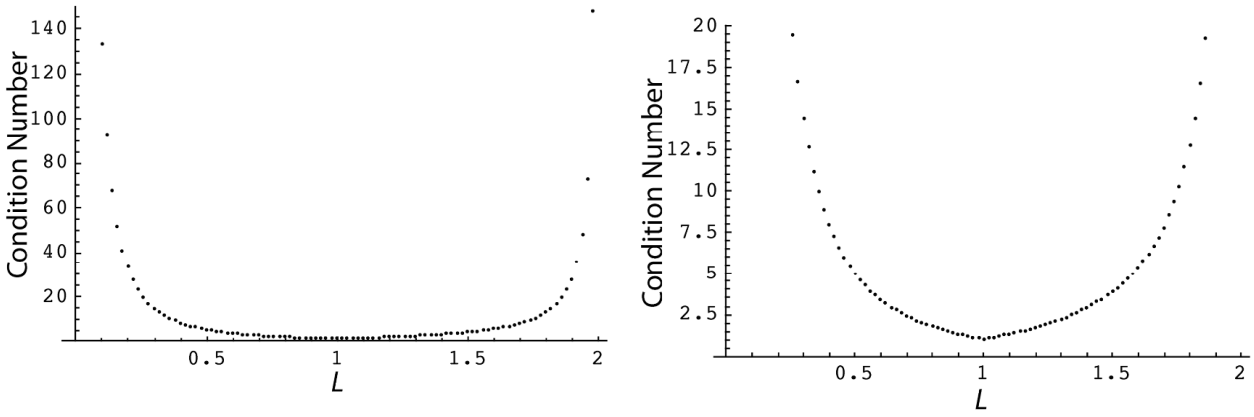


Figure 3.6. Condition number for stiffness of an isosceles triangle element (with unit equal sides) versus the third side length (L) – at two scales.

interface cut across a single hierarchical element. However, instead we decided to "back-up" and consider the scaling for a single standard finite element. It allows us to examine this issue of how element scaling factors might be used to determine a diagonal preconditioner for a global system, in the context of a simpler element formulation. Also for this problem, we will see for the geometry examined, an element scaling can be determined that yields a perfect condition number for the element.

3.4 Optimal Scaling Factors for a Single Standard Finite Element

The initial intent of this investigation was to determine if this provided an alternate path to the scale factors for hierarchical element formulations. Examining a single standard element requires us to address two questions: (1) can one determine scaling for a single finite element that gives a condition number of 1, and (2) can one use the result from a single element to determine a global scaling for a mesh of elements? This later question is nontrivial for multi-dimensional problems and is shared by approaches that adopt hierarchical elements. For now let's consider the simpler question, and analyze a single element that in the limits approaches a sliver.

Initially we examine the condition number of a single element without scaling, which motivates the need for the scaling. The condition number is simply the ratio of 3^{rd} and 2^{nd} Eigenvalues of the element stiffness (3.18) which can be obtained analytically. As a model problem we consider isosceles triangles, with equal sides of unit length, and a third side of varying length L. Both limit cases, $L \rightarrow 0$ and $L \rightarrow 2$, produce sliver like elements. Figure 3.6 depicts the variation of the condition number as a function of L.

A condition number of 1 is given by an equilateral triangle, and for a right isosceles triangle (corresponding to the master element) $L = \sqrt{2} \cong 1.41$ the condition number is 3. The condition number becomes unbounded in both limit cases, illustrating the potential problem with the arbitrary sliver elements that can result from slicing an existing element along a material interface.

Now we consider a diagonal scaling of the element matrix to determine the optimal scaling of this form.

$$\mathbf{K}^{es} = \mathbf{S}\mathbf{K}^{e}\mathbf{S} \tag{3.34}$$

where $\mathbf{S}$=$diag(S_1,S_2,S_3)$. Omitting the details for brevity, one can analytically solve for the condition number for this case. Since one Eigenvalue is zero, the remaining two nonzero Eigenvalues are obtained by solving the quadratic equation, and thus the form of the condition number (C) is given by

$$C\left(K^{es}\right) = \frac{b + \sqrt{d}}{b - \sqrt{d}}$$

where

$$b = \kappa \left\{ S_1^2 \left[\left(x_{21} - x_{31}\right)^2 + \left(y_{21} - y_{31}\right)^2 \right] + S_2^2 \left[x_{31}^2 + y_{31}^2 \right] + S_3^2 \left[x_{21}^2 + y_{21}^2 \right] \right\}$$

$$d = \kappa^2 \left[b^2 - 4(x_{31}y_{21} - x_{21}y_{31})^2 \left(S_1^2 S_2^2 + S_1^2 S_3^2 + S_2^2 S_3^2 \right) \right]$$

Note that if we can determine scaling factors such that d=0, the condition number will be one. If we assume the scaling factors are real (or purely imaginary, but not complex), then $\mathbf{K}^{es}$ is a real symmetric matrix and thus has real Eigenvalues. This requires d to be a non-negative real number. For this case we can seek to minimize the condition number by minimizing d. Since the scale factors only occur as squares, we will seek to minimize d with respect to $s \equiv \{S_1^2, S_2^2, S_3^2\}$. Stationary points occur where

$$\nabla_s d = 0$$

Of the three equations, two are linearly independent. Solving them for S_2^2 and S_3^2 in terms of S_1^2 gives

$$S_2^2 = S_1^2 \frac{x_{21}\left(x_{21} - x_{31}\right) + y_{21}\left(y_{21} - y_{31}\right)}{x_{21}x_{31} + y_{21}y_{31}}$$

$$S_3^2 = S_1^2 \frac{x_{31}\left(x_{31} - x_{21}\right) + y_{31}\left(y_{31} - y_{21}\right)}{x_{21}x_{31} + y_{21}y_{31}} x$$

thus giving scale factors of

$$S_2 = \pm S_1 \sqrt{\frac{x_{21}\left(x_{21} - x_{31}\right) + y_{21}\left(y_{21} - y_{31}\right)}{x_{21}x_{31} + y_{21}y_{31}}} \tag{3.35}$$

$$S_3 = \pm S_1 \sqrt{\frac{x_{31}\left(x_{31} - x_{21}\right) + y_{31}\left(y_{31} - y_{21}\right)}{x_{21}x_{31} + y_{21}y_{31}}} \tag{3.36}$$

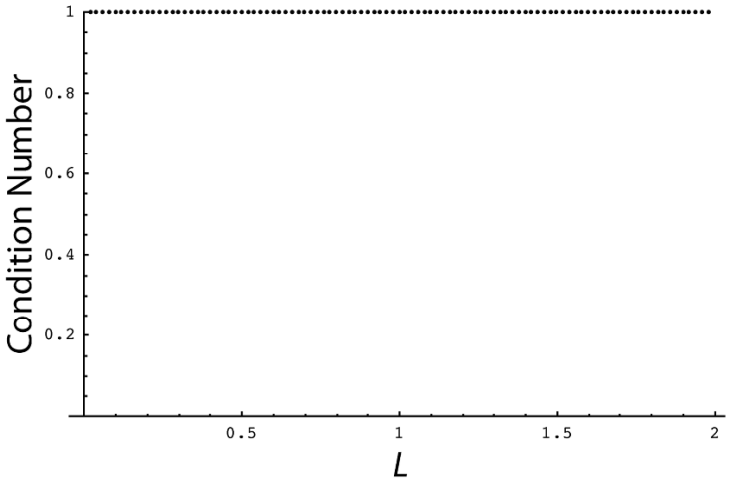


Figure 3.7. Condition number for scaled stiffness of an isosceles triangle element (with unit equal sides) versus the third side length.

Without loss of generality let's examine the scaling when we adopt the positive signs and take S_1=1. Figure 3.7 depicts the analytically derived results for our simple model problem. While a boring plot, it shows that the scaling not only minimized d but gave d=0, yielding perfect conditioning.

Figure 3.8 shows that the Eigenvalues are positive for acute isosceles triangle elements and negative for obtuse isosceles triangle elements. The corresponding stiffness for an acute isosceles triangle is non-negative definite and for an obtuse isosceles triangle non-positive definite. Ironically the Eigenvalues become unbounded as the right isosceles triangle case is approached and approach zero for both sliver element cases.

Now consider how the dependent scaling factors change with L. Figure 3.9 shows the real and imaginary parts of the scaling factors for the model problem. The nature of the scaling factors changes going from acute to obtuse isosceles triangle elements – real for the former and imaginary for the later. For the case of isosceles triangles, because these scaling factor forms the scaled stiffness is always real valued.

The single element model problem was selected because of its simplicity in generating the sequence of geometries ranging from the *vertical sliver* ($L\rightarrow$ 0) to the *horizontal sliver* ($L\rightarrow$ 2), however it is also a special case, because of the element symmetry. In general the scaling factors do not necessarily produce a scaled element stiffness matrix that is real valued as obtained for the model problem. Another example problem that was examined had two sides of fixed lengths (l_1=2/3 and l_2=4/3). As the angle between these legs changed, S_2 and S_3 changed from real to imaginary at different angles, and thus $\mathbf{K}^{es}$ had real and imaginary elements. As such $\mathbf{K}^{es}$ was complex symmetric ($\mathbf{K}^{es}$=$\mathbf{K}^{esT}$) but not Hermitian ($\mathbf{K}^{es}\neq\mathbf{K}^{esH}$). The nonzero Eigenvalues were real and equal thus giving a condition number of one. Unfortunately, the scaling while optimal in one respect, would require the global

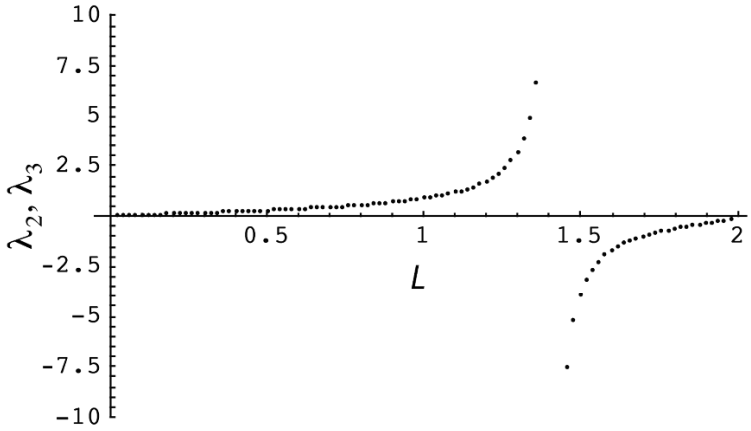


Figure 3.8. Highest two, equal Eigenvalues for the scaled element stiffness.

