

EFFICIENCY BASED ADAPTIVE LOCAL REFINEMENT FOR FIRST-ORDER SYSTEM LEAST-SQUARES FORMULATIONS*

J. H. ADLER[†] T. A. MANTEUFFEL[‡] S. F. MCCORMICK[‡] J. W. NOLTING[§] J. W. RUGE[‡]
AND L. TANG[‡]

Abstract. In this paper, we propose new adaptive local refinement (ALR) strategies for first-order system least-squares (FOSLS) finite element in conjunction with algebraic multigrid (AMG) methods in the context of nested iteration (NI). The goal is to reach a certain error tolerance with the least amount of computational cost and nearly uniform distribution of the error over all elements. To accomplish this, the refinement decision at each refinement level is determined based on optimizing efficiency measures that take into account both error reduction and computational cost. Two efficiency measures are discussed, predicted error reduction and predicted computational cost. These methods are first applied to a 2D Poisson problem with steep gradients and the results are compared with the threshold-based methods described in [16]. Next, these methods are applied to a 2D reduced model of the incompressible, resistive magnetohydrodynamic (MHD) equations. These equations are used to simulate instabilities in a large aspect-ratio tokamak. We show that, by using the new ALR strategies on this system, we are able to resolve the physics using only 10 percent of the computational cost used to approximate the solutions on a uniformly refined mesh within the same error tolerance.

Key words. adaptive local refinement; AMG; FOSLS; nested iteration; magnetohydrodynamics

AMS subject classifications. 65F10, 65N55, 65N50, 76W05

Acknowledgments. This work was sponsored by the Department of Energy under grant numbers DE-FG02-03ER25574 and DE-FC02-06ER25784, Lawrence Livermore National Laboratory under contract numbers B568677, and the National Science Foundation under grant numbers DMS-0621199, DMS-0749317, and DMS-0811275.

1. Introduction. Adaptive finite element methods (AFEMs) are being used extensively to approximate solutions of partial differential equations (PDEs) containing local features; see, e.g., [18, 26, 31, 16, 3]. Consider a partial differential equation (PDE), or a system of PDEs, written abstractly as

$$\mathcal{P}\mathbf{w} = \mathbf{f} \quad \text{in } \Omega \subset \mathbb{R}^d, \quad (1.1)$$

with appropriate boundary conditions. Let \mathcal{T} be a regular partition [7] of the domain, Ω , into elements. Define the mesh size, $h = \max\{\text{diam}(\tau), \tau \in \mathcal{T}\}$. In general, the refinement process starts on a coarse grid, \mathcal{T}_0 (level = 0), and iteratively refines and approximates the PDE on levels $\ell = 1, 2, \dots$ until the error satisfies a certain criteria. At each level, some elements are refined in h by splitting them into sub-elements, and some are refined in p by increasing the element order. In [22, 23], this concept is described in the following form:

$$\text{Solve} \rightarrow \text{Estimate} \rightarrow \text{Mark} \rightarrow \text{Refine}. \quad (1.2)$$

*Submitted to SIAM journal on Scientific Computing, February 23, 2010

[†]Department of Mathematics, Pennsylvania State University, University Park, State College, PA 16802 (adler@math.psu.edu).

[‡]Department of Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526 (tmanteuf@colorado.edu, stevem@colorado.edu, jruge@colorado.edu, tangl@colorado.edu).

[§]GeoEye Inc, 12076 Grant St, Thornton, CO 80241 (nolting.josh@geoeye.com).

One goal of adaptive refinement is to construct a sequence of nearly optimal grids, meaning that a certain error is achieved with a minimum number of elements (or degrees of freedom (DOF)). In one dimension, it has been proved that this is accomplished by equally distributing the error over all elements; [18]. It is believed that this also holds for higher dimensions. Based on this premise, a simple method to mark element for refinement was introduced by Babuška [18]: an element is marked for refinement if its local error is within a certain factor of the largest local error at that level. In [16], a more complicated algorithm was proposed:

ALGORITHM 1 (Threshold-based Marking). *Given a parameter $0 < f \leq 1$, construct a minimal subset $\hat{\mathcal{T}}$ of \mathcal{T} such that*

$$\sum_{\tau \in \hat{\mathcal{T}}} \epsilon_{\tau}^2 \geq f \sum_{\tau \in \mathcal{T}} \epsilon_{\tau}^2, \quad (1.3)$$

and mark all elements in $\hat{\mathcal{T}}$ for refinement.

The AFEMs in [22, 23] start with this approach, then further mark elements based on oscillation terms. This marking approach often produces satisfactory results. With a proper choice of f , one can establish the convergence of the AFEM as well as optimality of the finest grid. However, the real computational cost was not addressed. Also, the proper choice of f is different for various problems and unknowns *a priori*. In order to achieve the optimal grid with the least amount of work, we expect that the value of f would require freedom for it to change on each level. We may need to refine a large fraction of elements at the coarser levels when the grids are too coarse to resolve the local features of the solution. Then, at intermediate levels, refinement should concentrate on the elements containing relatively large error. Lastly, at finer levels, once the error is equally distributed, near global refinement is preferred.

A new approach, described in [6], was developed to address this issue. The algorithm refines elements that minimize a ‘work-times-error-reduction’ efficiency factor (WEE) at each refinement level. Later, in [15], it was shown that the WEE algorithm was inefficient for problems with spatial dimension, d , less than the polynomial degree, p , of the finite element space. Another algorithm, which determined the fraction of elements to be refined, r , by optimizing the ‘accuracy-per-computational-cost’ efficiency factor (ACE) was proposed and analyzed mainly in one dimension [15]. The results show that the ACE algorithm is capable of effectively and efficiently detecting the solution’s local features. In this paper, we extend the ACE algorithm to two and three dimensions for first-order system least-square (FOSLS) finite element methods. The resulting linear systems are solved using algebraic multigrid (AMG) in the context of nested iteration (NI). Two variations of the ACE algorithm are also proposed. The first requires a fixed increase of DOF. The second forces a fixed anticipated error reduction. This is similar to the threshold-based refinement algorithm except that an element is allowed to be refined more than once at a single level. Finally, another algorithm, which minimizes the ‘anticipated-overall-computational-work’ efficiency factor (NACE) is developed.

NI-FOSLS-AMG yields measures that allow us to estimate the anticipated error reduction and computational cost, which can be used to make the refinement decisions based on optimizing computational efficiency. The performance of the efficiency-based adaptive refinement algorithm, applied to NI-FOSLS-AMG, is compared for a 2D Poisson problem with steep gradients and a time-dependent nonlinear magneto-hydrodynamics (MHD) problem. The numerical results show that all of the ACE

algorithms used with NI-FOSLS-AMG are capable of approximating the solutions within the same error tolerance with much less computational cost than the threshold-based refinement and uniform refinement. Although we propose the efficiency-based refinement algorithms for the NI-FOSLS-AMG approach, they can also be applied to other FEMs and linear solvers, provided that a locally sharp and globally reliable a posteriori error estimator is available.

This paper is organized as follows. In Section 2, the basic concepts of the NI-FOSLS-AMG approach are described. The notation used in this paper is also introduced. The efficiency-based refinement strategies are described in Section 3. Then, the performance of the ACE and NACE type strategies for the 2D test problems are discussed in Section 4. Next, in Section 5, results on convergence of adaptive refinement for FOSLS are discussed. Section 6 describes how the efficiency-based refinement strategies can be implemented in parallel. Finally, conclusions are formulated in Section 7.

2. Preliminaries. In this section, we briefly describe the basic concepts behind the NI-FOSLS-AMG approach and introduce notations used in the rest of the paper.

2.1. FOSLS methodology. First-order system least squares (FOSLS) is a special type of finite element method that reformulates a PDE as a system of first-order equations and poses the problem as a minimization of a functional. Here, the first-order differential terms appear quadratically and, thus the functional norm is equivalent to a norm meaningful to the problem. To illustrate the basic concepts of FOSLS, consider the PDE written abstractly in (1.1). Introducing new variables, we arrive at a first-order system:

$$\mathcal{L}_i \mathbf{u} = f_i, \quad i = 1, 2, \dots, M. \quad (2.1)$$

Assuming $f_i \in L^2(\Omega)$, consider the associated FOSLS functional given by

$$\mathcal{G}(\mathbf{u}, \mathbf{f}) = \sum_{i=1}^M \|\mathcal{L}_i \mathbf{u} - f_i\|_{0,\Omega}^2, \quad (2.2)$$

where $\|u\|_{0,\Omega} = \sqrt{\int_{\Omega} |u|^2}$ is the L^2 -norm. The minimization problem is

$$\mathbf{u} = \arg \min_{\mathbf{v} \in \mathcal{V}} \mathcal{G}(\mathbf{v}; \mathbf{f}). \quad (2.3)$$

Here, \mathcal{V} is an appropriate Hilbert space, usually (equivalent to) a product of H^1 spaces. In many cases, under general regularity assumptions, $\mathcal{G}(\mathbf{u}; \mathbf{f})$ is “elliptic” with respect to the \mathcal{V} norm; see, e.g., [11, 12]. That is, its homogeneous part, $\mathcal{G}(\mathbf{v}; 0)$, is equivalent to the squared \mathcal{V} norm:

$$c_1 \leq \frac{\mathcal{G}(\mathbf{v}; 0)}{\|\mathbf{v}\|_{\mathcal{V}}^2} \leq c_2 \quad (2.4)$$

for some positive constants c_1 and c_2 and for every $\mathbf{v} \in \mathcal{V}$. This ellipticity enables the existence and uniqueness of the solution \mathbf{u} . Let $\mathcal{V}^h \subset \mathcal{V}$ be a finite-dimensional subspace of \mathcal{V} , often it consists of continuous piecewise polynomials. Note that the discretization can be written as the minimization problem

$$\mathbf{u}^h = \arg \min_{\mathbf{v}^h \in \mathcal{V}^h} \mathcal{G}(\mathbf{v}^h; \mathbf{f}). \quad (2.5)$$

Well-posedness of (2.5) follows directly from the ellipticity. Therefore, the FOSLS formulation is not restricted by any LBB condition. While not a necessary condition, if \mathcal{V} is a product of H^1 spaces, then ellipticity also enables an optimal multigrid solver of the discrete system [12], that is, standard multigrid solvers converge with factors bounded uniformly in mesh size, h .