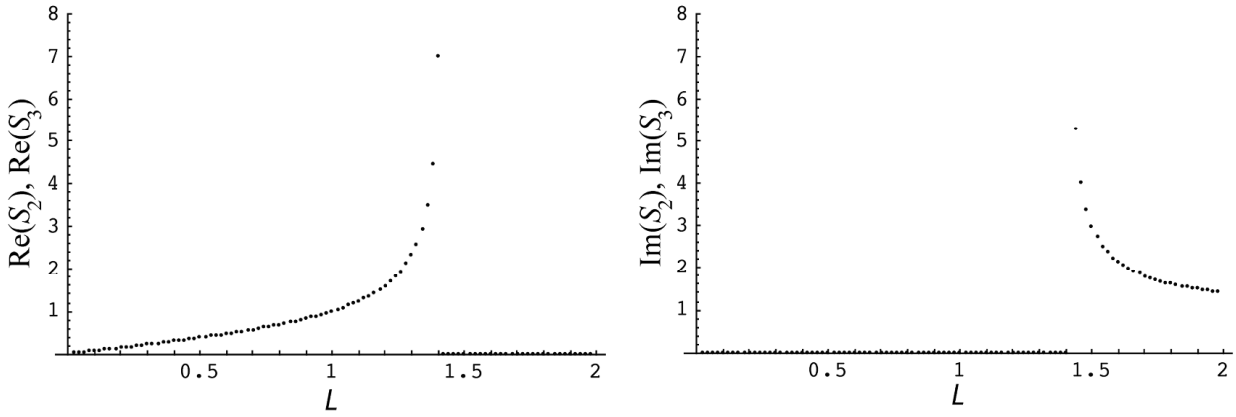


Figure 3.9. Real and imaginary parts of the scaling factors S_2 and S_3.

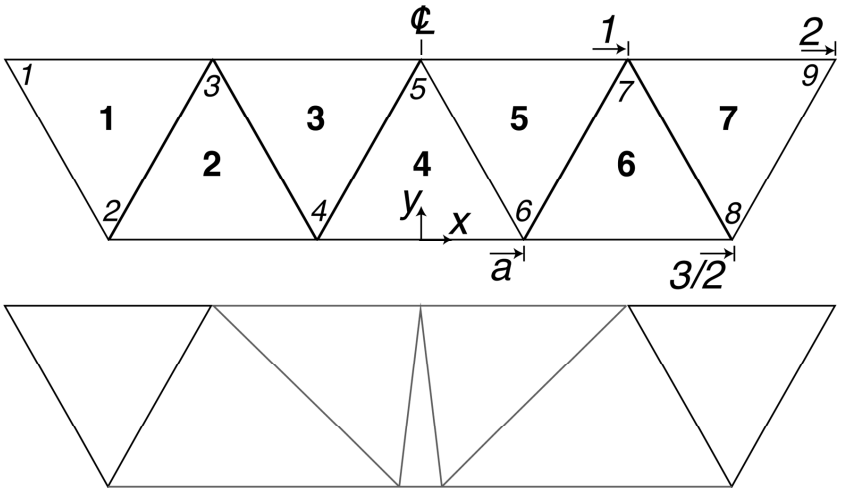


Figure 3.10. Schematic of triangle element, bar model with the center element degenerating into a sliver.

stiffness matrix to be represented by complex numbers.

The above results make it unclear if the optimal, single element, scaling developed above can be used in a general scaling approach, however, maybe (in the context of hierarchical approach) a different sub-element partitioning could be adopted to avoid the complex scaling parameters, or the optimal real or imaginary values (not investigated) would be sufficient.

3.4.1 Combining Element Scale Factors

For now let's just consider the case of element geometries that yield real scaling factors, and seek approaches of using the element scaling to determine a scaling for the global matrix. Unfortunately we lack a theoretical basis for determining how to combine the element scaling into a global scaling vector, so we resort to a numerical test using a simple bar model having one sliver element.

The test examines a bar initially composed of equilateral triangles, for reference. Then it modifies the geometry of the center element toward a vertical sliver. The original geometry is shown below, Figure 3.10, followed by a modified geometry. The modified geometry maintains an isosceles triangle (equal lengths of edges $\bar{45}$ and $\bar{56}$) for the middle element, thus shortening edge $\bar{46}$ in a symmetric manner. Note that for this short bar, all of the elements except the end two are changed by the center element modification, but only the center element approaches a "pathological shape."

As already noted, the resulting issue is that the sliver element can produce a very ill-conditioned system of equations, and the lingering question is "can "perfect scaling" for each element (as obtained in the previous section) be combined to form a global diagonal preconditioner that effectively eliminates the ill-conditioning problem?"

There are two aspects to the combination of element scaling factors: (1) how does one

Table 3.4. Element scaling and element scaling assembly approaches.

Scheme	**Element scaling approach**	**Element assembly approach**
1	maximum S_i=1	multiplicative
2	minimum S_i=1	multiplicative
3	maximum S_i=1	additive
4	minimum S_i=1	additive
5	maximum S_i=1	minimum element term only
6	minimum S_i=1	maximum element term only
7	maximum S_i=1	multiplicative, critical element only

scale the factors for each element (recall only two of the three factors were linearly independent) and (2) how does one "assemble" a global diagonal scaling matrix from the element contributions?

The following schemes (Table 3.4) were considered for forming a global diagonal scaling matrix (preconditioner).

The element scaling approaches define the linearly independent scaling factor by requiring that either the minimum or maximum scaling factor equals one. The element assembly approaches vary significantly. The "multiplicative" approach is motivated by the element by element work of Hughes (see, *e.g.*s, where the initial global scaling matrix is the identify matrix, and each element contribution is multiplied by the corresponding global scaling matrix term). The "additive" approach follows the standard assembly approach for element stiffness terms. The "minimum" and "maximum" element term approaches (strategies 5 and 5) only assemble the minimum and maximum element scaling terms, respectively. Lastly, strategy 7 adopts an identity scaling except for the nodes attached to the critical sliver element, for which the sliver element's optimal scaling values are adopted.

None of these schemes to construct a global diagonal scaling matrix from the element diagonal scaling matrices yielded much improvement in the conditioning of the system. For equilateral triangle elements the condition number was 12.7. For a mesh modified to include a sliver element of 1%[1] the condition number for the un-scaled system was about 426. Scheme 5 gave the best result this case, yet only improved the condition number to about 144.

With such a modest improvement, one obvious question is "how much can a diagonal preconditioning approach improve the conditioning of our system (*i.e.*, does the approach of using the element scaling to determine a diagonal system scaling even have the potential to significantly improve the condition of our system)?"

To examine this last question we took the sensitivity (*i.e.*, tangent stiffness) matrix for the above problem, scaled it by a diagonal preconditioner (symbolically), and then solved the optimization problem of minimizing the condition number as a function of the scaling

[1] Denoting that the length of the horizontal leg was 1% the height of the triangle.

Table 3.5. Condition numbers for unscaled and optimum-scaled cases.

Element 4 Geometry	K-unscaled Condition Number	K-opt-scaled Condition Number	K-$\mathbf{S}_d$-scaled Condition Number
Equilateral	12.7	7.71	7.91
1% Isosceles	426	10.1	11.8
0.1% Isosceles	4219	10.5	13.1
0.01% Isosceles	42152	10.5	13.2

Table 3.6. Scaling factors obtained from optimization.

Element 4 Geometry	S[1]	S[2]	S[3]	S[4]	S[5]	S[6]	S[7]	S[8]	S[9]
Equilateral	1.73	1.25	1.00	1.00	0.80	0.90	1.00	1.24	1.80
1% iso	11.9	6.8	6.6	1.3	5.4	1.0	6.4	7.5	13.3
0.1% iso.	36.4	21.0	18.8	1.16	14.6	0.97	18.9	21.0	38.1
0.01% iso.	119	71	62	1.2	50	1.0	61	71	126

parameters. As another measure for the effectiveness of preconditioner schemes that have their roots in the analytical solution for a single element, we also compared our remaining results to a classical preconditioner based upon the diagonal of the sensitivity matrix, that is

$$\mathbf{S}_d = \mathbf{d}^{-1/2} \tag{3.37}$$

where $\mathbf{d}$ is simply the diagonal partition of the sensitivity matrix, in the form of a diagonal matrix.

Table 3.5 presents the condition numbers for several element geometry cases for the solver cases of (a) no preconditioning (original system), (b) optimum diagonal preconditioner, and (c) the classical diagonal preconditioner above ($\mathbf{S}_d$).

The optimum scale values are presented in Table 3.6; coloring is used to denote scaling associated with symmetric nodes, which should have similar scaling values. Note that the optimum scalings scale up all the nodes except the two nodes that are approaching one another, as these are the nodes which contribute to the large shape function gradient terms and corresponding large stiffness terms. As such, the neighbors are scaled to more closely match the gradient terms, yielding a smaller condition number. In fact the condition number is slightly smaller than that obtained from an unscaled stiffness with ideal (equilateral) elements. This feature was expected, but we did not necessarily expect the growth that occurred in scaling terms at the edges of the domain is approached. We expect that these scaling terms grow because fewer elements are connected to the nodes, thus yielding smaller

Table 3.7. Condition numbers for trial nodal scale factors for element geometry 0.01% Isosceles case.

K-opt-scaled Condition Number	S[1]	S[2]	S[3]	S[4]	S[5]	S[6]	S[7]	S[8]	S[9]
20.56	50	50	50	1	50	1	50	50	50
15.56	150	75	50	1	50	1	50	75	150

assembled (*i.e.*, global) stiffnesses.

Of course we would prefer a theoretical basis for determining how to use the element scaling to obtain the global system scaling; in lieu of that, let's take an empirical approach and determine if a variation of one the strategies that we had previously examined (without knowing the nature of the optimum scaling) might potentially be modified to yield similar improvements. It is beyond the scope of this study to actually pursue strategies further, but we can still examine which features of the optimum solution are most important to guide future work.

Let's first examine what the effect on the condition number is if all the nodes except those associated with the high gradient terms have the same magnitude – in particular let's assume they match the scaling of node 5.

The first set of scaling values shows that we get most the "return" on our preconditioner just by scaling the "problem element." The second set of scaling parameters are modified to account for the connectivity, where the stiffness scaling is increased linearly to account for less elements connecting to a node. (Maybe the factor should be $\sqrt{1.5}$ instead of 1.5 for nodes 2 and 8).

The above results suggest that rather than trying to assemble the optimal analytical scaling for each element, it might be more effective to only include the analytical contributions from the sliver elements, or possibly combine them with the scaling of the classical diagonal preconditioner (3.37) for the remainder of the system. Strategies 5–7 (Table 3.4) were defined to more directly emphasize the effects of the sliver element upon the conditioning, but clearly the effects of adjacent elements must be accounted for too, even if to only determine the scaling for the three nodes connected to the sliver element.

One approach that came to mind in the "12^{th} hour," and thus was not examined, was a weighting of element scaling values to determine the nodal scaling such that it would give the same global, diagonal scaled stiffness term as assembling all of the scaled diagonal element stiffness terms. That is, we solve the following equation for S, a global scale factor for a particular node, where all the stiffness terms correspond to diagonal stiffness terms of the elements and global system

$$S^{e_1} k^{e_1} S^{e_1} + S^{e_2} k^{e_2} S^{e_2} + \ldots + S^{e_n} k^{e_n} S^{e_n} = SkS \tag{3.38}$$

where there are n elements connected to the node. This assembly strategy has the property that it weights the element scaling factor with the largest stiffness contribution in forming

the global scaling value. While that property would seem to be consistent with the global optimization results examined above, there is no theoretical basis for asserting that it is the best weighting. Note this approach, as suggested above, could be applied to the whole system or limited to the nodes associated with the sliver elements, that are the cause of the ill-conditioning.

3.5 Closing Discussion

The hierarchical basis function approach for modeling material discontinuities can produce a subelement which has a small sliver (or chip) shape and that leads to ill-conditioning. To address this issue we examined an algebraic scaling approach for the one-dimensional static conduction problem, which eliminated the singularity of the stiffness term that occurs in the "hierarchical-hierarchical partition" of the element stiffness matrix. The approach eliminated the ill-conditioning problem in the one-dimensional problem, but the effectiveness of the approach in one-dimension was tied to the fact that the material interface was completely contained in one hierarchical element. We extended the element scaling approach to the two-dimensional static conduction problem and obtained scale factors for many cases that would eliminate the potentially unbounded stiffness terms while not reducing the rank of the element stiffness. These scaling values have the potential to to improve the conditioning of the global system. Unfortunately for two dimensions, the material interface will typically cross many elements, so how to determine the global preconditioner (diagonal scaling) is an unknown. That is, while the scaling may very well eliminate the ill-conditioning in an individual hierarchical element, other elements would have scaling factor contributions to the same nodes, and we have no theoretical basis for their combination or assembly.

We "backed up" to examine the scaling for a single standard finite element for a few reasons: (1) to see if it provided insight to the form of the scaling for the hierarchical elements, and (2) to examine the issue of how to obtain a preconditioner for the global system based upon element scaling results, in the context of a simpler element formulation. Note that the second issue would also have to be considered to address the first issue, since the hierarchical element is composed of several subelements. For the geometric case of isosceles triangles, we were able to analytically solve for the perfect element diagonal scaling, *i.e.*, the scaling that would always produce an element condition number of 1. As such, we had the best possible case of examining approaches to combining element scaling contributions to obtain a global preconditioner. We sampled other element geometries too, and noted that the ideal scaling factors become complex for the general case. Examining these more general cases in detail was outside the scope of this initial study, because if the issue of combining element contributions is not resolved even perfect element scaling has no value, and thus merited some investigation. We examined several strategies for combining element scaling results using a simple bar problem. None of the strategies had a theoretical basis, and even the best one only gave moderate improvements in the global conditioning.

In the last section, we sought to answer the question (for the model problem) of how well

a diagonal preconditioner might affect the conditioning when a very small sliver element exists. The optimal diagonal scaling for the global system was determined for several sliver cases by solving the optimization problem for the best scale factors. The optimal results were considerably better than those obtained from the schemes that combined optimal element scaling results, and the classical diagonal preconditioner gave results relatively close to the global optimum. Variations of the optimal global results showed that most of the improvement in conditioning came from the global scaling of the nodes associated with the sliver element. This suggests that it might be sufficient to only address the analytical results for the pathological elements (slivers or chips), and for example simply use a method like the classical diagonal preconditioner for the remainder of the mesh. Another simple idea for determining the global scaling values (that emphasizes the importance of the diagonal stiffness terms) was proposed; this could be applied to the nodes of pathological elements alone or the whole mesh. More complicated test problems are needed to investigate approaches, as this very simple one does not motivate a more complicated approach. The issue of how to best utilize these novel analytical results for the hierarchical and standard finite elements to obtain a global stiffness preconditioner remains an open question. Numerical experimentation, for example, testing the untested scheme proposed above, could yield an *adhoc* but effective approach, but a theoretical basis is certainly preferable.

Chapter 4

Physics-Compatible Remap of Node and Edge Fields

4.1 Theoretical Foundation

Presented here is the progress made on the problem of remapping fields described by enriched compatible discretizations. Only the CDFEM enrichment is considered at this time, as it is conceptually somewhat easier than XFEM with respect to remap, and there is no reason to expect that the results here will not be extensible to the XFEM-AC discretization with the appropriate accommodations. The foundation of the present approach is the Constrained Interpolation (CI) remap work of Bochev and Shashkov [2]. Their work developed the CI remap method for a specialized version of the transverse magnetic reduction of the eddy current equations. This is different from the equations considered here, but they demonstrated that CI remap is a viable alternative to the constrained transport approach to remap (e.g., for these applications, [14]). Our interest in this approach is that it considers remap not as an advection problem, but as a method by which a discretized field is transferred from one mesh to another, in such a way as to preserve important properties of the field. CI remap is therefore well suited to our arbitrarily refined enriched meshes, and has certain similarities to mesh transfer methods, such as [5].

4.2 The Algorithm

Fundamentally, the constrained remap algorithm as developed here has three steps: from a given field $\mathbf{A}^o$ defined on the source mesh,

1. Construct 'low'- and 'high'-order interpolations of the source field on the destination mesh, $\mathbf{A}_l^n$ and $\mathbf{A}_h^n$,

2. Form the desired norm from the source field on the destination mesh, $\|\mathbf{B}^o\|^2$, (where $\mathbf{B}$ is related to $\mathbf{A}$ in some way),

3. (Locally) optimize the combination of low- and high-order interpolations such that this

norm is preserved, to obtain the remapped field:

$$\mathbf{A}^n = \lambda_{\text{opt}} \mathbf{A}_l^n + (1 - \lambda_{\text{opt}}) \mathbf{A}_h^n, \tag{4.1}$$

for λ_{opt} that minimizes the difference between the norm of the source field and the norm of the remapped field.

The following sections provide some detail regarding the methods used for each step. Note that at this level, the algorithm is agnostic to the field discretization, so could be applied to scalar or vector fields discretized by any compatible basis, whether nodal, edge or face based. As shown in Section 4.4, enough progress was made to demonstrate CI remap in 2-D for nodal fields based on both field and gradient (H^1) norms, and for edge fields using the $H(curl)$ norm.

4.2.1 Interpolation and Patch Recovery

Nodal Fields

The first step of the CI remap algorithm requires evaluation of different approximations of the source field on the destination mesh. Here we consider three orders of approximation: zeroth, first and second. First-order interpolation of the source field onto the destination mesh is almost trivial, requiring for each node only the location of the destination node and the element on the source mesh in which it falls. Interpolation then follows simply by evaluating the basis function at that location. The computational expense of this step is in the routine that finds the destination node on the source mesh, but even this can be minimized by exploiting a CFL-type constraint that ensures a destination node will lie inside the patch of elements it is connected to on the source mesh (we do not consider mesh motion that could alter the mesh connectivity).

Zeroth-order interpolation begins with the calculation of element averages of the source field on the source mesh. To compute the nodal field on the destination mesh, we again exploit the CFL-bound element patch connectivity, and compute the nodal value as the area-weighted patch-average of the element values from all source elements connected the the destination node. Formally,

$$\mathbf{A}_{0,i}^n = \frac{1}{\sum_{e \ni i} a_e} \sum_{e \ni i} a_e \mathbf{A}_{0,e}^o, \tag{4.2}$$

where a_e is the area of element e. Note that the zeroth-order approximation was used only for CI remap with a field L^2 norm, because the gradient of the zeroth-order field was too inaccurate to be useful, and dissipativity could be achieved without it.

Second-order interpolation requires first a patch recovery step on the source mesh. This constructs a local second-order accurate (quadratic) polynomial approximation of the field

in each element of the form

$$A(x, y) \approx c_1 + c_2 x + c_3 y + c_4 xy + c_5 x^2 + c_6 y^2, \tag{4.3}$$

using a largely standard procedure. For each element, a 6×6 system is constructed from the contributions of all nodes of the elements connected to it, which on the structured meshes tested could be shown to always exceed 6 to ensure an nonsingular system and a valid least-squares solution. The innovation here was to make the patch recovery aware of the interface in the CDFEM formulation. Each subelement of the CDFEM mesh has its own set of coefficients, and contributions are taken only from nodes on the same side of the interface as the (sub-)element of interest. It should be obvious that allowing patches to form a quadratic approximation across an interface where the field is known to be only C^0 continuous is a bad idea. A consequence of this decision, however, is that elements could now have fewer than 6 nodes in their interface-aware patches (or simply be poorly conditioned), in which case the patches are expanded to include the next layer of element neighbors, or in the worst case, are simply reduced to linear interpolation only.