The introduction of the new dependent variables increases the number of DOF, much like mixed finite element methods. However, unlike mixed methods, FOSLS yields a symmetric positive definite algebraic system that is, in general, more amenable to multilevel solution techniques.

2.2. A Sharp and Reliable A-posteriori Error Estimate. The FOSLS functional provides a unique capability for adaptive refinement: an effective error indicator at no additional computation cost. Because the functional value is zero at the solution, the FOSLS functional itself is a measure of the total error in a given approximation. It provides both absolute and relative error measures, as well as global and local error estimates, that are much simpler and potentially sharper than conventional error estimators. To illustrate this, for any element $\tau \in \mathcal{T}$, define the local FOSLS functional

$$\mathcal{G}_\tau(\mathbf{u}^h; \mathbf{f}) = \sum_{i=1}^M \|\mathcal{L}_i \mathbf{u}^h - f_i\|_{0,\tau}^2. \quad (2.6)$$

Writing $\epsilon_\tau = \sqrt{\mathcal{G}_\tau(\mathbf{u}^h; \mathbf{f})}$, the ellipticity in (2.4) implies that

$$\frac{1}{c_2} \epsilon_\tau^2 = \frac{1}{c_2} \mathcal{G}_\tau(\mathbf{u}^h - \mathbf{u}; 0) \leq \|\mathbf{u}^h - \mathbf{u}\|_{\mathcal{V},\tau}^2 \quad (2.7)$$

and

$$\|\mathbf{u}^h - \mathbf{u}\|_{\mathcal{V}}^2 \leq \frac{1}{c_1} \mathcal{G}(\mathbf{u}^h - \mathbf{u}; 0) = \frac{1}{c_1} \sum_{\tau \in \mathcal{T}} \epsilon_\tau^2. \quad (2.8)$$

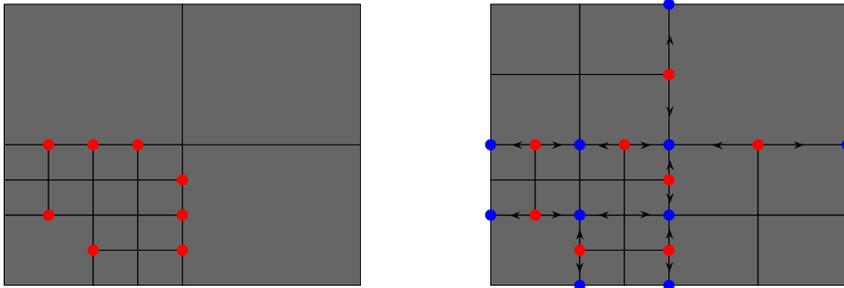
An error estimator, ϵ_τ , that satisfies an inequality of type (2.7) is called locally sharp. It implies that if ϵ_τ is large, then the error is large within that element. In the literature, an inequality of type (2.8) is called a reliability bound; see [31]. Note that a small sum of local estimators, ϵ_τ , implies a small global error.

2.3. Nested Iteration and Algebraic Multigrid. Nested iteration (NI), or full-multigrid [10] (FMG) as it is called in the multigrid context, involves starting the solution process on a relatively coarse grid, where the computational cost is relatively cheap. The solution on the coarse grid is used as an initial guess for the problem on the next finer grid. Since the objective on each grid is to minimize the FOSLS functional, the coarse-grid solution should provide a good starting guess. On each refinement level, solving discrete minimization problem (2.5) involves fast iterative solvers applied to the matrix equations. If the FOSLS functional is equivalent to a product H^1 norm, then there exists an optimal multilevel solution algorithm [32]. Experience shows that, in this context, AMG also yields an approximate solution to the discrete equations associated with quasi-uniform grids in optimal time with convergence factor, ρ , bounded uniformly below 1, independent by mesh size h . AMG methods, together with the NI strategy and local refinement, provide a powerful approach for approximating solutions of PDEs. Numerical and theoretical results confirm that the overall cost of such a scheme resides predominantly in the cost of

the finest-level processing. The total cost is usually cheaper than solving the problem directly on the finest grid, which generally is not even known in advance.

The NI strategy presents a special opportunity for AMG methods. In general, AMG requires a substantial setup phase. The NI approach with local refinement yields a hierarchy of quasi-nested block-structured grids, that is, the coarsest grid may be irregular, but subsequent grids are increasingly more structured. This hierarchy, together with AMG, may reduce or eliminate the need for a setup phase at each level. This will be investigated in future work.

2.4. Refinement. In this section, we discuss the method for subdividing elements into subelements. The FOSLS methodology allows us to use simple bisection of elements. Here we describe the algorithm in the context of quadrilaterals in two dimensions and hexahedral elements in three dimensions. The quadrilateral elements are partitioned into four sub-elements of equal area, and the hexahedral elements are partitioned into eight sub-elements of equal volume. In the AFEM context, triangles (tetrahedron in three dimensions) are often employed as finite elements, which requires more attention in the subdivision stage to ensure conforming elements. The newest vertex routine in conjunction with simple bisection is often used there; see [23]. Although the simple methods we employ produce hanging nodes, the FOSLS method handles this situation easily. Hanging nodes are nodes along element edges or faces, in which the edge or face is shared by multiple elements, and the nodes are not defined for some elements; see Fig. 2.1. If conforming elements for FOSLS discretization are



(a) Possible grid after refinement without clean-up stage (b) grid after a clean-up stage is used to maintain 2-to-1 balance

FIG. 2.1. A mesh resulting in slave nodes and master nodes; the red solid circle marks slave nodes, the blue solid circle marks master nodes. The arrows represent explicit correlations between slave and master nodes.

desired, each slave node is dependent on its master nodes at the endpoints of the edge or the face on which it is hanging through an explicit algebraic constraint. Explicit correlations between slave nodes and master nodes are established, see Fig. 2.1(b).

Although not required for FOSLS convergence, we perform an additional clean-up stage in our implementation so that two adjacent elements sharing an edge or a face should not differ in edge size by a factor greater than 2. Such a constraint is often referred to as the balance condition or 2-to-1 balance constraint [30, 29]. This clean-up stage helps to minimize the number of slave nodes, which can in turn improve AMG convergence. Also, it helps track local features that are not stationary during a time-dependent simulation.

Remark. Although not pursued here, FOSLS is particularly amenable to non-conforming finite element spaces. See Bernt's lemma (Theorem 5.2) [5]

3. Efficiency-based Adaptive Local Refinement for NI-FOSLS-AMG.

Recall that the goal of efficiency-based refinement strategies is to reach a certain error tolerance with the least amount of computational cost. Given a potential refinement strategy, one needs to estimate

- the reduction in the functional norm of the error and
- the computational cost of solving the resulting system of equations.

These estimates are established in the next two sections.

3.1. FOSLS Approximation Heuristics. Assume the solution space, \mathcal{V} , is a product of $H^1(\Omega)$ Sobolev spaces. For any tessellation, \mathcal{T}_h , with mesh size, h , let \mathcal{V}^h be the finite-dimensional subspace consisting of continuous piecewise polynomials of degree p . Define $I_h \mathbf{u}$ to be the interpolant of the exact solution, \mathbf{u} , into the subspace \mathcal{V}^h . Then, there exists a constant, C , independent of \mathbf{u} , such that

$$\|I_h \mathbf{u} - \mathbf{u}\|_1 \leq Ch^s |\mathbf{u}|_{s+1} \quad (3.1)$$

for $0 < s \leq p$. Here, $\|\cdot\|_1$ is the $H^1(\Omega)$ -norm and $|\cdot|_{s+1}$ is the $H^{s+1}(\Omega)$ semi-norm, (c.f.[7]). We further assume that the solution, \mathbf{u} , is smooth enough, i.e., $\mathbf{u} \in H^{p+1}(\Omega)$, so that (3.1) is valid for $s = p$. The following error bound is used to estimate the functional reduction:

$$\begin{aligned} \mathcal{G}(\mathbf{u}^h; \mathbf{f}) &:= \sum_{\tau \in \mathcal{T}_h} \mathcal{G}_\tau(\mathbf{u}^h; \mathbf{f}) \leq \sum_{\tau \in \mathcal{T}_h} \mathcal{G}_\tau(I_h \mathbf{u}; \mathbf{f}) \\ &\leq c_2 \|I_h \mathbf{u} - \mathbf{u}\|_1^2 \\ &\leq c_2 C^2 h^{2p} |\mathbf{u}|_{p+1}^2. \end{aligned} \quad (3.2)$$

Similar bound holds for the local interpolate error:

$$\begin{aligned} \hat{\epsilon}_\tau^2 &:= \mathcal{G}_\tau(I_h \mathbf{u}; \mathbf{f}) \leq Dh_\tau^{2p} |\mathbf{u}|_{p+1, \tau}^2 \\ &\leq Dh_\tau^{2p} M_{p+1, \tau} H_\tau, \end{aligned} \quad (3.3)$$

where D is independent of \mathbf{u} and h_τ , H_τ is the area of element τ , and $M_{p+1, \tau} H_\tau$ is a bound on $|\mathbf{u}|_{p+1, \tau}^2$. We assume that D and $M_{p+1, \tau}$ are relatively constant over element τ . Moreover, we assume $I_h \mathbf{u}$ is close enough to \mathbf{u}^h so that bound (3.3) holds for local FOSLS functionals:

$$\epsilon_\tau^2 := \mathcal{G}_\tau(\mathbf{u}^h; \mathbf{f}) \approx \mathcal{G}_\tau(I_h \mathbf{u}; \mathbf{f}) \leq Dh_\tau^{2p} M_{p+1, \tau} H_\tau. \quad (3.4)$$

Modifications of the assumptions might be necessary for certain situations, for example, when the solution contains a singularity (c.f. [15]) or the grid is not fine enough to resolve features of the solution. In such cases, these assumptions can be adaptively monitored so that run-time adjustments can be made.