This approach was verified to reproduce exactly not only linear fields, but also quadratic fields that were approximated by linear basis functions on the original mesh. Finally, the reconstructed field on the destination mesh is calculated at each node by identifying the source element in which it falls, using the same algorithm described for the linear interpolation. From the source element, the patch coefficients are evaluated at the destination node location to compute the new field value.

Edge Fields

For edge fields, the interpolation and patch recovery processes look broadly similar to that for nodal fields. However, we did not consider zeroth-order interpolation for the low-order field approximation, since the $H(curl)$ norm will be very poorly approximated in this case. Most significantly, and critical to preservation of convergence in the edge discretization, was the addition of a step to intersect the edges of the destination mesh with the source mesh. Note that this is only a one-way intersection, so is significantly less computationally expensive than a full mesh intersection. This is necessary because edge elements require line, rather than only point, evaluation of the source field. Considerable information can be lost if evaluation of the source field along each destination edge is left simply to where Gauss points happen to fall on the source mesh. Also, note that once the edge segments have been defined by the mesh intersection, only one or two Gauss points are required to exactly integrate the source field on the new edges, mollifying the computational expense compared to the many point-in-element searches necessary with multiple Gauss points.

Linear interpolation on each destination edge follows the nodal practice of evaluating the basis functions from the source elements, now at the Gauss points used to integrate along each edge segment. Patch recovery for edges is extended to represent a vector field, in the form

$$\mathbf{A}(x, y) \approx \begin{bmatrix} c_1 + c_2 x + c_3 y + c_4 xy + c_5 y^2 \\ c_6 + c_7 x + c_8 y + c_9 xy + c_{10} x^2 \end{bmatrix}. \tag{4.4}$$

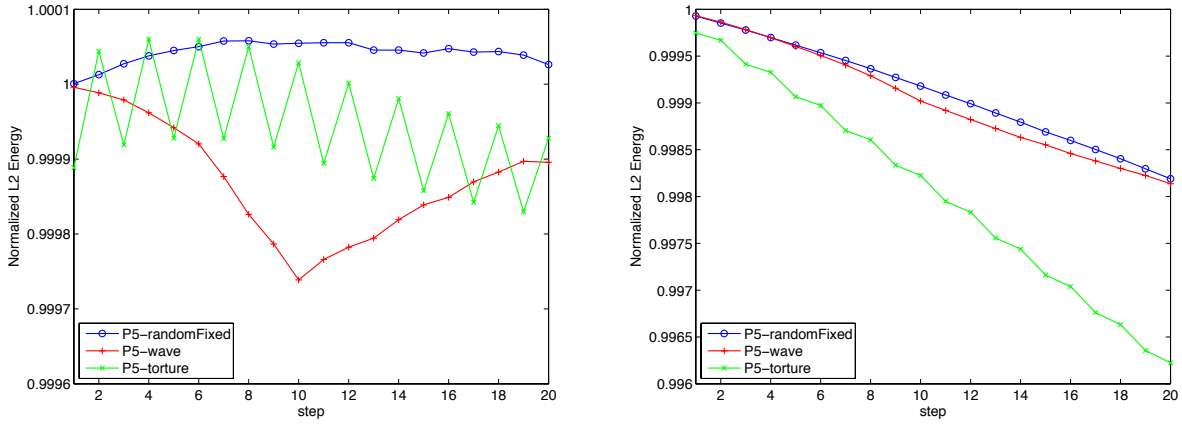
Note that this form excludes the x^2 term from the x component, and y^2 from the y component. This was motivated by the basis functions of second-order Nédélec edge elements, which do not contain these components. Testing with complete quadratic polynomials indicated that results were (at best) no better than with the reduced form. Also, because the basis of the first-order Nédélec edge elements is incomplete, exact recovery of quadratic fields representable with (4.4) is not possible. Calculation of the patch coefficients for a given element forms the 10×10 linear system by evaluating the vector field at three Gauss points in each element neighboring the element of interest, three being chosen to correspond to the number of degrees of freedom represented in each triangle. Being internal to each element, the Gauss points also avoid issues of partial continuity across edges. As with the nodal patch recovery, patch coefficients respect the interface and will use a larger patch if the immediate element neighbors do not yield a solvable system.

4.2.2 Evaluation of the Reference Norm

The second step of the algorithm constructs the element-wise target norms. This requires an estimate of the norm of the source field in each element of the destination mesh, $\|\mathbf{B}^o\|^2$. For the demonstration problems in Section 4.4, this step was idealized by integrating the analytic field directly, which is equivalent to a perfect remap or transferal of the source energy to the destination mesh. In practice, when an analytic solution is not available, some approximation of this quantity will need to be made.

A simple approximation that we investigated was to use quadrature to integrate the source field on the destination mesh. This approach relies simply on Gauss points sampling adequately from the source elements overlapped by the destination element being integrated. Clearly, this approach will introduce greater error for rapidly changing fields and small mesh deformations where source and destination elements differ only slightly. Accuracy is also arbitrarily dependent on the number of Gauss points and where they happen to fall on the source mesh. In general, though, this approach was found to be an adequate simplification and slightly more dissipative than the exact integration. As shown in Figure 4.1, for test problem (4.21) from Section 4.4, this is not necessarily a bad thing.

An alternative approach is to remap the reference energy $\|\mathbf{B}^o\|^2$ from the source mesh to the destination mesh as a conserved element quantity. An obvious and accurate method for doing this is by intersection remap, which is conservative and essentially geometric in formulation, though can be computationally expensive. Note that the nodal (or edge, etc.) fields are still remapped using the CI approach, and it is only this reference energy that would be remapped by mesh intersection. We note that [2] uses a sign-preserving conservative interpolation method from [9] for their demonstrations, and we have not investigated this particular approach in any detail.



(a) Analytic $\|\mathbf{B}^o\|^2$ (b) Quadrature $\|\mathbf{B}^o\|^2$

Figure 4.1. Normalized L^2 norm (solution energy) over time for CI remap using different methods for calculating the target norm.

4.2.3 Optimization

From the previous two steps, all the necessary components for remap have been computed. In this final step, a simple optimization problem is solved to obtain the new representation of the field on the destination mesh. The problem statement is to find from the source field $\mathbf{A}^o$ an approximation $\mathbf{A}^n$ on the destination mesh such that

$$\int_\Omega |\mathbf{B}^n|^2 d\Omega = \int_\Omega |\mathbf{B}^o|^2 d\Omega, \tag{4.5}$$

where $\mathbf{B} = \mathbf{DA}$, to allow for the norm to be taken in a different space from the field depending on the choice of operator $\mathbf{D}$. For the present work, we consider L^2, H^1 and $H(curl)$ (semi-)norms, i.e., $B = A$, $\mathbf{B} = \nabla A$, and $\mathbf{B} = \nabla \times \mathbf{A}$, respectively. Equation (4.5) is very expensive to enforce exactly, so we weaken it to instead preserve the sum over all elements, i.e.,

$$\sum_{e \in T^n} |\mathbf{B}^n|_e^2 = \sum_{e \in T^o} |\mathbf{B}^o|_e^2. \tag{4.6}$$

Based on the interpolated field approximations $\mathbf{A}_l^n$ and $\mathbf{A}_h^n$, we define the candidate remapped field as the convex combination

$$\mathbf{A}^n(\lambda) = \lambda \mathbf{A}_l^n + (1 - \lambda)\mathbf{A}_h^n. \tag{4.7}$$

This allows the energy mismatch functional to be defined:

$$\mathcal{E}(\lambda) = \sum_{e \in T^n} \left| \|\mathbf{B}^o\|_e^2 - \|\mathbf{B}^n(\lambda)\|_e^2 \right|^2. \tag{4.8}$$

At this point, λ is a continuous function. In smooth regions of the source field, λ will tend to zero to pick the higher order component, and near discontinuities, λ will tend to one to ensure

conservation, acting somewhat as a limiter (though not in the strict sense of guaranteeing monotonicity).

In practice, λ must be discretized. Ideally, (4.8) would be optimized with the same discretization for λ as $\mathbf{A}$ (i.e., at nodes or edges, etc.), but this would result in a global optimization problem that could be very expensive to solve. Instead, we discretize λ as an element quantity to optimize (4.8) element-wise, with the element energy mismatch functional

$$\mathcal{E}_e(\lambda^e) = \left| \|\mathbf{B}^o\|_e^2 - \|\mathbf{B}^n(\lambda^e)\|_e^2 \right|^2. \tag{4.9}$$

This allows the optimization problem for λ^e to be defined element-wise:

$$\lambda^e_{\text{opt}} = \arg\min \left| \|\mathbf{B}^o\|_e^2 - \|\mathbf{B}^n(\lambda^e)\|_e^2 \right|^2 \quad \forall e \in T^n. \tag{4.10}$$

This optimization problem is in fact quite simple. Expanding (4.7), the norm of the remapped field can be written as

$$\|\mathbf{B}^n(\lambda^e)\|_e^2 = (\lambda^e)^2 \int_e (\mathbf{B}_l^n)^2 d\Omega_e + \lambda^e(1-\lambda^e) \int_e \mathbf{B}_l^n \mathbf{B}_h^n d\Omega_e + (1-\lambda^e)^2 \int_e (\mathbf{B}_l^n)^2 d\Omega_e, \tag{4.11}$$

to show that (4.9) is a simple squared quadratic function in λ^e. This can be almost trivially solved by testing the roots of the derivative of the equation, which are

$$\lambda_1^e = \frac{\mathcal{E}_{hh} - \mathcal{E}_{lh}}{\mathcal{E}_{hh} - 2\mathcal{E}_{lh} + \mathcal{E}_{ll}}, \tag{4.12}$$

$$\lambda_2^e = \frac{\mathcal{E}_{hh} - \mathcal{E}_{lh} - \sqrt{\mathcal{E}_{lh}^2 + \mathcal{E}_o(\mathcal{E}_{hh} - 2\mathcal{E}_{lh} + \mathcal{E}_{ll}) - \mathcal{E}_{hh}\mathcal{E}_{ll}}}{\mathcal{E}_{hh} - 2\mathcal{E}_{lh} + \mathcal{E}_{ll}}, \tag{4.13}$$

$$\lambda_3^e = \frac{\mathcal{E}_{hh} - \mathcal{E}_{lh} + \sqrt{\mathcal{E}_{lh}^2 + \mathcal{E}_o(\mathcal{E}_{hh} - 2\mathcal{E}_{lh} + \mathcal{E}_{ll}) - \mathcal{E}_{hh}\mathcal{E}_{ll}}}{\mathcal{E}_{hh} - 2\mathcal{E}_{lh} + \mathcal{E}_{ll}}, \tag{4.14}$$

where

$$\mathcal{E}_o = \int_e (\mathbf{B}^o)^2 d\Omega_e, \quad \mathcal{E}_{ll} = \int_e (\mathbf{B}_l^n)^2 d\Omega_e, \quad \mathcal{E}_{lh} = \int_e \mathbf{B}_l^n \mathbf{B}_h^n d\Omega_e, \quad \mathcal{E}_{hh} = \int_e (\mathbf{B}_h^n)^2 d\Omega_e.$$

The value for λ^e_{opt} is then simply that from the set $\{0, \lambda_1^e, \lambda_2^e, \lambda_3^e, 1\}$ which yields the minimum value for (4.9).