If element τ_i is split in two in each dimension, then we have 2^d new elements $\tau_{i,1}, \dots, \tau_{i,2^d}$ in \mathbb{R}^d . Using (3.4) as an asymptotic bound, we can estimate the local functional after refinement:

$$\sum_{j=1}^{2^d} \epsilon_{\tau_{i,j}}^2 \approx D \left(\frac{h_{\tau_i}}{2}\right)^{2p} M_{p+1, \tau_i} \frac{H_{\tau_i}}{2^d} 2^d \approx \frac{1}{2^{2p}} \epsilon_{\tau_i}^2. \quad (3.5)$$

Next, assume that the error is equally distributed among $\tau_{i,j}$, which yields

$$\epsilon_{\tau_{i,j}}^2 \approx \left(\frac{1}{2^{2p+d}} \right) \epsilon_{\tau_i}^2. \quad (3.6)$$

To give a little insight as to what this means, suppose quadratic elements are used in \mathbb{R}^2 . Then the functional in each child element should be about $\frac{1}{64}$ of its parent. If we allow an element to be refined twice, its grand-children are expected to have a local functional of about $\frac{1}{4096}$ of its grandparent. This suggests local errors can be reduced quickly and equally distributed if multiple refinements are correctly implemented.

3.2. Work estimate for NI-AMG. Here, we develop a procedure to estimate the computational work depending on the refinement decision made at each level. Let ℓ denote the refinement level, with $\ell = 0$ the coarsest and $\ell = L$ the finest grid. We make the following level-dependent definitions:

- N_ℓ = number of elements at level ℓ ;
- $\mathcal{G}_\ell(\mathbf{u}^h; \mathbf{f})$ = FOSLS functional at level ℓ ;
- $\epsilon_i^2 = \mathcal{G}_{\ell, \tau_i}(\mathbf{u}^h; \mathbf{f})$ = local functional on each element, τ_i , at level ℓ ;
- $\mathcal{M}_\ell(\mathbf{u}^h; \mathbf{f}) = \sqrt{\mathcal{G}_\ell(\mathbf{u}^h; \mathbf{f})}$ = error at level ℓ .

The algorithm works by initially ordering the elements at level ℓ so that

$$\epsilon_1^2 \geq \epsilon_2^2 \geq \dots \geq \epsilon_{N_\ell}^2. \quad (3.7)$$

Let $r \in [0, 1]$ be the fraction of elements to be refined versus the total number of elements. Define $E_\ell(r)$, the associated fraction of functional to be refined; that is,

$$E_\ell(r) = \frac{\sum_{i \leq r N_\ell} \epsilon_i^2}{\sum_{i=1}^{N_\ell} \epsilon_i^2}. \quad (3.8)$$

The functional distribution function, $E_\ell(r)$, is monotonically increasing and concave down from $E_\ell(0) = 0$ to $E_\ell(1) = 1$, that is, $E'_\ell(r) \geq 0$ and $E''_\ell(r) \leq 0$. The derivative $E'_\ell(0)$ can be used to indicate whether the functional is equally distributed. If it is large, then the functional is dominant in the first few elements. For example, in Fig. (3.2), when the error is dominated in the first two elements, the derivative at $r = 0$ is much greater than 1.

The algorithm allows an element to be refined more than once at each level. Let $r_i \in [0, 1]$ be the fraction of elements to be refined i times at level ℓ . Let m be the maximum refinements allowed on a single level. Writing $\mathbf{r} = (r_1, r_2, \dots, r_m)$ and combining the functional distribution (3.8) and the functional reduction heuristic (3.5), we estimate the functional reduction as a function of \mathbf{r} :

$$\gamma_\ell(\mathbf{r}) = 1 - E_\ell(r_1) + \sum_{k=1}^{m-1} \frac{1}{2^{2kp}} [E_\ell(r_{k+1}) - E_\ell(r_k)] + \frac{1}{2^{2mp}} E_\ell(r_m). \quad (3.9)$$

For the work required to achieve this reduction, the main concern is the increase in the DOF. Assuming simple bisection of the elements, the anticipated increase in DOF is easily computed. We have

$$N_{\ell+1} \simeq \eta_\ell(\mathbf{r}) N_\ell, \quad (3.10)$$

where

$$\eta_\ell(\mathbf{r}) = 1 - r_1 + \sum_{k=1}^{m-1} 2^{kn} (r_{k+1} - r_k) + 2^{md} r_m. \quad (3.11)$$

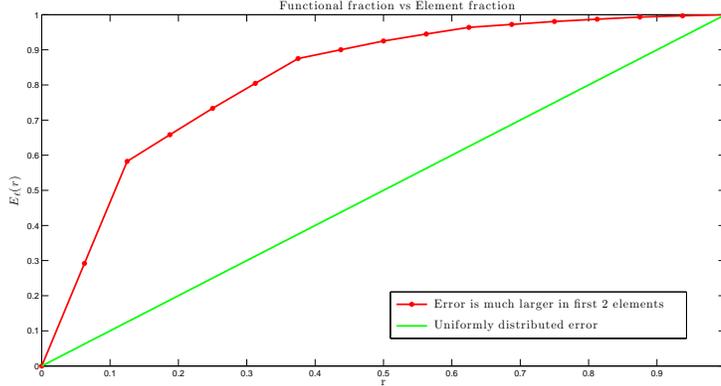


FIG. 3.1. Fraction of functional versus the fraction of elements to be refined.

Assume that the AMG convergence factor is bounded by $0 < \rho < 1$ uniformly in mesh-size. This value can be determined dynamically during computation. We further assume that, at each level, AMG V-cycles are applied until the discretization error, $\mathcal{M}_\ell(\mathbf{u}^h; \mathbf{f})$, is resolved. Then, the anticipated number of V-cycles, $\kappa_{\ell+1}(\mathbf{r})$, is determined by

$$\rho^{\kappa_{\ell+1}} \geq \frac{\mathcal{M}_{\ell+1}}{\mathcal{M}_\ell} = \sqrt{\gamma_\ell(\mathbf{r})}. \quad (3.12)$$

Solving for $\kappa_{\ell+1}$ gives

$$\kappa_{\ell+1}(\mathbf{r}) = \left\lceil \frac{1 \log \gamma_\ell(\mathbf{r})}{2 \log \rho} \right\rceil. \quad (3.13)$$

In a real simulation, a certain number of V-cycles are needed to extrapolate the discretization error. Denoting this number by $ncyc_{\min}$, we have

$$\kappa_{\ell+1}(\mathbf{r}) = \max \left(\left\lceil \frac{1 \log \gamma_\ell(\mathbf{r})}{2 \log \rho} \right\rceil, ncyc_{\min} \right). \quad (3.14)$$

Now, the work on the next level $\ell + 1$ is given by

$$\mathcal{W}_{\ell+1}(\mathbf{r}) = [C_0 + c_1 \kappa_{\ell+1}(\mathbf{r})] \times N_{\ell+1} = [C_0 + c_1 \kappa_{\ell+1}(\mathbf{r})] \times \eta_\ell(\mathbf{r}) \times N_\ell, \quad (3.15)$$

where C_0 and c_1 represent the respective set-up cost and cost factor for a V-cycle.

3.3. Efficiency-based Algorithms. Let $\mathcal{E}_T = \frac{\mathcal{G}_T}{\mathcal{G}_0}$ be the desired total factor of reduction from the initial functional. We wish to find an overall refinement strategy that minimizes the total work required to achieve a reduction of the functional by the factor \mathcal{E}_T . That is, we want to find a sequence, $\{\mathbf{r}_\ell\}_{\dots}$, to minimize the total work:

$$\mathcal{W}_T = \sum_{\ell=1}^L \mathcal{W}_\ell \quad \text{with} \quad \prod_{\ell=0}^{L-1} \gamma_\ell = \mathcal{E}_T. \quad (3.16)$$

Here, \mathcal{W}_ℓ is the work at level ℓ and γ_ℓ is the functional reduction between level ℓ and $\ell + 1$. Several difficulties arise in the solution of such a minimization problem.

The choice of \mathbf{r}_ℓ depends on the functional distribution, E_ℓ , which is unavailable before refinement is performed at level ℓ . We can estimate the distributions at finer levels based on heuristics (3.5) and the functional distribution at coarser levels. Such estimates are often not accurate enough, especially when the grid is not fine enough to resolve the solution.

Our first approach to (3.16) is based on local optimization at each level ℓ . Define the effective functional reduction measure as follows:

$$\gamma_\ell(\mathbf{r})^{\frac{1}{\mathcal{W}_{\ell+1}(\mathbf{r})}}. \quad (3.17)$$

The ACE algorithm, first developed in [15, 24], marks elements for refinement based on minimizing the anticipated effective functional reduction.

ALGORITHM 2 (ACE). *At level ℓ , order the elements so that*

$$\epsilon_1^2 \geq \epsilon_2^2 \geq \dots \geq \epsilon_{N_\ell}^2.$$

Allow m -multiple refinements, e.g., $m = 1, 2, 3$. Let $\mathbf{r} = (r_1, r_2, \dots, r_m)$ with $0 \leq r_m \leq \dots \leq r_1 \leq 1$. Find

$$\mathbf{r}_{opt} = \arg \min_{\mathbf{r}} \gamma_\ell(\mathbf{r})^{\frac{1}{\mathcal{W}_{\ell+1}(\mathbf{r})}}, \quad (3.18)$$

or

$$\mathbf{r}_{opt} = \arg \min_{\mathbf{r}} \frac{\log \gamma_\ell(\mathbf{r})}{\mathcal{W}_{\ell+1}(\mathbf{r})}. \quad (3.19)$$

Then, refine the first $\lceil r_i N_\ell \rceil$ elements i times, $i = 1, 2, \dots, m$.

In some instances, however, the ACE algorithm only refines a few elements. This may be optimal for the move from grid ℓ to grid level $\ell + 1$, but, if this happens at all levels, then the total work (3.16) will be unnecessarily large. Although C_0 and ρ are factored into the algorithm, the above behavior may occur if ρ is very close to 1.0 and C_0 is not large relative to the cost of one V-cycle. Furthermore, if elements are allowed to be refined more than twice, finding \mathbf{r}_{opt} can be expensive at finer levels. One modification that reduces this expense is the use of bins, which we discuss in Section 6.

Below, we propose three variations of the ACE algorithm. The first two, ACE-DOF and ACE-Reduc, enforce fixed increase in the DOF and a fixed reduction of the functional, respectively. The third algorithm, NACE, attempts to optimize (3.16). Numerical results in Section 4 indicate that all the ACE algorithms used with NI-FOSLS-AMG are able to approximate the solutions to the same accuracy with much less computational cost than the threshold-based refinement and uniform refinement.

As indicated above, the algorithm ACE-DOF has the goal of forcing ACE to refine a certain number of elements such that the number of elements at the next level is a prescribed factor of the number that would result from performing a single refinement globally.