The final step is to define the compatible λ^i_{opt} from the elemental λ^e_{opt} to compute the final remapped field. Following [2], we simply patch-average to the node or edge:

$$\lambda^i_{\text{opt}} = \frac{1}{N_i} \sum_{e \ni i} \lambda^e_{\text{opt}}, \tag{4.15}$$

where N_i is the number of elements in the patch of the node/edge i (e.g., for edges in 2-D, $N_i = 2$). This is generally a good approximation, but some ambiguity exists in the CDFEM context for nodes and edges that lie on the interface. It is not clear that averaging from both sides of the interface onto these entities is the best approach for preserving discontinuities.

On the other hand, the evidence from the convergence studies suggests that, at least for the nodal problems, this strategy may be adequate. Further work is recommended in this regard.

A final comment should be made regarding the energy mismatch functionals. In general, these converged to near-perfect conservation quite uniformly with mesh refinement. However, numerous examples have been observed where the element-wise norms of the low- and high-order field approximations did not strictly bound the norm of the source field, and this was seen regardless of the method used to compute the target norm. Clearly, the principle behind the concept of CI remap is that

$$\|\mathbf{B}_l^n\|^2 \leq \|\mathbf{B}^o\|^2 \leq \|\mathbf{B}_h^n\|^2.$$

Most problematic is therefore the case when $\|\mathbf{B}_l^n\|^2 > \|\mathbf{B}^o\|^2$, as this implies that the remapped field will have greater energy than the source field. An explanation for why this occurs was never found. However, these errors tended to be small and canceled across the mesh, which meant that remap was in general slightly dissipative, as desired. However, some cycle-to-cycle anomalies where the total energy of the field increased can be attributed to this effect.

4.3 A Simple Demonstration

The following example demonstrates the concept of Constrained Interpolation remap. Consider the field

$$U(x, y) = \cos(\pi y)(\sinh(\pi x) - \coth(\pi)\cosh(\pi x)), \tag{4.16}$$

that is initially discretized on the nodes of the base (unperturbed) mesh shown in Figure 4.2(a). An interface is located on the line $x = 0.4$, and those elements cut by the interface are enriched using the CDFEM approach.

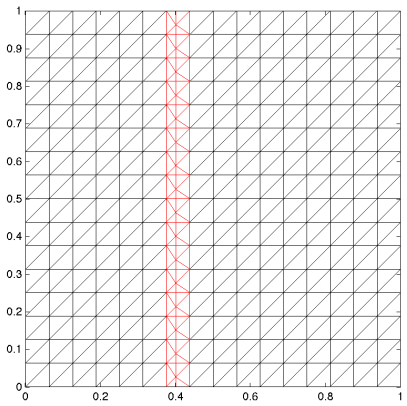
Three different mesh motions are demonstrated over $N = 20$ remap cycles, a snapshot of each being illustrated in Figure 4.2. Note that the base mesh is structured, so the nodal locations may be indexed by (i, j) pairs where $i \in [1, I]$ and $j \in [1, J]$. The `RandomFixed` motion perturbs all interior nodes of the mesh by a random increment each cycle, until the final cycle when the nodes are returned to their original locations. The motion is described by

$$\boldsymbol{x}_{i,j}^{n+1} = \boldsymbol{x}_{i,j}^n + r\Delta\left(\text{rand}() - \frac{1}{2}\right), \tag{4.17}$$

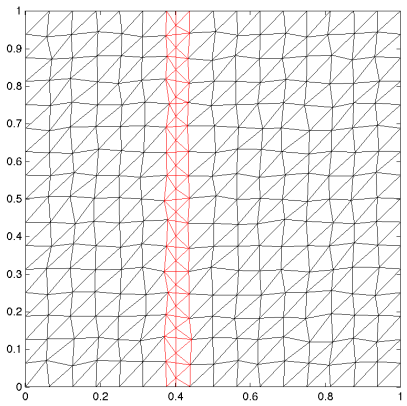
for $i \in [2, I-1]$ and $j \in [2, J-1]$ and $n < N-1$, where $\boldsymbol{x}_i^N = \boldsymbol{x}_i^0$. For the examples that follow, $r = 1/4$ and $\Delta = 1/\min(I-1, J-1)$, and rand() is a random number between 0 and 1.

The `Wave` motion is a linear in time sinusoidal spatial perturbation of the mesh, of the following form:

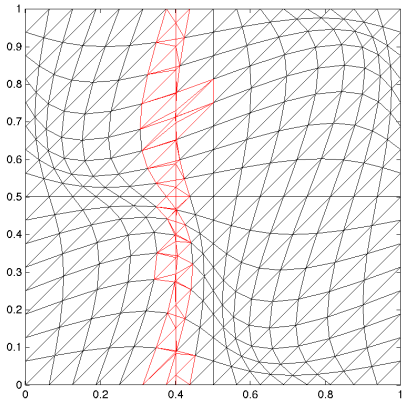
$$\boldsymbol{x}_{i,j}^{n+1} = \boldsymbol{x}_{i,j}^n + \alpha(t)\sin(2\pi x_{i,j}^n)\sin(2\pi y_{i,j}^n), \tag{4.18}$$



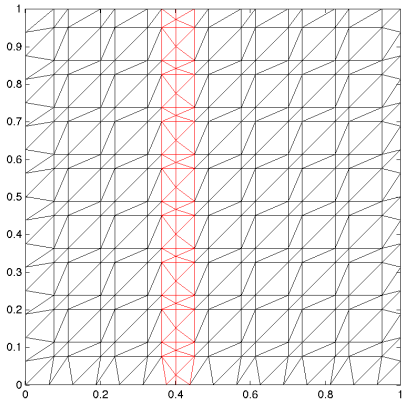
(a) Base mesh



(b) RandomFixed



(c) Wave



(d) Torture

Figure 4.2. Base and perturbed CDFEM meshes for different mesh motions, with enriched elements highlighted in red. The base mesh (a) is the initial and final mesh of each remap sequence, while the RandomFixed, Wave and Torture meshes are snapshots from each mesh motion sequence described by Equations (4.17)–(4.19).

where the artificial time variable is defined as $t = n/N$, and

$$\alpha(t) = \begin{cases} t/5, & t < 1/2, \\ (1-t)/5, & t \geq 1/2. \end{cases}$$

This smoothly displaces the mesh nodes, but in such a way as to generate highly distorted elements.

The third motion is `Torture`, which alternately shifts grid lines up and down, to cause compression and expansion of the elements in a discontinuous fashion. The motion is described by

$$\boldsymbol{x}_{i,j}^{n+1} = \begin{cases} \left[\begin{cases} x_{i,j}^n + r\Delta, & i \text{ even} \\ x_{i,j}^n - r\Delta, & i \text{ odd} \end{cases}, \begin{cases} y_{i,j}^n + r\Delta, & j \text{ even} \\ y_{i,j}^n - r\Delta, & j \text{ odd} \end{cases} \right], & n \text{ even}, \\ \boldsymbol{x}_{i,j}^0, & n \text{ odd}, \end{cases} \tag{4.19}$$

where $r = 0.2$ and Δ is the nominal mesh size as defined above.

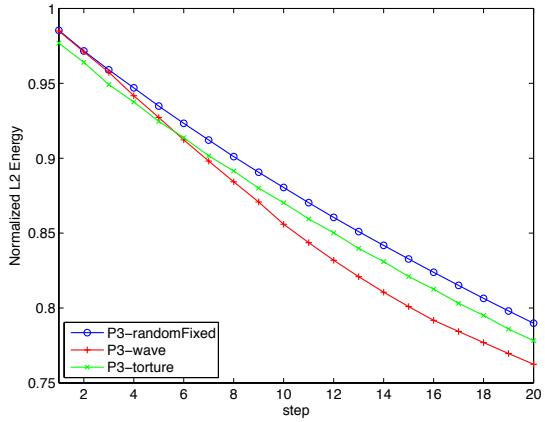
For all three mesh motions, the material interface does not move, and therefore each mesh of the remap sequence must be cut and enriched independently. Although interface nodes do not move in the x-direction, the background mesh motion causes them to move along the interface in the y-direction, which makes the remap problem nontrivial along the interface (although for (4.16), there is no discontinuity there).

The following sequence of figures will demonstrate visually the concept of constrained interpolation remap. First, using remap by zeroth-order interpolation, Figure 4.3 shows the normalized energy (L^2) norm of the nodal field through 20 remap cycles and the difference between the initial and final solutions at the last cycle. Note the scale on this plot compared to those that follow; as expected, this approach is highly dissipative.