ALGORITHM 3 (ACE-DOF). *On level ℓ , order the elements so that the local functional is decreasing. Assume $m \geq 2$. Given a parameter $1 < \theta_{DOF} \leq (2^d)^m$. Find*

$$\mathbf{r}_{opt} = \arg \min_{\mathbf{r}} \frac{\log \gamma_\ell(\mathbf{r})}{\mathcal{W}_{\ell+1}(\mathbf{r})} \quad \text{with} \quad \eta_\ell(\mathbf{r}) \simeq \theta_{DOF}. \quad (3.20)$$

In particular, one can choose $\theta_{DOF} = 2^d$ such that the number of elements at next level is the same as the number that would result from a single global refinement.

The second variation finds the optimal fraction, \mathbf{r}_{opt} , by fixing the anticipated functional reduction such that it is a prescribed factor of the anticipated functional reduction that would result from a single global refinement.

ALGORITHM 4 (ACE-Reduc). *At level ℓ , order the elements so the local functional is decreasing. Assume $m \geq 2$. Given a parameter $(\frac{1}{2^{2^p}})^m \leq \theta_{Reduc} < 1$. Find*

$$\mathbf{r}_{opt} = \arg \min_{\mathbf{r}} \frac{\log \gamma_{\ell}(\mathbf{r})}{\mathcal{W}_{\ell+1}(\mathbf{r})} \quad \text{with } \gamma_{\ell}(\mathbf{r}) \simeq \theta_{Reduc}, \quad (3.21)$$

or

$$\mathbf{r}_{opt} = \arg \min_{\mathbf{r}} \mathcal{W}_{\ell+1}(\mathbf{r}) \quad \text{with } \gamma_{\ell}(\mathbf{r}) \simeq \theta_{Reduc}. \quad (3.22)$$

All of the above algorithms are developed based on local optimization between two consecutive levels. Of course, this does not guarantee global optimization. We also devise a marking algorithm that minimizes the ‘anticipated-**overall**-computational-cost’ efficiency, as defined in (3.16), (which we call NACE). Let

$$\epsilon_{T,\ell} = \frac{\mathcal{G}_L}{\mathcal{G}_{\ell}} \quad (3.23)$$

be the overall functional reduction needed from the current functional to the desired tolerance. Let

$$K_{\ell}(\mathbf{r}) = \lceil \frac{\log(\epsilon_{T,\ell})}{\log(\gamma_{\ell}(\mathbf{r}))} \rceil. \quad (3.24)$$

In order to obtain \mathcal{G}_L , we repeat $\gamma_{\ell}(\mathbf{r})$ reduction $K_{\ell}(\mathbf{r})$ times. The anticipated total work to accomplish this is

$$\begin{aligned} \mathcal{W}_{T,\ell}(\mathbf{r}) &= [C_0 + c_1 \kappa_{\ell+1}(\mathbf{r})] \left(\eta_{\ell} + \eta_{\ell}^2 + \dots + \eta_{\ell}^{K_{\ell}(\mathbf{r})-1} \right) N_{\ell} \\ &= [C_0 + c_1 \kappa_{\ell+1}(\mathbf{r})] \frac{(\eta_{\ell}(\mathbf{r})^{K_{\ell}(\mathbf{r})} - 1)}{\eta_{\ell}(\mathbf{r}) - 1} N_{\ell}, \end{aligned} \quad (3.25)$$

where $\kappa_{\ell+1}(\mathbf{r})$, the anticipated number of V-cycles associated with reduction $\gamma_{\ell}(\mathbf{r})$, is defined in (3.14). Now, the NACE algorithm is described.

ALGORITHM 5 (NACE). *At level ℓ , order the elements so that the local functional is decreasing. The refinement decision is made by finding \mathbf{r}_{opt} to minimize the estimated remaining total work, (3.25). This is equivalent to finding*

$$\mathbf{r}_{opt} = \arg \min_{\mathbf{r}} \log \left((C_0 + c_1 \kappa_{\ell+1}(\mathbf{r})) \frac{\eta_{\ell}(\mathbf{r})^{K_{\ell}(\mathbf{r})} - 1}{\eta_{\ell}(\mathbf{r}) - 1} \right), \quad (3.26)$$

where $\kappa_{\ell+1}$, η_{ℓ} , and K_{ℓ} are given by (3.14), (3.11), and (3.24), respectively.

While this algorithm cannot guarantee optimal work as defined in (3.16), it attempts to take into consideration the total work required on all remaining levels.

4. Numerical Results. In this section, we explore the use of the proposed efficiency-based ALR algorithms in two spatial dimensions. The algorithms are first applied to a Poisson problem with steep gradients. With comparisons to uniform refinement and threshold-based (bulk-chasing) refinement, which we have called Algorithm 1, we show that the efficiency-based algorithms are capable of capturing the error generated at the steep gradients with much less work than the threshold-based refinement methods.

Next, we investigate a time-dependent nonlinear MHD test problem to show that the efficiency-based ALR methods work well with the NI-Newton-FOSLS-AMG method. Qualitatively, the test results demonstrate that within the NI-Newton-FOSLS-AMG framework, all efficiency-based methods resolve the physical features with much less work than global refinement. Moreover, for all test problems, the efficiency-based methods yield a sequence of meshes that equally distribute the functional across elements on relatively coarse levels. This is further discussed in Section 6. All tests are implemented using the First-Order System PACKage (FOSPACK [27]).

4.1. Poisson Equation. Consider the Poisson problem on the unit square, $\Omega = (0, 1) \times (0, 1)$,

$$\begin{cases} -\Delta p = f(x, y) & \text{in } \Omega, \\ p = g & \text{on } \partial\Omega, \end{cases} \quad (4.1)$$

with Dirichlet boundary condition. The equivalent first-order system we study here is

$$\begin{cases} -\nabla \cdot U = f & \text{in } \Omega, \\ U = \nabla p \\ \nabla \times U = 0 \\ p = g & \text{on } \partial\Omega, \\ \tau \cdot U = \frac{\partial g}{\partial \tau}, \end{cases} \quad (4.2)$$

where U is a vector of auxiliary unknowns and τ is the unit vector tangent to $\partial\Omega$. H^1 -ellipticity of the corresponding FOSLS functional is shown in [12].

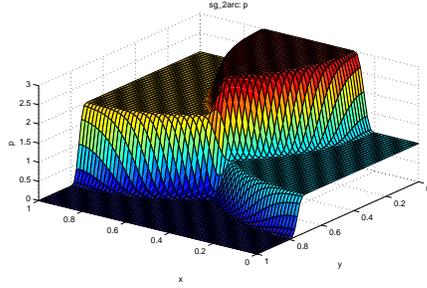
4.1.1. Test Problem: Steep Gradients and Flats. Define the function

$$p_1(r, \theta) = \begin{cases} 1 & r \leq 0.7, \\ h_1(r) & 0.7 \leq r \leq 0.8, \\ 0 & r \geq 0.8, \end{cases} \quad (4.3)$$

where (r, θ) is the polar coordinate centered at the origin and h_1 is a unique degree 7 polynomial such that $p_1 \in H^4(\Omega)$. Similarly, define the function

$$p_2(r', \theta') = \begin{cases} 2 & r' \leq 0.7, \\ h_2(r') & 0.7 \leq r' \leq 0.8, \\ 0 & r' \geq 0.8, \end{cases} \quad (4.4)$$

where (r', θ') is the polar coordinate centered at $(1, 0)$.

FIG. 4.1. *Exact solution*

The right-hand side, f , and boundary data, g , are chosen such that the exact solution is given by

$$p(x, y) = p_1(x, y) + p_2(x, y). \quad (4.5)$$

The 3D plot of p displayed in Fig. 4.1 shows a large gradient within two thin strips with constants elsewhere. For a given mesh size and approximation order, the error should be relatively large in the thin strips. To get an accurate approximation, the refinement algorithm needs to concentrate elements here to effectively resolve these gradients.

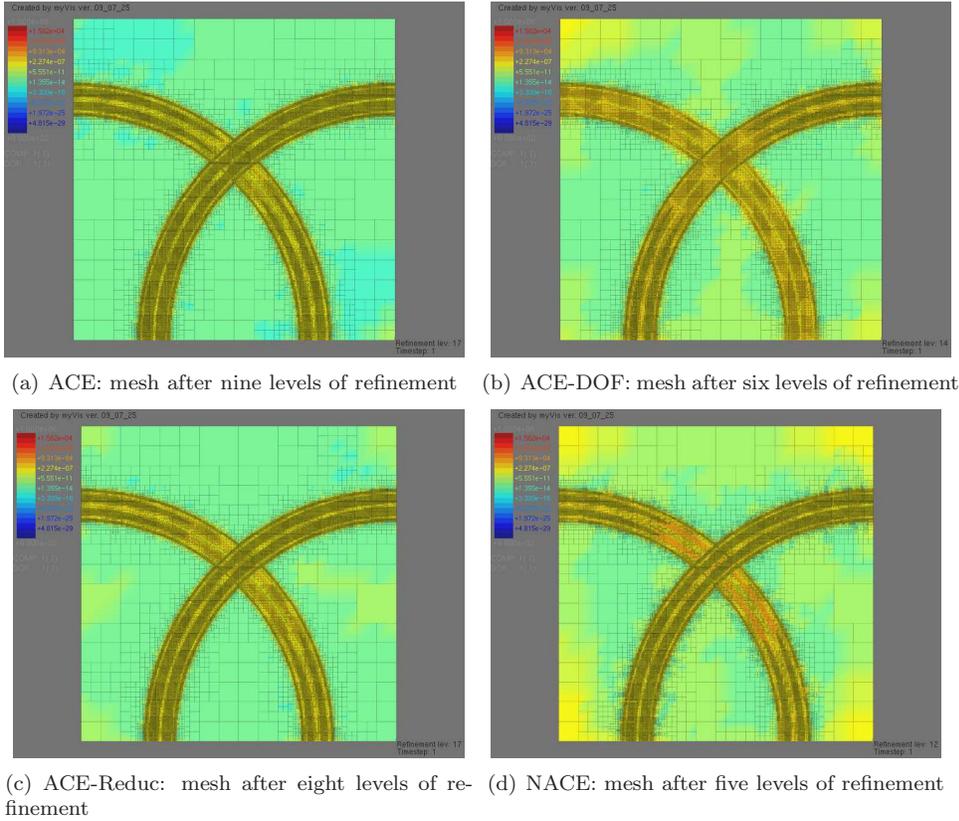
4.1.2. Test results. All ACE algorithms are applied to test problem (4.1) with bi-quadratic elements. Refinement stops when the functional is reduced by a factor of 10^{-7} . Elements are allowed to be refined at most twice at each level. The finest grids and functional distribution are depicted in Fig. 4.2. They are consistent with the anticipated mesh because the finest resolution encompasses the strips containing the large gradient. Furthermore, we see that all schemes result in equal distribution of error on the finest grids. Only a small fraction of elements contain large local error, marked as red.

To investigate the behavior of each scheme, we tabulate various relevant values with respect to each refinement level. These results are given in Tables 4.1, 4.2, 4.3, and 4.4. All schemes work as expected. A large fraction of elements are refined at the initial levels when grids are too coarse to resolve features of the solution. Then, at the intermediate levels, once local features of the solution are exposed, a small fraction of elements that contain large local error are refined. Later, at finer levels, since error is fairly equally distributed, a large fraction of elements are refined once again. Furthermore, the last column in each table shows that the anticipated functional reduction, γ_ℓ , provides an accurate estimate to the actual reduction at finer levels. This verifies that FOSLS approximation heuristics derived in section 3.1.

Next, to demonstrate the efficiency of each scheme, we compute the total computational cost in terms of a work unit (WU) on the finest grid, defined as the amount of computation required to perform one matrix vector multiplication on the finest grid. The total computational cost is, then, given in terms of the total work units (TWU):

$$TWU = \frac{\sum_{\ell=1}^L (C_0 + ncy_\ell) \times (\nu_1 + \nu_2) \times \sigma_\ell \times nnz_\ell}{nnz_L}. \quad (4.6)$$

Here, ν_1 and ν_2 are the number of pre- and post-relaxations, respectively, σ_ℓ is the operator complexity of the AMG solver at level ℓ , nnz_ℓ is the number of nonzeros

FIG. 4.2. *Locally-refined meshes and functional distribution*

at level ℓ , and C_0 is defined as the set up cost in terms of the cost of a single V-cycle on level ℓ . For this test problem, $V(1,1)$ -cycles are employed and the set up cost is proportional to 30.0 V-cycles. Results show that the total work to solve the linear systems throughout all levels is about 22 work unit, and the total set up cost is between 158 and 173 work units. To illustrate what these numbers mean, we take the finest grid resulting from the original ACE, set up the FOSLS discrete problem, and solve it using AMG with a zero initial guess. The results in Table.4.5 show that the NI-FOSLS-AMG-ACE method requires only about 137% of the work of solving the problem directly on the same finest grid.

Next, we see from Table.4.4 that the NACE scheme takes the least levels of refinement to reach the error tolerance due to a lot more double refinements. This may lead to possible over-refinement and less accurate grids, which is indeed the case; see Fig.4.3(a), where the functional-versus-number-of-elements curve and work units are depicted. The convergence rates of ACE, ACE-DOF, and ACE-Reduc approach the optimal rate of quadratic elements, while the convergence rate of the NACE scheme is slightly slower. Double refinements also have the potential of introducing more nonzeros in the resulting matrices, e.g., the NACE scheme results in more nonzeros in the finest-grid matrix than ACE and ACE-Reduc, but less elements. To compare the computational work required to reach a certain functional value, we compute the work units in terms of relaxation on the finest grid of ACE, which

ℓ	\mathcal{G}_ℓ	N_ℓ	nnz_ℓ	r_1	r_2	$E(r_1)$	$E(r_2)$	η_ℓ	$ncyc$	$\gamma_{est}(\gamma_{act})$
1	1.37e+5	16	9,801	63%	0.00%	96%	00%	2.87	4	0.10(0.49)
2	6.70e+4	46	29,709	52%	0.00%	98%	00%	2.56	4	0.08(0.42)
3	2.85e+4	130	84,277	52%	0.00%	99%	00%	2.57	4	0.07(0.19)
4	5.49e+3	364	234,237	56%	0.55%	98%	16%	2.74	4	0.07(0.11)
5	5.92e+2	1,114	664,061	53%	2.33%	98%	46%	2.88	4	0.05(0.09)
6	5.48e+1	3,505	2,093,261	34%	0.63%	92%	28%	2.09	4	0.12(0.13)
7	7.33e+0	7,756	4,595,173	34%	0.04%	91%	02%	2.02	4	0.15(0.15)
8	1.14e+0	16,213	9,531,203	69%	0.12%	98%	06%	3.09	4	0.08(0.08)
9	1.01e-1	51,157	29,884,277	82%	0.27%	99%	09%	3.50	4	0.07(0.07)
10	7.62e-3	181,633	105,595,645							

TABLE 4.1

ACE, $\sigma_L = 1.899$, set up 171.804 WU, solve 22.907 WU.

ℓ	\mathcal{G}_ℓ	N_ℓ	nnz_ℓ	r_1	r_2	$E(r_1)$	$E(r_2)$	η_ℓ	$ncyc$	$\gamma_{est}(\gamma_{act})$
1	1.37e+5	16	9,801	81%	6.25%	98%	29%	4.19	4	0.06(0.42)
2	5.77e+4	67	41,873	60%	10.45%	99%	59%	4.04	4	0.03(0.25)
3	1.44e+4	298	184,313	59%	10.40%	99%	82%	4.01	4	0.02(0.05)
4	6.99e+2	1,333	809,373	64%	9.00%	99%	76%	4.00	4	0.02(0.02)
5	1.60e+1	5,920	3,573,901	70%	7.62%	99%	49%	4.00	4	0.04(0.05)
6	7.84e-1	25,153	15,128,019	80%	5.01%	99%	47%	4.00	4	0.04(0.04)
7	3.12e-2	103,444	61,036,269							

TABLE 4.2

ACE-DOF, $\sigma_L = 1.983$, set up 160.695 WU, solve 21.426 WU.

ℓ	\mathcal{G}_ℓ	N_ℓ	nnz_ℓ	r_1	r_2	$E(r_1)$	$E(r_2)$	η_ℓ	$ncyc$	$\gamma_{est}(\gamma_{act})$
1	1.37e+5	16	9,801	100%	0.00%	100%	00%	4.00	4	0.065(0.48)
2	6.60e+4	64	38,025	47%	1.56%	99%	13%	2.59	4	0.065(0.39)
3	2.59e+4	169	102,677	46%	1.18%	99%	14%	2.52	4	0.065(0.13)
4	3.24e+3	472	284,901	57%	1.27%	99%	11%	2.86	4	0.065(0.07)
5	2.39e+2	1,492	888,377	58%	1.14%	99%	17%	2.86	4	0.065(0.09)
6	2.16e+1	4,627	2,741,061	60%	0.63%	98%	22%	2.87	4	0.065(0.09)
7	1.95e+0	13,849	8,169,677	70%	0.86%	98%	27%	3.20	4	0.065(0.08)
8	1.52e-1	45,448	26,632,607	81%	0.51%	99%	17%	3.49	4	0.065(0.07)
9	1.06e-2	160,897	93,704,707							

TABLE 4.3

ACE-Reduc, $\sigma_L = 1.994$, set up 173.298 WU, solve 23.106 WU.

ℓ	\mathcal{G}_ℓ	N_ℓ	nnz_ℓ	r_1	r_2	$E(r_1)$	$E(r_2)$	η_ℓ	$ncyc$	$\gamma_{est}(\gamma_{act})$
1	1.37e+5	16	9,801	100%	50%	100%	93%	10.00	4	0.010(0.228)
2	3.11e+4	160	104,967	46%	23%	99%	86%	5.16	4	0.014(0.095)
3	2.96e+3	958	653,599	72%	29%	99%	97%	6.63	4	0.006(0.008)
4	2.45e+1	6,868	5,511,917	36%	01%	97%	50%	2.18	4	0.061(0.065)
5	1.59e+0	15,277	12,768,435	56%	55%	95%	95%	9.18	4	0.057(0.029)
6	4.58e-2	153,064	108,988,185							

TABLE 4.4

NACE, $\sigma_L = 2.220$, set up = 158.112 WU, solve 21.082 WU.

Method	Setup Cost	n_{cyc}	Solve Cost	Total Work
NI-FOSLS-AMG-ACE	171.80	4	22.91	194.71
FOSLS-AMG	113.94	10	37.98	141.92

TABLE 4.5

Comparison of NI-FOSLS-AMG-ACE and applying FOSLS-AMG directly to the finest-grid, n_{cyc} is the number of V-cycles used on the finest grid.

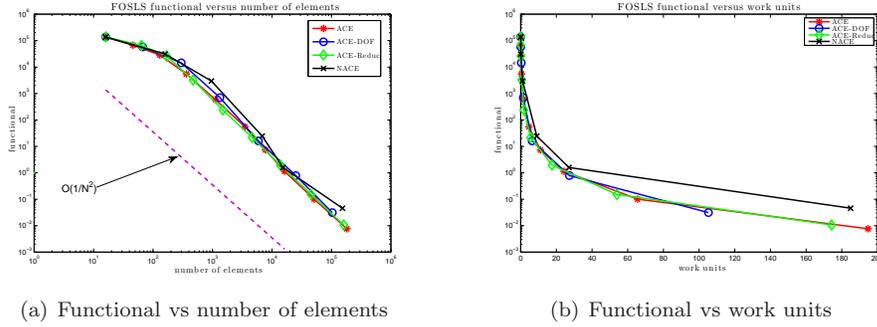


FIG. 4.3. Comparison of all ACE schemes, where a work unit is defined as the cost of one matrix vector multiplication on the finest grid of ACE.

contains 105,595,645 nonzeros. The ACE scheme (and its two variations) results in smaller functional values compared with the NACE scheme. It appears that, for this test, when the work units equal 180, the functional resulting from NACE is almost an order of magnitude larger than the functional using the ACE scheme, as seen in Fig. 4.3(b).