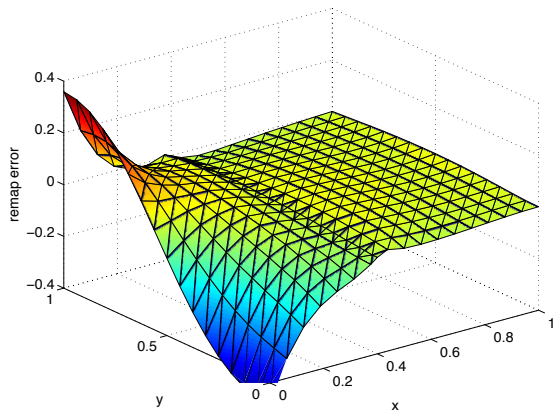
Next, using remap by linear interpolation, Figure 4.4 shows the normalized energy norm of the nodal field through 20 remap cycles and the difference between the initial and final solutions at the last cycle. In this case, while the energy under `RandomFixed` and `Wave` motions is nearly exactly preserved, it grows significantly with the `Torture` motion.

Next, using remap by interpolation with second-order patch recovery, Figure 4.5 shows the normalized energy norm of the nodal field through 20 remap cycles and the difference between the initial and final solutions at the last cycle. For this problem, the solution is potentially quite acceptable, with a relatively small amount of dissipation with each mesh motion; however, it is not difficult to find examples where the energy will grow in a manner similar to that seen using linear interpolation.

Finally, using CI remap based on zeroth- and second-order (patch recovery) interpolations, Figure 4.3 shows the normalized energy (L^2) norm of the nodal field through 20 remap cycles and the difference between the initial and final solutions at the last cycle. This clearly gives the best result, with minimal dissipation that maintains the solution energy near the initial value through multiple remap cycles by balancing dissipation from the low-order approximation with the accuracy of the high-order interpolation.

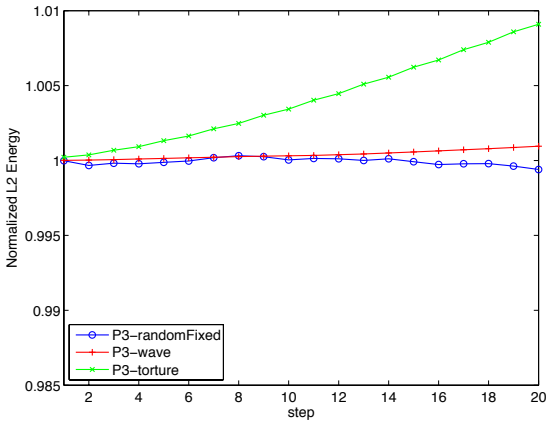


(a) Energy norm

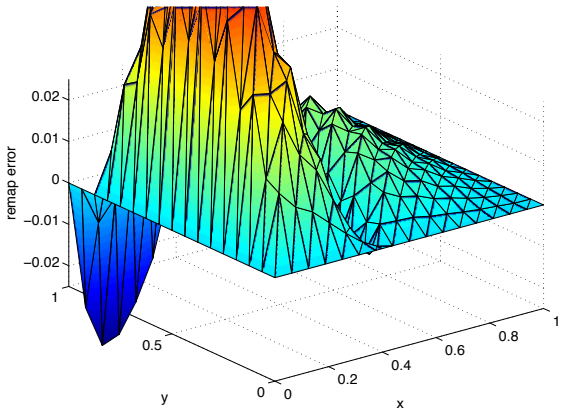


(b) Remap error

Figure 4.3. Normalized L^2 energy norm with remap cycle and the difference between initial and final solutions following 20 remap cycles, using zeroth-order interpolation remap.

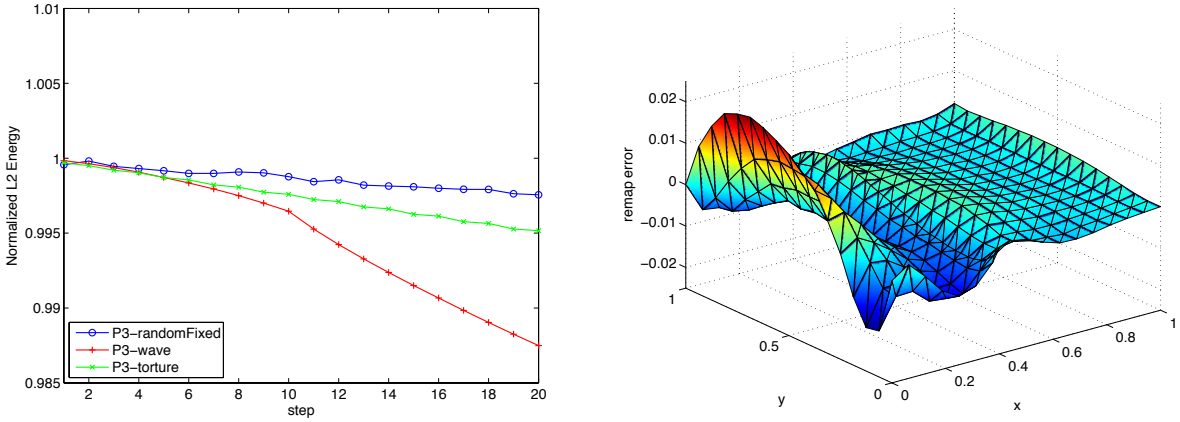


(a) Energy norm



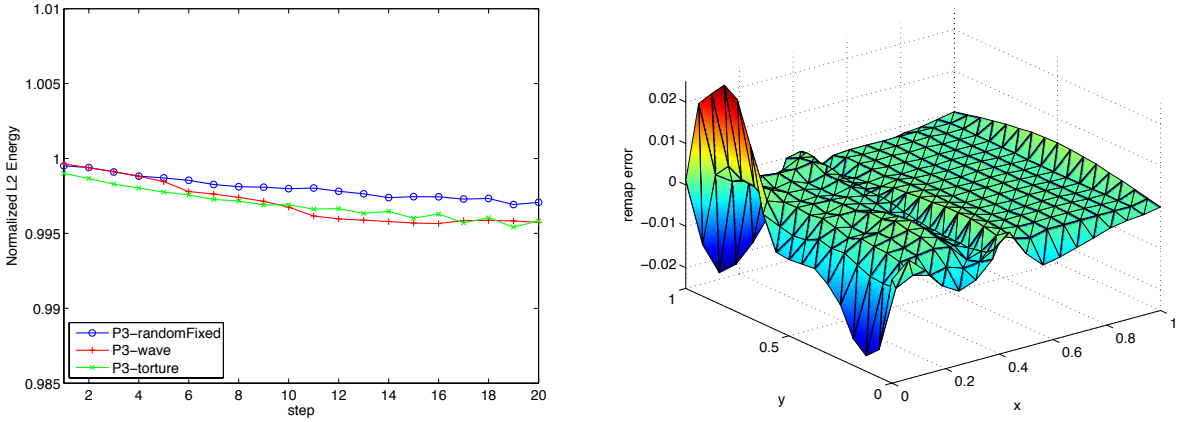
(b) Remap error

Figure 4.4. Normalized L^2 energy norm with remap cycle and the difference between initial and final solutions following 20 remap cycles, using linear interpolation remap.



(a) Energy norm (b) Remap error

Figure 4.5. Normalized L^2 energy norm with remap cycle and the difference between initial and final solutions following 20 remap cycles, using interpolation with second-order patch recovery.



(a) Energy norm (b) Remap error

Figure 4.6. Normalized L^2 energy norm with remap cycle and the difference between initial and final solutions following 20 remap cycles, using CI remap.

4.4 Examples

The following set of examples demonstrate constrained interpolation remap in 2-D for both nodal and edge-based field discretizations, with different reference norms. The focus of these results is to demonstrate preservation of convergence rates with mesh refinement following multiple remap cycles.

4.4.1 Nodal B-form

Reduction of the eddy current equations to 2-D in the transverse electric regime assumes that

$$\mathbf{E} = [E_x(x, y, t), E_y(x, y, t), 0],$$
$$\mathbf{B} = [0, 0, B_z(x, y, t)].$$

Choosing B_z as the field of interest, the eddy current equations reduce to the nodal formulation (2.1). The norm of interest for remap is the magnetic energy, $\|B_z\|^2$, in this case simply the L^2 norm of the nodal field. The remapped solution was constructed from zeroth- and second-order interpolated fields, which provided adequate dissipation in the solution over time.

The first example, shown in Figure 4.7, remaps the continuous function

$$B_z(x, y) = \cos(\pi y)(\sinh(\pi x) - \coth(\pi)\cosh(\pi x)), \tag{4.20}$$

over 20 cycles. An interface is placed at $x = 0.4$, although the solution is C^∞ continuous there. Each of the mesh motions (4.17)–(4.19) are shown, and the convergence plots indicate that the expected second order in L^2 and first order in H^1 are recovered.

The second example uses a manufactured solution to (2.1), with a slope discontinuity across a circular interface:

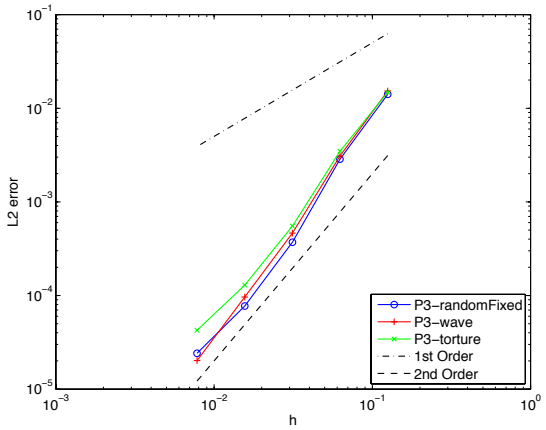
$$B_z(r) = \begin{cases} 2 - r^2/\sigma_I, & r < r_c, \\ 1 - r^2, & r > r_c. \end{cases}, \tag{4.21}$$

where $r = \sqrt{(x - 1/2)^2 + (y - 1/2)^2}$, $\tau = 1$, and

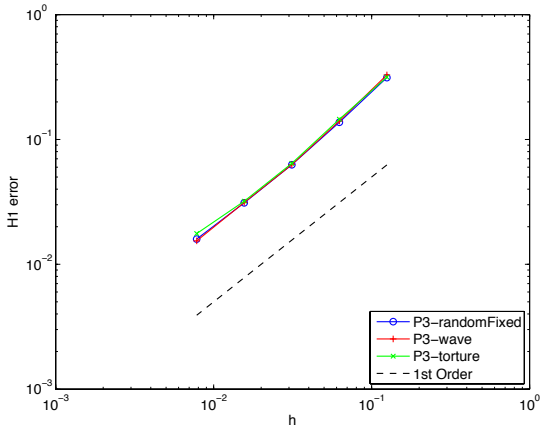
$$\sigma = \begin{cases} \sigma_I, & r < r_c, \\ 1, & r > r_c, \end{cases}$$

with $\sigma_I = r_c^2/(1 + r_c^2)$. For Figure 4.8, the interface is located at $r_c = 1/3$. This example is taken from [6], where it was demonstrated that the proper convergence rates could be achieved only when the discontinuity is captured.