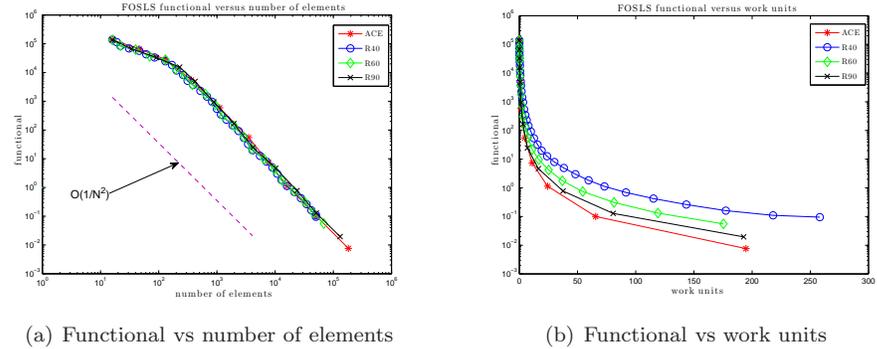


FIG. 4.4. Comparison of threshold-based schemes with refinement of 40, 60, and 90 percent of the functional at each level, and the ACE scheme.

We conclude our analysis of the Poisson equation by comparing the original ACE algorithm with the threshold-based marking scheme (1.3). Three threshold-based algorithms that refine 40, 60, and 90 percent of the functional at each refinement level are considered. Work units for Fig. 4.4(b) are defined as one relaxation on the finest grid of ACE. It is shown in Fig. 4.4(a) that the convergence rate of ACE is the same as the best convergence rate of the three threshold-based algorithm. For the same amount of work, the ACE scheme results in the smallest functional value compared

with the threshold-based schemes. For example, when work units = 200, ACE results in functional one order of magnitude less than threshold-based refinement schemes. This is expected since the ACE algorithm is based on optimizing computational efficiency.

4.2. Magnetohydrodynamics. In this section, an incompressible, resistive magnetohydrodynamics (MHD) test problem is investigated. The results in [1, 2] shows that methods such as nested iteration and first-order system least-squares are capable of solving the nonlinear MHD systems in a minimal amount of work units. Here, the various forms of adaptive mesh refinement described above are applied to a tokamak test problem [13, 14, 25, 28]. A reduced set of MHD equations is obtained that models a “large-aspect-ratio” tokamak, with non-circular cross-sections. The magnetic B-field along the z-direction, or the toroidal direction, is very large and mostly constant. In this context, we are able to look at plasma behavior in the poloidal cross-section. The 2D reduced model is described by the following equations:

$$\frac{1}{\sqrt{R_e}} \nabla \times \mathbf{u} - \sqrt{R_e} \omega = 0, \quad (4.7)$$

$$\frac{1}{\sqrt{R_e}} \nabla \cdot \mathbf{u} = 0, \quad (4.8)$$

$$\frac{1}{\sqrt{R_e}} \frac{\partial \mathbf{u}}{\partial t} - \mathbf{u} \times \boldsymbol{\omega} - \mathbf{j} \times \mathbf{B} - \sqrt{R_e} \nabla p + \frac{1}{\sqrt{R_e}} \nabla^\perp \omega = \mathbf{f}, \quad (4.9)$$

$$\frac{1}{\sqrt{S_L}} \nabla \times \mathbf{B} - \sqrt{S_L} j = 0, \quad (4.10)$$

$$\frac{1}{\sqrt{S_L}} \nabla \cdot \mathbf{B} = 0, \quad (4.11)$$

$$\frac{1}{\sqrt{S_L}} \frac{\partial \mathbf{B}}{\partial t} + \frac{1}{\sqrt{R_e S_L}} (\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}) + \frac{1}{\sqrt{S_L}} \nabla^\perp j = \mathbf{g}. \quad (4.12)$$

The x-direction denotes the periodic poloidal direction in the tokamak, whereas the y dimension represents a thin annulus in the poloidal cross section. In this 2D setting, vorticity, ω , and current density, j , are both scalar variables. The remaining variables are the fluid velocity, \mathbf{u} , the fluid pressure, p , and the magnetic field, \mathbf{B} . The equations have been scaled using the Reynolds number, R_e , which is the ratio of fluid speed to viscosity, and the Lundquist number, S_L , which is the ratio of fluid speed to magnetic resistivity. This scaling produces a first-order system that is amenable to algebraic multigrid methods in the FOSLS context, as shown in [1].

One important application of MHD physics is the study of instabilities that can occur in tokamak fusion reactors. One such instability, the island coalescence problem, is described below. The various ACE schemes are applied to see which one most efficiently captures the magnetic reconnection that results from this instability.

4.2.1. Test Problem: Island Coalescence. This test problem simulates an island coalescence in the current density arising from perturbations in an initial current density sheet. A current density sheet in the toroidal direction of the tokamak is perturbed, resulting in an instability that causes a reconnection in the magnetic field lines and the merging of two islands in the current density field. This produces a sharp peak in current density where the magnetic field lines reconnect. This region is known as the reconnection zone, and the point at which the magnetic field lines break is known as the \mathcal{X} point. See [4, 20, 25] for more detail. We choose a low enough

resistivity (i.e., Lundquist number above 50,000) in order to observe the interesting physics. For the following simulations, we define

$$\begin{aligned}\Omega &= [-1, 1] \times [-1, 1], \\ R_e = S_L &= 50,001.\end{aligned}$$

The initial conditions at equilibrium are

$$\mathbf{B}_0(x, y) = \frac{1}{\cosh(2\pi y) + k \cos(2\pi x)} \begin{pmatrix} \sinh(2\pi y) \\ k \sin(2\pi x) \end{pmatrix}, \quad (4.13)$$

$$\mathbf{u}_0(x, y) = \mathbf{0}, \quad (4.14)$$

$$\omega_0(x, y) = 0, \quad (4.15)$$

$$j_{30}(x, y) = \nabla \times \mathbf{B}_0 = \frac{2\pi(k^2 - 1)}{(\cosh(2\pi y) + 0.2 \cos(2\pi x))^2}, \quad (4.16)$$

$$p_0(x, y) = \frac{(1 - k^2)}{2} \left(1 + \frac{1}{(\cosh(2\pi y) + 0.2 \cos(2\pi x))^2} \right), \quad (4.17)$$

where $k = 0.2$. These initial conditions are perturbed away from equilibrium as follows:

$$\delta \mathbf{B}_0(x, y) = \begin{pmatrix} -\epsilon \frac{1}{\pi} \cos(\pi x) \sin(\pi \frac{y}{2}) \\ \frac{1}{2} \epsilon \frac{1}{\pi} \cos(\pi \frac{y}{2}) \sin(\pi x) \\ 0 \end{pmatrix}, \quad (4.18)$$

$$\delta j_{30}(x, y) = \epsilon \cos(\pi \frac{y}{2}) \cos(\pi x), \quad (4.19)$$

where $\epsilon = -0.01$. The boundary conditions are periodic in x and Dirichlet for the current density and vorticity on the top and bottom of the domain. We also have $\mathbf{n} \cdot \mathbf{u}$ and $\mathbf{n} \cdot \mathbf{B}$ known on the top and bottom. Again, the FOSLS formulation, (4.7)-(4.12), is H^1 elliptic.

4.2.2. Results. The problem was run to time $8\tau_A$ with a timestep of $0.1\tau_A$, using a BDF-2 implicit time-stepping scheme. Here, τ_A is the time in Alfvén units. It is the time needed for an Alfvén wave to travel across the domain [4, 28]. By this time, the islands have coalesced and the large peak in current density has occurred at the reconnection point. Using both uniform refinement and the ACE schemes, the instability was captured. With ACE employed, the grids evolve over time to refine in areas with steeper gradients. In this problem, as time progresses, a steep gradient occurs at the reconnection point. This is seen in the bottom graph in Fig. 4.5. We expect, then, that most of the refinement occurs in this region, which is indeed the case. Next, a comparison of the 4 ACE schemes is done relative to uniform refinement. The work at one time step is calculated by first determining the work of all the V-cycles on a given refinement level for that particular scheme. These values, times the number of matrix nonzeros for the level, are then summed over all grids and divided by the number of nonzeros on the finest refinement level for the given problem. In Table 4.6, the work unit values given are with respect to the finest level of the given refinement scheme. They are an average over all time steps. To compare two schemes, the average work unit value is multiplied by the fine-grid nonzeros for that scheme

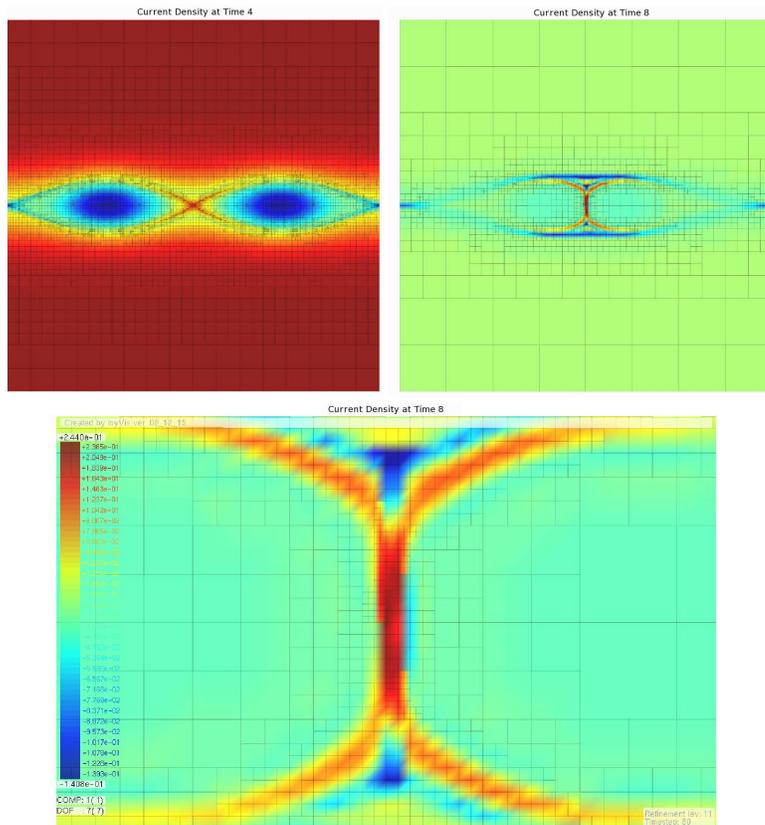


FIG. 4.5. Numerical solution using 10 levels of adaptive refinement. $S_L = R_e = 50,001$. Top Left: Current Density at Time $4\tau_A$. Top Right: Current Density at Time $8\tau_A$. Bottom: Zoomed in plot of current density peak at Time $8\tau_A$.

and then the ratio is taken. This ratio is defined as the Work Ratio in Table 4.6. Similarly, the Element Ratio column is the ratio of elements on the finest grid of the adaptive scheme compared to the number of elements on the finest grid of the uniform scheme.

The results show that using adaptive refinement greatly reduces the amount of work needed, compared to that of using uniform refinement. ACE requires 12% of the work that uniform refinement requires. The physics is more localized in this problem, especially by the time the reconnection begins to develop and, thus, the refinement is more localized. It appears that, for this problem, ACE gives the best efficiency. The functional is reduced to the same order of magnitude in all cases, but original ACE needs less elements. The ACE-DOF and ACE-Reduc schemes appear to add unnecessary elements just to get a certain total number or to reduce the functional more than is needed. The NACE scheme also appears to oversolve. In this case, the NACE scheme is performing many double refinements and, in fact, frequently jumps the functional tolerance prescribed on many time steps. However, it is still on par with the ACE-DOF and ACE-Reduc schemes. Qualitatively, all 4 ACE schemes appear to capture the coalescence of the two islands.

Uniform		Ratio to Uniform	
Work Units 80.473	Avg Elements 13,380		
ACE			
Work Units 9.789	Avg Elements 1,779.9	Work Ratio 0.12	Element Ratio 0.13
ACE-DOF			
Work Units 29.610	Avg Elements 3,040.7	Work Ratio 0.37	Element Ratio 0.23
ACE-Reduc			
Work Units 23.513	Avg Elements 3,083.0	Work Ratio 0.29	Element Ratio 0.23
NACE			
Work Units 22.907	Avg Elements 2,895.2	Work Ratio 0.28	Element Ratio 0.22

TABLE 4.6

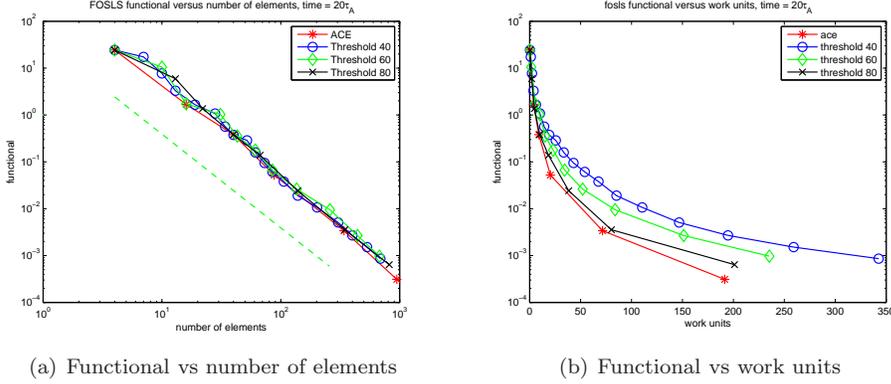
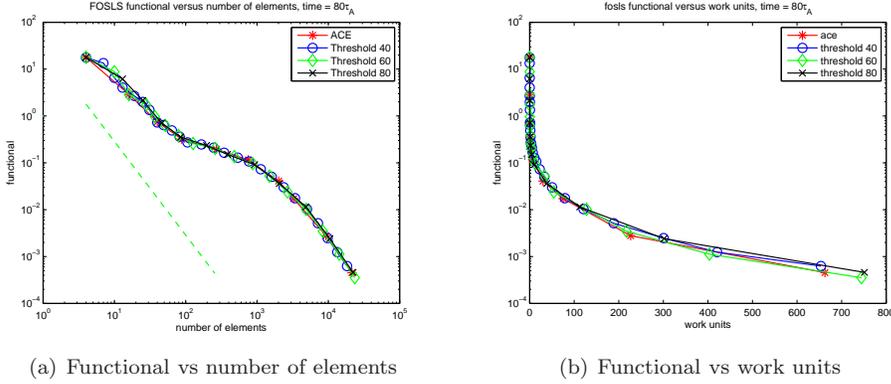
Average number of work units per timestep using uniform refinement versus various ACE refinement. All values are relative to finest grid of uniform refinement. A total of 45 time steps were performed to compute the averages.

As a comparison to the threshold-based schemes described, the island problem was also run using these schemes with values of 40, 60, and 80%, for the number of elements to refine. Comparisons were made at various time steps throughout the run. While all different schemes captured the qualitative behavior of the island coalescence problem, the threshold schemes often required more elements and more work units to resolve the problem to the same functional values. Figures 4.6 and 4.7 give a comparison of the schemes for time step 20 ($t = 2\tau_A$) and time step 80 ($t = 8\tau_A$), respectively. These figures show the relationship between number of elements on the finest grid versus functional and the relationship between the number of work units and functional. At time step 20, the solution is still rather smooth and ACE appears to get the optimal grid, requiring less elements and less work units to get to the same functional value as the threshold-based schemes. At time step 80, the reconnection has taken place and steep gradients have developed. At this time, all schemes appear to require more work and elements to resolve the physics. However, ACE is no worse than any of the threshold-based schemes. While ACE picked the optimal refinement pattern from the efficiency measures while running, the best threshold method required knowing the correct percentage ahead of time.

5. Convergence Analysis. When devising a refinement scheme it is important to consider the convergence. Recent theorems, proven in [21], address convergence for local adaptive refinement in a FOSLS framework. Some quick notational background is necessary to discuss FOSLS convergence. Note that $\mathcal{G}(\mathbf{u}_h; f) = \mathcal{G}(E; 0)$, where $E = \mathbf{u}^h - \mathbf{u}$ is the error. An \mathcal{F} -orthogonal decomposition of E into \mathcal{F} -harmonic and \mathcal{F} -local components is given by

$$E(x, y) = \mathcal{H} + \eta.$$

See [21] for details. To formalize \mathcal{H} and η , consider $E^- := E(x, y)$ restricted to the refinement region, Ω^- , and let \mathcal{H}^- and η^- be the associated \mathcal{F} -harmonic and local error components, respectively. We define a set of local functions having support in

FIG. 4.6. Comparison between ACE and threshold-based schemes at time step $t = 2\tau_A$.FIG. 4.7. Comparison between ACE and threshold-based schemes at time step $t = 8\tau_A$.

the refinement region. We denote the complement of Ω^- by Ω^+ . Define

$$\mathcal{V}^- := \{u \in \mathcal{V} : u = 0 \text{ on } \Omega^+ \cup \Gamma\}.$$

Then,

$$\eta^- := \arg \min_{u \in \mathcal{V}^-} \mathcal{G}(E - u; 0)$$

and

$$\mathcal{H}^- := \arg \min_{u=E \text{ on } \Omega^+ \cup \Gamma} \mathcal{G}_{\Omega^-}(u; 0).$$

This decomposition is relevant because error of type \mathcal{H}^- cannot be substantially reduced even by infinite refinement.

Assume that the error in the refinement region satisfies a local saturation criteria. This means that refinement reduces the \mathcal{F} -local error component by a substantial amount. The theory in [21] affirms that if enough of the total error resides in the refinement region, the functional is reduced by a substantial amount, on that refinement level. At present, we do not constrain our schemes to insure that we have enough

error in the refinement region to satisfy the theory however, in practice, the scheme typically choses a refinement region containing most of the error. This enables us to qualify a certain confidence that our scheme is guaranteed to have level by level convergence. In the future, we will explore the utility in determining this constraint precisely.

6. Parallel Considerations. To accommodate the continual need for greater computing power, it is imperative to implement NI-FOSLS-AMG and the efficiency-based ALR algorithms in parallel for two- and three- dimensional problems. In this section, the base parallel implementation is detailed. Clearly, a global sort of the local functional values is not efficient, especially in a massively parallel environment. To overcome this difficulty, a binning strategy is developed that first determines and broadcasts the maximum local functional value over the entire domain, denoted by ϵ_{\max}^2 . The algorithm establishes bins based geometrically on the maximum value and the factor by which children elements are reduced. That is, in the two-dimensional case mentioned above using bi-quadratic elements, the top bin consists of the range $[\epsilon_{\max}^2/64, \epsilon_{\max}^2]$ and the second bin consists of the range $[\epsilon_{\max}^2/64^2, \epsilon_{\max}^2/64]$. If an element of the first bin is refined, its children are expected to land in the second bin and so forth. Each processor establishes the number of elements and amount of functional in each bin in its domain. This information is then distributed to all processors by a global MPI_Allreduce so that every processor has the total number of elements and functional in each bin. Each bin is treated as an abstract element in the efficiency-based formulas. This amounts to a piecewise linear approximation of the functional distribution function, $E(r)$, defined in (3.8). Finding the optimal \mathbf{r} is greatly simplified because of the smaller number of bins. If a bin is chosen to be refined, then all elements of that bin are refined.

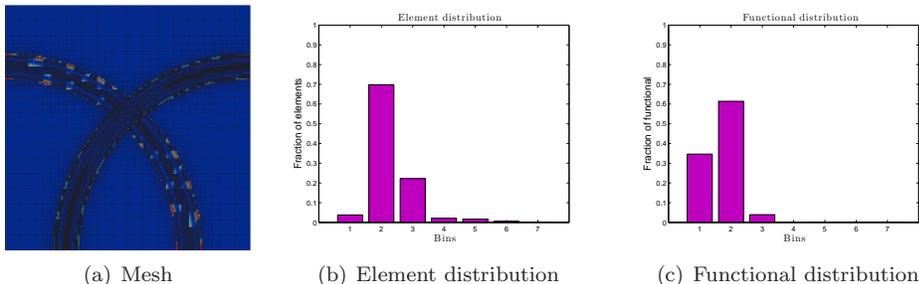


FIG. 6.1. Mesh after nine levels of refinement, parallel ACE using geometric binning with single refinement and bi-linear elements.