Both examples show that the expected convergence rates are achieved in both the L^2 norm and the H^1 seminorm with constrained interpolation remap. Note that these results use the analytic reference norm, and results suffer slightly (due to additional dissipation) when the source field is integrated approximately on the destination mesh.

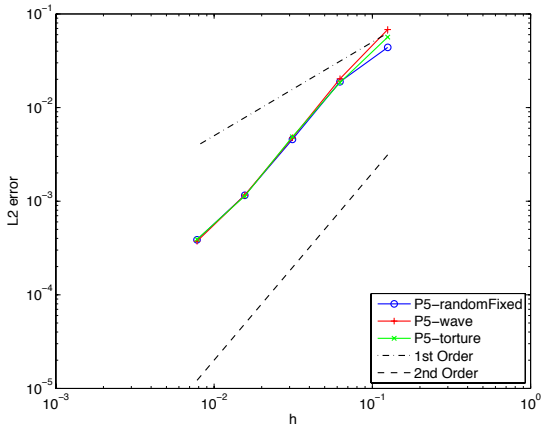


(a) $\|B_z - B_h^N\|_{L_2}$

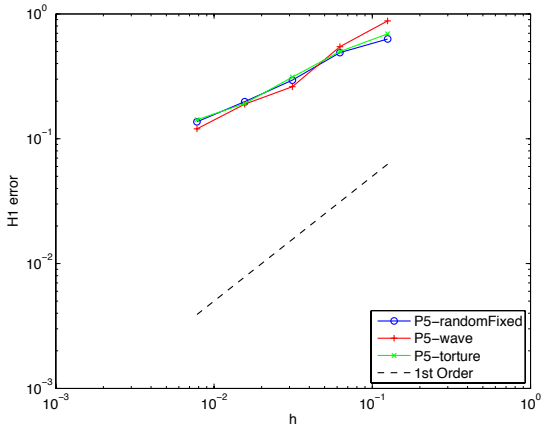


(b) $\left|B_z - B_h^N\right|_{H_1}$

Figure 4.7. Convergence plots for the nodal B-form problem with a linear interface.



(a) $\|B_z - B_h^N\|_{L_2}$



(b) $\left|B_z - B_h^N\right|_{H_1}$

Figure 4.8. Convergence plots for the nodal B-form problem with a circular interface.

4.4.2 Nodal A-form

In the 2-D transverse magnetic regime, the eddy current equations are simplified by assuming

$$\mathbf{B} = [B_x(x, y, t), B_y(x, y, t), 0].$$

In this case, a nodal scalar potential $A(x, y)$ may be defined such that $\mathbf{B} = \nabla A$. The potential function is then remapped, which ensures that $\nabla \cdot B \equiv 0$, but preservation of the magnetic energy implies that the norm of interest for remap is the gradient of A, $|(\nabla A)|^2$, i.e., the H^1 seminorm of the nodal field. The remapped solution is constructed from first- and second-order interpolated fields, because the gradient of the zeroth-order approximation is too inaccurate to be useful.

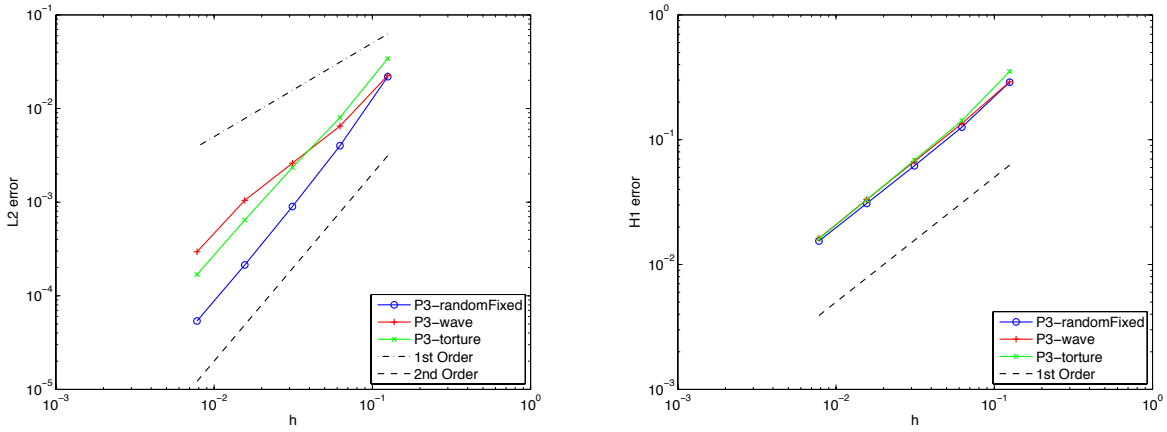
For this demonstration, the same functions are tested, but now

$$A(x, y) = \cos(\pi y)(\sinh(\pi x) - \coth(\pi)\cosh(\pi x)), \tag{4.22}$$

with the cut at $x = 0.4$ for Figure 4.9, and

$$A(r) = \begin{cases} 2 - r^2/\sigma_I, & r < r_c, \\ 1 - r^2, & r > r_c, \end{cases}, \tag{4.23}$$

with the cut at $r = 1/3$ for Figure 4.10. As with the B-form, the expected convergence rates are achieved in both the L^2 norm and the H^1 seminorm. Both examples use the analytic reference norm.

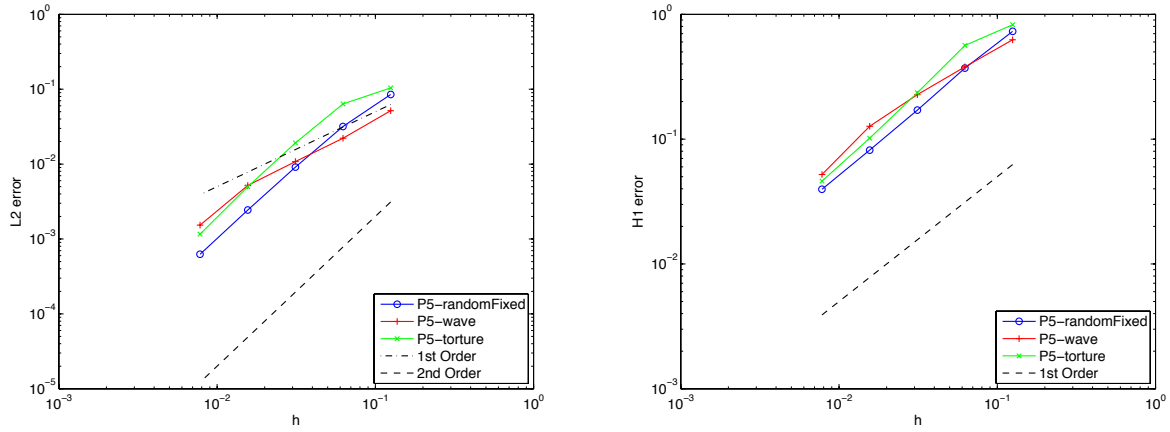


(a) $\|A - A_h^N\|_{L_2}$ (b) $\left|A - A_h^N\right|_{H_1}$

Figure 4.9. Convergence plots for the nodal A-form problem with a linear interface.

4.4.3 Edge Discretization

An alternative formulation for the transverse electric regime is a compatible edge discretization of the electric field, as in [7]. Here, for compatibility with the nodal A-form, we consider



(a) $\|A - A_h^N\|_{L_2}$ (b) $\left|A - A_h^N\right|_{H_1}$

Figure 4.10. Convergence plots for the nodal A-form problem with a circular interface.

the face-based (i.e., element, in 2-D) B-field and define an edge-based vector potential

$$\mathbf{A} = [A_x(x, y), A_y(x, y), 0], \tag{4.24}$$

such that $\nabla \times \mathbf{A} = \mathbf{B}$. The potential function is then remapped while preserving the magnetic energy, i.e., the $H(curl)$ seminorm of the edge potential field, $|(\nabla \times \mathbf{A})|^2$. The remapped solution was constructed from first- and second-order interpolated fields, as in the nodal A-form case.