Preliminary results with this algorithm are very promising, as Fig. 6.1 shows. It may be beneficial to coarsen the bins less aggressively, producing a finer approximation of $E(\mathbf{r})$. This remains for future research. Also, load balancing issues are important for parallel adaptive methods (see, e.g., [3]). The load balancing algorithms based on quadtree (octree) and space-filling curve described in [30] are being implemented in the parallel FOSPACK code. Our philosophy is to get as much of the load balanced as possible on coarser grids. Since our ACE algorithms concentrate on making the error approximately equal across the elements, nearly global refinements are expected at finer levels. This automatically ameliorates load balancing issues. Another difficulty in parallel is to enforce the 2-to-1 balance refinement. To accomplish this, we choose to balance the local mesh within each processor, followed by balancing the interprocessor

boundaries using the ripple propagation algorithm [29, 30]. Currently, two types of parallel AMG solvers are available in FOSPack: HYPRE’s BoomerAMG [17, 19] and the parallel smoothed aggregation multigrid package parSAMIS, developed at the University of Colorado at Boulder.

7. Conclusions. In this paper, efficiency-based refinement algorithms for the FOSLS finite element method with algebraic multigrid solvers in the context of nested iteration (NI-FOSLS-AMG) are developed. The algorithms choose which elements to refine based on optimizing computational efficiency, taking into account both error reduction and computational cost. Two efficiency measures are considered: predicted ‘accuracy-per-computational-cost’ (ACE) and the new ‘anticipated-overall-computational-cost’ (NACE). The use of the FOSLS local functional as a sharp a posteriori error estimate along with NI-AMG methods allows parameters to be computed that are used to estimate the current measures. In addition, several “flavors” of these efficiency-based schemes are tested to determine whether adding certain constraints to the efficiency measure, such as the total number of elements to add or the total amount of error to be reduced, would make it easier to obtain a near optimal grid. Numerical tests show that all of the efficiency-based algorithms effectively and efficiently capture local features of the solution. For the linear test problem, all schemes perform equally well, suggesting that the standard ACE scheme is sufficient without any extra constraints. For the more complicated nonlinear time-dependent MHD problem, this also is the case. In fact, the constrained schemes appear to at times perform unnecessary work, making them less optimal. However, all schemes greatly reduce the amount of computational cost for solving these problems to a specified accuracy compared to the cost of uniform refinement. In addition, in comparing the ACE scheme to threshold-based schemes, ACE either outperformed the threshold-schemes or was no worse than the best threshold-based method at any given time step. As the optimal refinement strategy varies over time steps, choosing a scheme such as ACE, which can adaptively choose the optimal refinement strategy is preferable in the case in which many time steps are needed and the physics can change dramatically.

Several aspects still need to be studied. In this work, a generic AMG solver was used. Deterioration in the AMG convergence for increased timestep size as well as Reynolds and Lundquist numbers are observed in the MHD test. Even a slight improvement in the AMG algorithm would greatly reduce the total work units required to achieve a specified accuracy. AMG algorithms specifically designed for systems of PDEs are a topic of current research. This might involve the use of newly developed adaptive multigrid algorithms described more in [8, 9]. In addition, the hierarchy of the grids resulting from adaptive refinement might be used to reduce or eliminate the set up phase of AMG at each level. A new multigrid solver might be developed for problems arising from adaptive refinement procedures. This would involve including more of the geometry or structure of the grids into the multilevel solver. Since the problems that would use such schemes, such as MHD, which is used in a variety of applications, including fusion energy physics and space weather, are gaining increased interest, it is reasonable to tune the numerics for such specific problems.

Many aspects of the adaptive refinement algorithms can be improved. The FOSLS approximation heuristics introduced in Sect. 3.1 require certain smoothness assumptions. When the solution contains singularities, for instance, one might want to adaptively determine the strength of the singularity and appropriately apply graded refinement techniques rather than splitting elements into subelements with equal size in each direction. Finally, modifications to accommodate parallel computing need

to be studied, such as various binning strategies used to derive the parallel marking algorithms. Also, load balancing issues are important for parallel adaptive refinement methods. At each refinement level, space filling curves might be used to redistribute elements such that each processor contains a subdomain with approximately the same amount of elements or error. As the grid becomes finer, the optimal refinement approaches uniform refinement, which requires almost no load balancing. This will be explored in future research.

REFERENCES

- [1] J. H. ADLER, T. A. MANTEUFFEL, S. F. MCCORMICK, AND J. W. RUGE, *First-order system least squares for incompressible resistive Magnetohydrodynamics*, SIAM J. Sci. Comp. (SISC), 32 (2010), pp. 229–248.
- [2] J. H. ADLER, T. A. MANTEUFFEL, S. F. MCCORMICK, J. W. RUGE, AND G. D. SANDERS, *Nested Iteration and First-Order System Least Squares for Incompressible, Resistive Magnetohydrodynamics*, SIAM J. Sci. Comp. (SISC), (submitted 2009).
- [3] R. E. BANK AND M. J. HOLST, *A New Paradigm for Parallel Adaptive Meshing Algorithms*, SIAM Review, 45 (2003), pp. 292–323.
- [4] G. BATEMAN, *MHD Instabilities*, The MIT Press, 1978.
- [5] M. BERNDT, T. A. MANTEUFFEL, AND S. F. MCCORMICK, *Analysis of First-order System Least Squares (FOSLS) for Elliptic Problems with Discontinuous Coefficients: Part II*, SIAM J. Numer. Anal, 43 (2005), pp. 409–436.
- [6] M. BERNDT, T. A. MANTEUFFEL, AND S. F. MCCORMICK, *Local Error Estimates and Adaptive Refinement for First-Order system Least Squares (FOSLS)*, E.T.N.A., 6 (1997), pp. 35–43.
- [7] S. C. BRENNER AND L. R. SCOTT, *The Mathematical Theory of Finite Element Methods*, Springer, 2nd ed., 2002.
- [8] M. BREZINA, R. FALGOUT, S. MACLACHLAN, T. A. MANTEUFFEL, S. F. MCCORMICK, AND J. W. RUGE, *Adaptive Smoothed Aggregation (aSA) Multigrid*, SIAM Review (SIGEST), (2005), pp. 317–346.
- [9] ———, *Adaptive Algebraic Multigrid*, SIAM J. Sci. Comp. (SISC), (2006), pp. 1261–1286.
- [10] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, SIAM, Philadelphia, 2nd ed., 1999.
- [11] Z. CAI, R. LAZAROV, T. A. MANTEUFFEL, AND S. F. MCCORMICK, *First-Order System Least Squares for Second-Order Partial Differential Equations. I*, SIAM J. Numer. Anal., 31 (1994), pp. 1785–1799.
- [12] Z. CAI, T. A. MANTEUFFEL, AND S. F. MCCORMICK, *First-Order System Least Squares for Second-Order Partial Differential Equations. II*, SIAM J. Numer. Anal, 34 (1996), pp. 425–454.
- [13] L. CHACON, D. A. KNOLL, AND J. M. FINN, *An Implicit, Nonlinear Reduced Resistive MHD Solver*, J. of Computational Physics, 178 (2002), pp. 15–36.
- [14] ———, *Nonlinear Study of the Curvature-Driven Parallel Velocity Shear-Tearing Instability*, Physics of Plasmas, 9 (2002), pp. 1164–1176.
- [15] H. DE STERCK, T. A. MANTEUFFEL, S. F. MCCORMICK, J. W. NOLTING, J. W. RUGE, AND L. TANG, *Efficiency-Based h- and hp-refinement Strategies for Finite Element Methods*, Num. Lin. Alg. Appl., 15 (2008), pp. 89–114.
- [16] W. DÖRFLER, *A Convergent Adaptive Algorithm for Poisson’s Equation*, SIAM J. Numer. Anal., 33 (1996), pp. 1106–1124.
- [17] R.D. FALGOUT AND U.M. YANG, *hypr: a Library of High Performance Preconditioners*, in Computational Science - ICCS 2002 Part III, vol. 2331, Springer-Verlag, 2002, pp. 632–641.
- [18] W. GUI AND I. BABUŠKA, *The h, p, and hp Versions of the Finite Element Method in 1 Dimension, Parts I, II, III*, Numerische Mathematik, 49 (1986), pp. 577–683.
- [19] V.E. HENSON AND U.M. YANG, *BoomerAMG a Parallel Algebraic Multigrid Solver and Preconditioner*, Applied Numerical Mathematics, 41 (2002), pp. 155–177.
- [20] D. A. KNOLL AND L. CHACON, *Coalescence of Magnetic Islands, Sloshing, and the Pressure Problem*, Physics of Plasmas, 13 (2006).
- [21] T. A. MANTEUFFEL, S. F. MCCORMICK, J. W. NOLTING, J. W. RUGE, AND G. D. SANDERS, *Further Results on Error Estimators for Local Refinement with First-Order System Least Squares (FOSLS)*, Num. Lin. Alg. Appl., (submitted 2009).
- [22] P. MORIN, R. H. NOCHETTO, AND K. G. SIEBERT, *Data Oscillation and Convergence of Adap-*

- tive FEM*, SIAM J. Numer. Anal., 38 (2000), pp. 466–488.
- [23] ———, *Convergence of Adaptive Finite Element Methods*, SIAM Review, 44 (2002), pp. 631–658.
- [24] JOSH NOLTING, *Efficiency-based Local Adaptive Refinement for FOSLS Finite Elements*, PhD thesis, University of Colorado, Applied Mathematics Department, 2008.
- [25] B. PHILIP, L. CHACON, AND M. PERNICE, *Implicit Adaptive Mesh Refinement for 2D Reduced Resistive Magnetohydrodynamics*, J. Comp. Phys, 227 (2008), pp. 8855–8874.
- [26] U. RUEDE, *Mathematical and Computational Techniques for Multilevel Adaptive Methods*, SIAM, Philadelphia, 1995.
- [27] J. W. RUGE, *Fospack users manual*, Unpublished, v1.0 ed., 2000.
- [28] H. R. STRAUSS, *Nonlinear, Three-Dimensional Magnetohydrodynamics of Noncircular Tokamaks*, Physics of Fluids, 19 (1976), pp. 134–140.
- [29] H. SUNDAR, R. S. SAMPATH, AND G. BIROS, *Bottom-Up Construction and 2:1 Balance Refinement of Linear Octrees in Parallel*, SIAM J. Sci. Comp. (SISC), 30 (2008), pp. 2675–2708.
- [30] T. TU, D. O’HALLARON, AND O. GHATTAS, *Scalable Parallel Octree Meshing for Terascale Applications*, in Proceedings of ACM/IEEE SC05, (2005).
- [31] R. VERFÜRTH, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley-Teunber, 1995.
- [32] X. ZHANG, *Multilevel Schwarz Methods*, Numer. Math, 63 (1992), pp. 521–539.