For this demonstration, the CDFEM edge formulation from [7] is used to enrich the mesh in cut elements. Two vector fields are used to demonstrate remap for edge elements. First, in Figure 4.11, the vector potential is given by

$$\mathbf{A} = \left[(1 - 8x + 2y + 8xy)/24, \left(-1 + 2x + 4y - 8x^2\right)/24, 0\right], \tag{4.25}$$

such that $\mathbf{B} = [0, 0, -x]$. Note that this particular form of the vector potential was chosen because it can be exactly represented using the incomplete polynomial basis of second-order Nédélec edge elements (although it is discretized using first-order elements here). Remap is demonstrated with a linear interface at $x = 4/7$. The second problem, shown in Figure 4.12, is a manufactured solution to the eddy current equations with a discontinuity across the same interface, taking the form

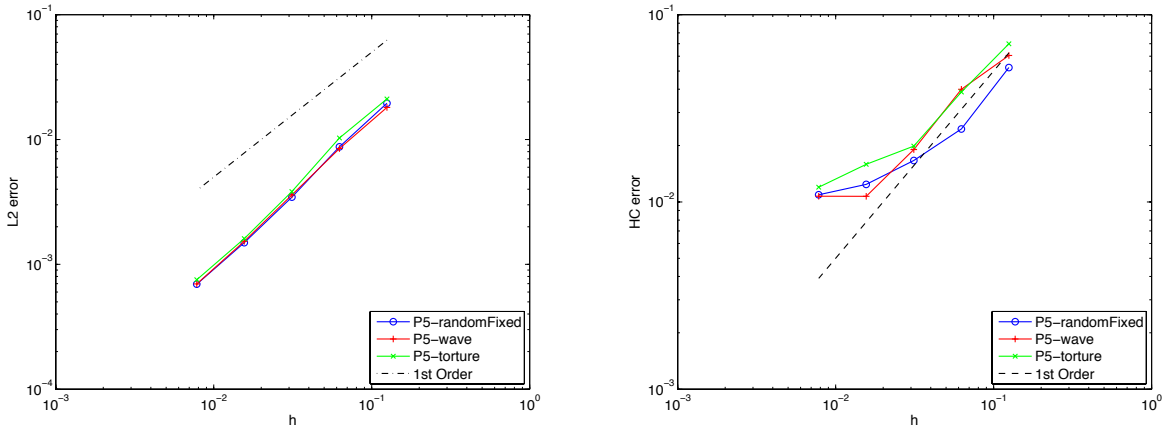
$$\mathbf{A} = \begin{cases} [-y(\beta x - q), 0, 0], & x < x_c, \\ [-y(\beta x_c - q + (1 - \beta x_c)(x - x_c)/(1 - x_c)), 0, 0], & x \geq x_c, \end{cases} \tag{4.26}$$

where $x_c = 4/7$, $q = (\beta x_c - x_c + 1)/2$, $\beta = 4/(4x_c + (1 - x_c))$, and

$$\sigma = \begin{cases} 4, & x < 4/7, \\ 1, & x \geq 4/7. \end{cases}$$

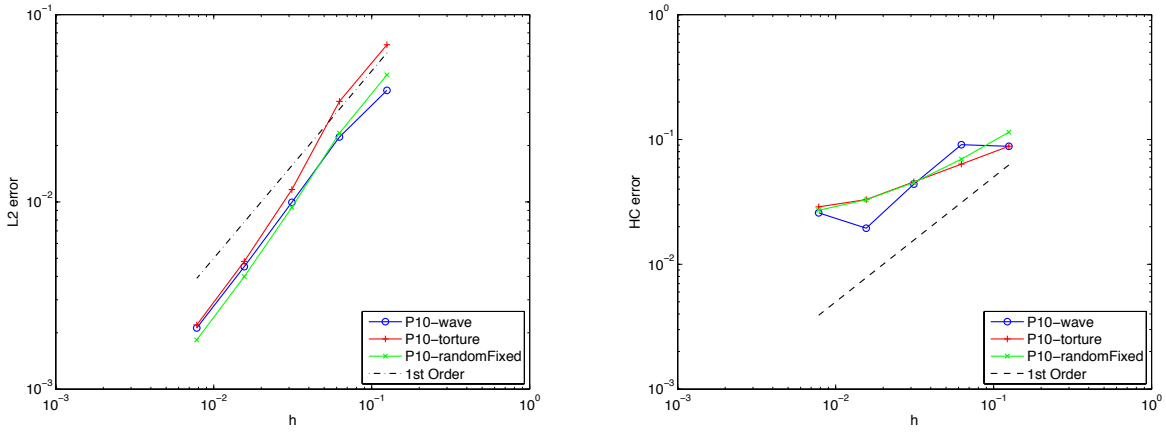
Both examples use the analytic reference norm for CI remap.

For both examples, the expected first-order rate of the edge elements is recovered in the L^2 norm after 20 remap cycles with any mesh motion. However, results are not as good in the $H(curl)$ seminorm, where some degradation is observed and the expected first-order rate is not seen. This happens regardless of the continuity of the solution. The cause is not yet known, but could be related to the patch recovery or the patch averaging that is used to distribute element-wise optimization parameters to the edges. Further work will be required to investigate this issue.



(a) $\|\mathbf{A} - \mathbf{A}_h^N\|_{L_2}$ (b) $\left|\mathbf{A} - \mathbf{A}_h^N\right|_{H(curl)}$

Figure 4.11. Convergence plots for the continuous edge remap problem.



(a) $\|\mathbf{A} - \mathbf{A}_h^N\|_{L_2}$ (b) $\left|\mathbf{A} - \mathbf{A}_h^N\right|_{H(curl)}$

Figure 4.12. Convergence plots for the discontinuous edge remap problem.

Chapter 5

Conclusions and the Path Forward

We have introduced a family of methods which allow us to resolve material interfaces at the sub-element level. For the XFEM-AC and CDFEM cases, we have demonstrated that these discretizations can be used to solve diffusion-like problems on nodes, edges and faces in 2D and 3D. We have also demonstrated an alternative approach for nodal and edge problems based on h-hierarchical methods. Finally, we have demonstrated results for constrained interpolation remap of nodal and edge-based fields in 2D.

This leaves a number of questions for future research. First, the question of conservation in the CI remap algorithms needs to be addressed in more depth. It is important for many applications that magnetic energy not increase through the remap process. The nodal formulation is more mature than the edge problem in this regard, but further work on this aspect of the algorithm is needed in particular. Other details also remain to be investigated to demonstrate the robustness, stability, and accuracy of the CI remap method for the problems of interest. Second, similar experiments to those described in Chapter 4 need to be performed in 3D. Third, the remap and diffusion algorithms need to be demonstrated in mixed (e.g. quad/tri in 2D or hex/pyramid/prism/tet in 3D) meshes, as these would better correspond to the environment seen in ALEGRA. Further research must also be performed in interface reconstruction algorithms to ensure that the interface given to the XFEM-AC or CDFEM algorithm does not have spurious discontinuities across element boundaries.

References

[1] Ivo Babuska. The finite element method for elliptic equations with discontinuous coefficients. Computing, 5:207–213, 1970.

[2] P. Bochev and M. Shashkov. Constrained interpolation (remap) of divergence-free fields. Comput. Methods Appl. Mech. Engrg., 194:511–530, 2005.

[3] J. Dolbow, N. Moës, and T. Belytschko. Discontinuous enrichement in finite elements with a partition of unity method. Finite Element Methods in Analysis and Design, 36:235–260, 2000.

[4] M. Hautefeuille, C. Annavarapu, and J. E. Dolbow. Robust imposition of dirichlet boundary conditions on embedded surfaces. International Journal for Numerical Methods in Engineering, 90:40–64, 2012.

[5] X. Jiao and M. T. Heath. Common-refinement-based data transfer between non-matching mehes in multiphysics simulations. Int. J. Numer. Meth. Engng., 61:2402–2427, 2004.

[6] R. Kramer, P. Bochev, C. Siefert, and T. Voth. An extended finite element method with algebraic constraints (XFEM-AC) for problems with weak discontinuities. Comput. Methods Appl. Mech. Engrg., 266:70–80, 2013.

[7] R. Kramer, P. Bochev, C. Siefert, and T. Voth. Algebraically constrained extended edge element method (**e**XFEM-AC) for resolution of multi-material cells. J. Comp. Phys., 266:596–612, 2014.

[8] R. Kramer, P. Bochev, C. Siefert, and T. Voth. Algebraically constrained extended finite elements (XFEM-AC) and conformal decomposition finite elements (CDFEM) for 3D compatible discretizations. In preparation, 2014.

[9] L. G. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. J. Comp. Phys., 184:266–298, 2003.

[10] N. Moës, E. Bèchet, and M. Tourbier. Imposing Dirichlet boundary conditions in the extended finite element method. International Journal for Numerical Methods in Engineering, 67:1641–1669, 2006.

[11] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. Int. J. Numer. Meth. Engng., 46:131–150, 1999.

[12] D. R. Noble, E. P. Newren, and J. B. Lechman. A conformal decomposition finite element method for modeling stationary fluid interface problems. Int. J. Numer. Meth. Fluids, 63:725–742, 2010.

[13] A. Robinson, T. Brunner, S. Carrol, R. Drake, C. Garasi, T. Gardiner, H. Hanshaw, D. Hensinger, D. Labreche, R. Lemke, E. Love, C. Luchini, S. Mosso, J. Niederhaus, C. Ober, S. Petney, W. Rider, G. Scovazzi, O. Strack, R. Summers, T. Trucano, V. Weirs, M. Wong, and T. Voth. ALEGRA: an arbitrary Lagrangian-Eulerian multimaterial, multiphysics code. In Proceedings of the 46th AIAA aerospaces sciences meeting, 2008.

[14] A.C. Robinson, J.H.J. Niederhaus, V.G. Weirs, and E. Love. Arbitrary lagrangian-eulerian 3d ideal MHD algorithms. International Journal for Numerical Methods in Fluids, 65:1438–1450, 2011.

[15] J. D. Sanders, J. E. Dolbow, and T. A. Laursen. On methods for stabilizing constraints over enriched interfaces in elasticity. International Journal for Numerical Methods in Engineering, 78:1009–1036, 2008.

[16] A. Simone. Partition of unity-based discontinous elements for interface phenomena: computational issues. Communications in Numerical Methods in Engineering, 20:465–478, 2004.

[17] S. Soghrati, A. M. Aragon, C. A. Duarte, and P. H. Geubelle. An interface-enriched generalized fem for problems with discontinuous gradient fields. International Journal for Numerical Methods in Engineering, 89:991–1008, 2012.

[18] S. Soghrati and P. H. Geubelle. A 3d interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries. Computer Methods in Applied Mechanics and Engineering, pages 46–57, 2012.

DISTRIBUTION:

1	MS 0840	James Cox, 1554
1	MS 1320	Pavel Bochev, 1442
1	MS 1321	Glen Hansen, 1443
1	MS 1322	Christopher Siefert, 1443
1	MS 1323	Richard Kramer, 1443
1	MS 1323	Thomas Voth, 1443
1	MS 1323	Allen Robinson, 1443
1	MS 0359	D. Chavez, LDRD Office, 1911
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories