ASYNCHRONOUS FAST ADAPTIVE COMPOSITE-GRID METHODS: NUMERICAL RESULTS

BARRY LEE *, STEPHEN F. MCCORMICK†, BOBBY PHILIP *, AND DANIEL J. QUINLAN *

Abstract. This paper presents numerical results for the Asynchronous Fast Adaptive Composite-Grid algorithm (AFACx), an asynchronous version of the Fast Adaptive Composite-Grid algorithm (FAC). A companion paper presented new multilevel condition number estimates to theoretically establish level-independent convergence rates for AFACx. In this paper, we present both numerical experiments supporting those theoretical results and numerical experiments for system partial differential equations. The latter experiments apply AFACx to first-order system least-squares finite element discretizations of the stationary Stokes equations on curvilinear adaptive mesh refinement grids.

1. Introduction. Adaptive mesh refinement (AMR) is a numerical process for dynamically introducing local fine resolution on computational grids, in response to unresolved error in a computation. AMR techniques were first introduced by Brandt [4] for general problems in the early 1970's and by Berger and Oliger [2] more specifically for hyperbolic problems in the 1980's. They are extremely attractive because typically one to two, and even possibly higher, orders of computational and memory efficiency improvements are achieved.

The Fast Adaptive Composite-Grid (FAC) method was developed in the 1980's [15, 16, 17, 18] to provide more robust discretization and solution methods for elliptic problems on AMR grids. The strength of FAC lies in its ability to use existing single grid solvers on uniform meshes to solve different refinement levels. This has the combined effect of solving non-uniform composite-grid problems with uniform grid solvers. However, although FAC permits asynchronous processing of grids at a given refinement level, and although its convergence rate is bounded independently of the number of refinement levels, the multiplicative way FAC treats the various refinement levels implies sequential processing, a serious bottleneck for large-scale parallel AMR applications. This difficulty led to the development of an asynchronous version of FAC, the AFAC algorithm [13, 16, 19]. Further research into improving the computational efficiency of AFAC led to the development of the AFACx algorithm [22, 20, 21]. AFACx achieves improvements in computational and communication efficiency by applying only simple relaxation sweeps on all but the coarsest refinement level, unlike AFAC, which solves each level using suitable solvers. Numerical results [22] show that these improvements come with no significant degradation of the convergence rates of AFACx when compared to the convergence rates of AFAC based on multigrid solvers.

In this paper, we present numerical results for the AFACx algorithm on Cartesian and curvilinear AMR grids. Since the success of adaptive solution methods depends critically on an accurate, computable, local error indicator, we present model applications involving first-order system least-squares (FOSLS) formulations of scalar elliptic partial differential equations (PDE's) and the stationary Stokes equations. FOSLS formulations are endowed with computable local error estimators; in particular, the local evaluation of the FOSLS functional [3, 10]. Numerical results verify the level

*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48.

†Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO, USA

independent convergence rates theoretically established in our companion paper [21]. To our knowledge, the AFACx algorithm has not been applied to system PDE's. Hence, these numerical results are of interest in themselves.

This paper proceeds as follow. In Section 2 we describe the algorithmic details of AFACx applied to a model PDE. Section 3 presents FOSLS formulations of a scalar elliptic equation and the stationary Stokes equations. We first establish notation and functional settings, followed by FOSLS formulations of the model problems, stating known theoretical results for these formulations. Section 4 reports numerical results for AFACx applied to finite element discretizations of these model problems on Cartesian and curvilinear AMR grids.

2. Model Problem and Algorithm Description. In this section we describe the AFACx algorithm. We do this by first presenting a model problem, its variational formulation, and its discretization on partially refined meshes. We then apply AFACx to this model problem. Note that the model problem used in this section is purely for illustrative purposes.

2.1. Model Problem. Consider a general self-adjoint boundary value problem

$$Lu = b.$$

Let the variational form of this problem be: find $u \in V$ such that

$$a(u,v) = f(v) \quad \forall v \in V, \tag{2.1}$$

where $(V, \|\cdot\|_V)$ a Hilbert space, $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ is a symmetric, continuous, and coercive bilinear form on V, i.e., there exists a constant, $\gamma > 0$, such that

$$a(u,u) \geq \gamma \|u\|_V^2 \quad \forall u \in V, \tag{2.2}$$

and $f(\cdot) : V \to \mathbb{R}$ is linear and continuous. It is known [11] that under these conditions there exists a unique solution, $u \in V$, for (2.1).

For simplicity in our illustration of the AFACx algorithm, we consider the model bvp

$$-\Delta u = f \text{ in } \Omega, \tag{2.3}$$
$$u = 0 \text{ on } \Gamma, \tag{2.4}$$

where $f \in L^2(\Omega)$. We choose $V = H_0^1(\Omega)$,

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v d\Omega \;\; \forall u,\ v \in V,$$

and

$$f(v) = \int_\Omega f v d\Omega \;\; \forall v \in V.$$

2.2. Partially Refined Meshes. To discretize (2.1) on partially refined meshes, we introduce the following notation. Let $\Omega_J \subseteq \Omega_{J-1} \ldots \subseteq \Omega_1 \equiv \Omega$ be a nested sequence of non-empty, bounded, open, Lipschitz domains. Subdomains $\Omega_k,\ k = 2, 3, \ldots, J$, can be viewed as regions where the solution can vary on increasingly finer scales and hence, regions where local refinement patches will be generated during the AMR process. Note that increasing values of the index k correspond to increasing

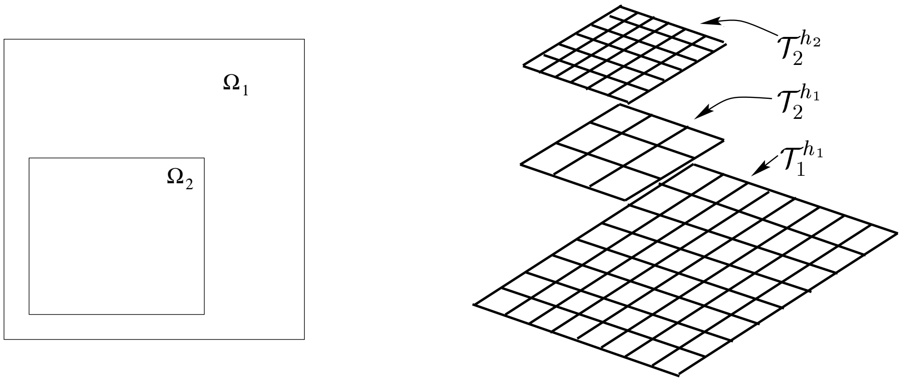

FIG. 2.1. *(a) Nested domains Ω_1 and Ω_2, and (b) tessellations $\mathcal{T}_1^{h_1}$, $\mathcal{T}_2^{h_1}$, and $\mathcal{T}_2^{h_2}$.*

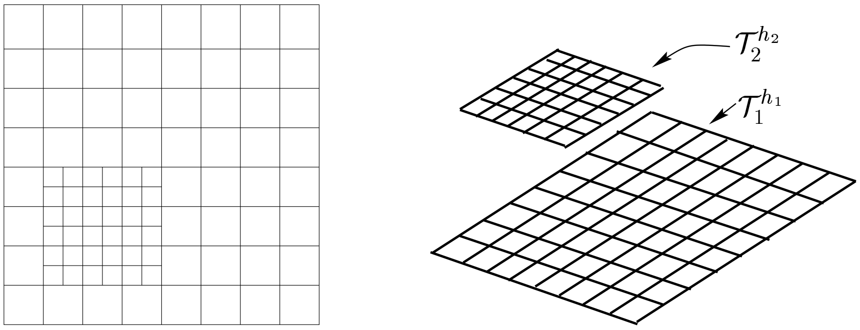

FIG. 2.2. *(a) Composite-grid $\mathcal{T}_2^c$ (b) composed of uniform subgrids, $\mathcal{T}_1^{h_1}$, $\mathcal{T}_2^{h_2}$.*

levels of refinement. Let $\mathcal{T}_1^c = \{\tau_i^1\}_{i=1}^{N_1}$ be a quasi-uniform tessellation of Ω_1, $N_1 \geq 4$. Tessellation $\mathcal{T}_k^c = \{\tau_i^k\}_{i=1}^{N_k}$, $k = 2, 3, \ldots, J$, of Ω, is obtained from $\mathcal{T}_{k-1}^c$ in the following manner. Since $\Omega_k \subseteq \Omega_{k-1}$, then there exists a local "coarse" tessellation, $\mathcal{T}_k^{h_{k-1}} = \{\tau_{i_j}^{k-1}\}_{j=1}^{M_k}$, $M_k \leq N_{k-1}$, consisting of elements of $\mathcal{T}_{k-1}^c$ that covers Ω_k, where h_{k-1} is the length of the longest side of the rectangles in $\mathcal{T}_k^{h_{k-1}}$. Here, we assume that the boundary of Ω_k aligns with the edges of the elements of $\mathcal{T}_{k-1}^c$ so that $\mathcal{T}_k^{h_{k-1}}$ becomes a quasi-uniform tessellation of Ω_k. Now we uniformly refine elements of $\mathcal{T}_k^{h_{k-1}}$ by subdividing each element into four elements by connecting the midpoints of the sides. This yields a "fine" local tessellation, $\mathcal{T}_k^{h_k}$, with $h_k = h_{k-1}/2$. Elements of $\mathcal{T}_{k-1}^c$ that lie in the complement of Ω_k and the elements of $\mathcal{T}_k^{h_k}$ together form the elements of $\mathcal{T}_k^c = (\mathcal{T}_{k-1}^c \backslash \mathcal{T}_k^{h_{k-1}}) \cup \mathcal{T}_k^{h_k}$. This process leads to a set of nested tessellations, $\{\mathcal{T}_k^c\}_{k=1}^J$, of Ω that form partially refined locally quasi-uniform meshes.

An example of this procedure is shown in Figure 2.1, where only one subdomain $\Omega_2 \subset \Omega_1 = \Omega$ is present. Ω_1 is tessellated into rectangular elements to obtain $\mathcal{T}_1^c \equiv \mathcal{T}_1^{h_1}$. This means that Ω_2 has a "coarse" tessellation (since $\Omega_2 \subset \Omega_1$) generated by elements of $\mathcal{T}_1^c$. This "coarse" tessellation of Ω_2 is denoted $\mathcal{T}_2^{h_1}$. The local "fine" tessellation, $\mathcal{T}_2^{h_2}$, of Ω_2, is obtained by subdividing each element of $\mathcal{T}_2^{h_1}$ into four smaller rectangular elements, as shown in Figure 2.1(b). $\mathcal{T}_2^c$ is the composite-grid tessellation shown in Figure 2.2(a) formed from the uniform subgrids shown in Figure 2.2(b).

2.3. Finite Element Spaces and the Discrete Variational Problem. Henceforth, we assume that conforming bilinear finite elements on quadrilateral meshes are used, although our results clearly apply to more general elements. We thus define $V_k^c \subset H_0^1(\Omega), k = 1, 2, \ldots J$, to be the space spanned by standard bilinear nodal basis functions with local support about the nodes of tessellation $\mathcal{T}_k^c$. Because of the conformity of the finite elements, note that there are no degrees of freedom associated with fine nodes that lie on the boundary $\partial\Omega_k$. Continuity implies that these "slave" nodes are evaluated simply by interpolation from adjacent coarse nodes. Now, the "fine" local finite element space defined in the interior of domain Ω_k is $V_k^{h_k} = V_k^c \cap H_0^1(\Omega_k)$. By our use of $H_0^1(\Omega_k)$ here, we mean that functions in $V_k^{h_k}$ have support only in the interior of Ω_k. Similarly, we define "coarse" local finite element spaces by $V_k^{h_{k-1}} = V_{k-1}^c \cap H_0^1(\Omega_k), V_k^{h_{k-1}} \subset V_k^{h_k},\ k = 2, \ldots, J$. Note that the local spaces are nested: $V_1^c \subseteq V_2^c \ldots \subseteq V_J^c \subset H_0^1(\Omega)$. However, the coarse local spaces are generally nonnested because they typically correspond to increasingly smaller local subdomains.

Having chosen finite-dimensional composite-grid space V_J^c, the discrete variational problem is: find $u^c \in V_J^c$ such that

$$a(u^c, v) = f(v) \ \ \forall v \in V_J^c, \tag{2.5}$$

where V_J^c is a finite-dimensional subspace of $H_0^1(\Omega)$.

2.4. AFACx. Let $u^c \in V_J^c$ denote the current approximation to the solution of (2.5). The AFACx algorithm applied to solving (2.5) with ν_1 smoothing steps on the restricted coarse grids at each level and ν_2 smoothing steps on the fine grids at each refinement level is then given by the expression

$$u^c \longleftarrow AFACx(u^c, \nu_1, \nu_2)$$

and is defined as follows:

Step 1: For each $k \in \{2, \cdots, J\}$, starting with initial guess $w^k \leftarrow 0$, perform ν_1 relaxation sweeps on the problem:

find $w^k \in V_k^{h_{k-1}}$ such that

$$a(u^c + w^k, w) = f(w) \ \ \forall w \in V_k^{h_{k-1}}.$$

On the coarsest level solve: find $u^1 \in V_1^c$ such that

$$a(u^c + u^1, v) = f(v) \ \ \forall v \in V_1^c.$$

Step 2: For each $k \in \{2, \cdots, J\}$, starting with initial guess $u^k \leftarrow 0$, perform ν_2 relaxation sweeps on the problem:

find $u^k \in V_k^{h_k}$ such that

$$a(u^c + w^k + u^k, v) = f(v) \ \ \forall v \in V_k^{h_k}.$$

Step 3: Update u^c by

$$u^c \longleftarrow u^c + \sum_{k=1}^{J} u^k.$$

By relaxation we mean application of smoothers like Richardson, damped Jacobi, or red-black Gauss Seidel.

The above pseudo-language shows that AFACx performs smoothing on all but the coarsest level. Smoothing is performed on the restricted coarse level to obtain the correction w^k on each level k. Smoothing on the fine level with initial guess w^k then yields u^k, which approximates the component of the correction that is representable only on level k.

AFACx is generally more efficient than AFAC because the various uniform grids (the local fine and restricted coarse refinement levels) are processed only by smoothing, instead of the somewhat more expensive multigrid solvers used in AFAC. This reduction in cost comes with no significant degradation in the convergence rates [22].

2.5. Symmetric AFACx. The operator corresponding to the AFACx algorithm described above is not symmetric with respect to the A-inner product. To facilitate condition number estimates, we work instead with a symmetrized form of AFACx developed as follows. Let $u^c \in V_J^c$ denote the current approximation to the solution of composite-grid equation (2.5).

The symmetric AFACx algorithm is then given by the expression

$$u^c \longleftarrow symmAFACx(u^c)$$

and is defined as follows:

Step 1: For each $k \in \{2, \cdots, J\}$, starting with the initial guess $w_1^k \leftarrow 0$, perform one relaxation sweeps on the problem:

find $w_1^k \in V_k^{h_{k-1}}$ such that

$$a(u^c + w_1^k, w) = f(w) \;\; \forall w \in V_k^{h_{k-1}}.$$

On the coarsest level solve: find $u^1 \in V_1^c$ such that

$$a(u^c + u^1, v) = f(v) \;\; \forall v \in V_1^c.$$

Step 2: For each $k \in \{2, \cdots, J\}$, starting with the initial guess $u_1^k \leftarrow 0$, perform one relaxation sweeps on the problem:

find $u_1^k \in V_k^{h_k}$ such that

$$a(u^c + w_1^k + u_1^k, v) = f(v) \;\; \forall v \in V_k^{h_k}.$$

Step 3: For each $k \in \{2, \cdots, J\}$, starting with the initial guess $u_2^k \leftarrow 0$, perform one relaxation sweeps on the problem:

find $u_2^k \in V_k^{h_k}$ such that

$$a(u^c + u_2^k, v) = f(v) \;\; \forall v \in V_k^{h_k}.$$

Step 4: For each $k \in \{2, \cdots, J\}$, starting with the initial guess $w_2^k \leftarrow 0$, perform one relaxation sweeps on the problem:

find $w_2^k \in V_k^{h_{k-1}}$ such that

$$a(u^c + u_2^k + w_2^k, w) = f(w) \;\; \forall w \in V_k^{h_{k-1}}.$$

Set u_2^k by $u_2^k \leftarrow u_2^k + w_2^k - w_1^k$.
Step 5: For each $k \in \{2, \cdots, J\}$, set u^k by

$$u^k \leftarrow \frac{1}{2}(u_1^k + u_2^k).$$

Step 6: Update u^c by

$$u^c \longleftarrow u^c + \sum_{k=1}^{J} u^k.$$

In practice, the unsymmetric form of AFACx is used while the symmetric form is useful for theoretical analysis. The following theorem was established in [20]:

THEOREM 2.1. *There exists a constant $C > 0$, independent of the number of levels J, such that*

$$\kappa(AFACx) \le C,$$

where $\kappa(AFACx)$ is the condition number of the operator $AFACx$, corresponding to symmetrized AFACx.

3. First-Order System Least Squares Formulations. Having described the AFACx algorithm, we now turn our attention on formulating our model problems as First-Order System Least-Squares (FOSLS) systems. In this section, we introduce notation, review FOSLS methodology, state some relevant FOSLS theory, and present the FOSLS formulations for a model elliptic PDE and for the Stokes equations. For a more thorough presentation of FOSLS methodology and additional references we refer the reader to [6, 7, 8, 9, 10].

3.1. Notation and Definitions. Let Ω be a bounded, open and connected domain in $\mathbb{R}^n$ $(n = 2, 3)$ with a $C^{1,1}$ or convex polyhedral boundary Γ. Let $\mathbf{u} = (u_1,\ u_2,\ \ldots,\ u_n)^t$ be an n-vector function defined in Ω. A scalar operator G applied to the components of $\mathbf{u}$ will be denoted by

$$G\mathbf{u}^t = (Gu_1, Gu_2, \ldots, Gu_n)$$

and

$$G\mathbf{u} = \begin{pmatrix} Gu_1 \\ Gu_2 \\ \vdots \\ Gu_n \end{pmatrix}.$$

Given a collection of column vectors $\mathbf{U}_i \equiv Gu_i,\ i = 1, 2, \cdots, n$, we denote the matrix composed of these columns as

$$\begin{aligned} \underline{\mathbf{U}} \equiv G\mathbf{u}^t &= (\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_n) \\ &= \begin{pmatrix} U_{11} & U_{12} & \ldots & U_{1n} \\ U_{21} & U_{22} & \ldots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ U_{n1} & U_{n2} & \ldots & U_{nn} \end{pmatrix}. \end{aligned}$$

For a square matrix $\underline{\mathbf{U}}$, we define its trace to be

$$\operatorname{tr}\underline{\mathbf{U}} = \sum_{i=1}^{n} U_{ii}.$$

Also, given an n-vector operator D (e.g., $D = \nabla\times$), we denote the operator D on the columns of $\underline{\mathbf{U}}$ as

$$D\underline{\mathbf{U}} = (D\mathbf{U}_1, D\mathbf{U}_2, \dots, D\mathbf{U}_n).$$

We further need to define boundary operations. Let $\mathbf{n}$ denote the outward normal on Γ. Then the tangential and normal operator on $\underline{\mathbf{U}}$ are extended componentwise as

$$\mathbf{n}\times\underline{\mathbf{U}} = (\mathbf{n}\times\mathbf{U}_1, \mathbf{n}\times\mathbf{U}_2, \dots, \mathbf{n}\times\mathbf{U}_n)$$

and

$$\mathbf{n}\cdot\underline{\mathbf{U}} = (\mathbf{n}\cdot\mathbf{U}_1, \mathbf{n}\cdot\mathbf{U}_2, \dots, \mathbf{n}\cdot\mathbf{U}_n).$$

Because FOSLS formulations are essentially well-posed minimization principles defined over appropriate Sobolev spaces, we will need some functional spaces. First, inner products and norms on matrix and vector functions are defined componentwise, e.g.

$$\|\underline{\mathbf{U}}\|^2 = \sum_{i=1}^{n} \|\mathbf{U}_i\|^2 = \sum_{i,j=1}^{n} \|U_{ij}\|^2.$$

For $m \geq 0$, we need the standard $m-th$ order Sobolev spaces $H^m(\Omega)^n$ with their standard norms denoted by $\|\cdot\|_{m,\Omega}$. For $m = 0$, we have $L^2(\Omega)^n$ with its inner product $(\cdot,\cdot)_{0,\Omega}$. Our FOSLS formulations require the spaces

$$\begin{aligned} H(\mathrm{div};\Omega) &= \{\mathbf{v}\in L^2(\Omega)^n \ :\ \nabla\cdot\mathbf{v}\in L^2(\Omega)\}, \\ H(\mathbf{curl};\Omega) &= \{\mathbf{v}\in L^2(\Omega)^n \ :\ \nabla\times\mathbf{v}\in L^2(\Omega)^{2n-3}\}, \end{aligned}$$

which are Hilbert spaces under the respective norms

$$\begin{aligned} \|\mathbf{v}\|_{H(\mathrm{div};\Omega)} &= (\|\mathbf{v}\|_{0,\Omega}^2 + \|\nabla\cdot\mathbf{v}\|_{0,\Omega}^2)^{\frac{1}{2}}, \\ \|\mathbf{v}\|_{H(\mathbf{curl};\Omega)} &= (\|\mathbf{v}\|_{0,\Omega}^2 + \|\nabla\times\mathbf{v}\|_{0,\Omega}^2)^{\frac{1}{2}}. \end{aligned}$$

Finally, we will need the spaces

$$\begin{aligned} \mathbf{H}_0(\mathbf{curl};\ \Omega) &= \{\mathbf{v}\in H(\mathbf{curl};\Omega) \ :\ \mathbf{n}\times\mathbf{v} = \mathbf{0} \text{ on } \Gamma\}, \\ L_0^2(\Omega) &= \{v\in L^2(\Omega) \ :\ \int_\Omega v\,\mathrm{d}\Omega = 0\ \}, \end{aligned}$$

which are subspaces of $H(\mathbf{curl};\Omega)$ and $L^2(\Omega)$, respectively.

3.2. Model Problem I. The first model problem we consider is

$$-\Delta p + \delta p = f \quad \text{in} \quad \Omega, \tag{3.1}$$

$$p = 0 \quad \text{on} \quad \Gamma, \tag{3.2}$$

where p is the unknown, f is the source term, and $\delta \geq 0$. It is known that under our assumptions on Γ, the associated weak form of (3.1)-(3.2) is uniquely solvable in $H^2(\Omega)$ for any $f \in L^2(\Omega)$ (cf. [12]).

FOSLS methodology essentially reformulates (3.1)-(3.2) into an augmented first-order system boundary value problem (bvp), and then converts the resulting system bvp into a well-posed minimization problem. Thus, let $\mathbf{U} = \nabla p$. Then (3.1)-(3.2) can be rewritten as

$$\mathbf{U} - \nabla p = 0 \text{ in } \Omega, \tag{3.3}$$
$$-\nabla \cdot \mathbf{U} + \delta p = f \text{ in } \Omega, \tag{3.4}$$
$$p = 0 \text{ on } \Gamma. \tag{3.5}$$

Also, noting that $\nabla \times \nabla = \mathbf{0}$ in Ω, and $\mathbf{n} \times \nabla p = \mathbf{0}$ on Γ, we have

$$\nabla \times \mathbf{U} = \mathbf{0} \text{ in } \Omega, \tag{3.6}$$
$$\mathbf{n} \times \mathbf{U} = \mathbf{0} \text{ on } \Gamma. \tag{3.7}$$

The augmented first-order system is (3.3)-(3.7).

3.2.1. FOSLS Minimization Problem for Model Problem I. We recast (3.3)-(3.7) into a least-squares minimization problem over a suitable linear space. Let $\mathbf{V}_1 = (H(\mathrm{div};\Omega) \cap \mathbf{H}_0(\mathbf{curl};\Omega)) \times H_0^1(\Omega)$, and define the FOSLS functional $\mathcal{G}_1(\cdot,\ \cdot;\ \cdot) : \mathbf{V}_1 \to \mathbb{R}$

$$\mathcal{G}_1(\mathbf{V}, q;\ f) = \|\mathbf{V} - \nabla q\|_{0,\Omega}^2 + \| - \nabla \cdot \mathbf{V} + \delta q - f\|_{0,\Omega}^2 + \|\nabla \times \mathbf{V}\|_{0,\Omega}^2 \ \ \forall (\mathbf{V},\ q) \in \mathbf{V}_1. \tag{3.8}$$

Then the FOSLS minimization problem is: find $(\mathbf{U},\ p) \in \mathbf{V}_1$ such that

$$\mathcal{G}_1(\mathbf{U}, p;\ f) = \inf_{(\mathbf{V},\ q) \in \mathbf{V}_1} \mathcal{G}_1(\mathbf{V}, q;\ f). \tag{3.9}$$

The equivalent variational problem for (3.9) is: find $(\mathbf{U}, p) \in \mathbf{V}_1$ such that

$$\mathcal{F}_1(\mathbf{U}, p; \mathbf{V}, q) = f_1(\mathbf{V}, q) \ \ \forall (\mathbf{V},\ q) \in \mathbf{V}_1, \tag{3.10}$$

where the bilinear form $\mathcal{F}_1(\cdot;\cdot) : \mathbf{V}_1 \times \mathbf{V}_1 \to \mathbb{R}$ is defined by

$$\mathcal{F}_1(\mathbf{U}, p; \mathbf{V}, q) = (\mathbf{U} - \nabla p, \mathbf{V} - \nabla q)_{0,\Omega} + (-\nabla \cdot \mathbf{U} + \delta p, -\nabla \cdot \mathbf{V} + \delta q)_{0,\Omega} + (\nabla \times \mathbf{U}, \nabla \times \mathbf{V})_{0,\Omega},$$

and linear functional $f_1(\cdot) : \mathbf{V}_1 \to \mathbb{R}$ is given by

$$f_1(\mathbf{V}, q) = (f, -\nabla \cdot \mathbf{V} + \delta q)_{0,\Omega}.$$

We note the following theorem [7]:

THEOREM 3.1. *Assume that Ω is a bounded open domain in $\mathbb{R}^n$, $n = 2, 3$, with a $C^{1,1}$ or convex polygonal boundary. Then there exist positive constants α_1 and α_2 such that*

$$\alpha_1(\|\mathbf{U}\|_{1,\Omega}^2 + \|p\|_{1,\Omega}^2) \leq \mathcal{F}_1(\mathbf{U}, p;\ \mathbf{U}, p) \ \ \forall (\mathbf{U}, p) \in \mathbf{V}_1 \tag{3.11}$$

and

$$\mathcal{F}_1(\mathbf{U},p;\ \mathbf{V},q) \le \alpha_2(\|\mathbf{U}\|_{1,\Omega}^2 + \|p\|_{1,\Omega}^2)^{\frac{1}{2}}(\|\mathbf{V}\|_{1,\Omega}^2 + \|q\|_{1,\Omega}^2)^{\frac{1}{2}} \ \ \forall (\mathbf{U},p),\ (\mathbf{V},q) \in \mathbf{V}_1. \tag{3.12}$$

Noting that $\mathcal{F}_1(\mathbf{U},p;\ \mathbf{U},p) = \mathcal{G}_1(\mathbf{U},p;\ 0)$, Theorem 3.1 immediately implies that

$$\alpha_1(\|\mathbf{U}\|_{1,\Omega}^2 + \|p\|_{1,\Omega}^2) \le \mathcal{G}_1(\mathbf{U},p;\ 0) \le \alpha_2(\|\mathbf{U}\|_{1,\Omega}^2 + \|p\|_{1,\Omega}^2) \ \ \forall (\mathbf{U},p) \in \mathbf{V}_1, \tag{3.13}$$

establishing $(H^1)^{n+1}$ norm equivalence of functional $\mathcal{G}_1^{\frac{1}{2}}(\cdot;\ \mathbf{0})$.

3.3. Model Problem II. The second model problem we consider is the stationary Stokes problem:

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f} \ \text{ in } \ \Omega, \tag{3.14}$$

$$\nabla \cdot \mathbf{u} = 0 \ \text{ in } \ \Omega, \tag{3.15}$$

$$\mathbf{u} = \mathbf{0} \ \text{ on } \ \Gamma, \tag{3.16}$$

$$\int_\Omega p \mathrm{dx} = 0, \tag{3.17}$$

where $(\mathbf{u},p)$ is the $(n+1)$-vector of unknowns and $\mathbf{f}$ is the source. It is known that the weak form of boundary value problem (3.14)-(3.17) has a unique solution $(\mathbf{u},p) \in H_0^1(\Omega)^n \times L_0^2(\Omega)$ for any $\mathbf{f} \in H^{-1}(\Omega)^n$ (cf. [12]).

3.3.1. First-Order System for Model Problem II. Introducing velocity gradients $\underline{\mathbf{U}} = \nabla \mathbf{u}^t = (\nabla u_1, \nabla u_2, \ldots, \nabla u_n)$, (3.14)-(3.17) can be rewritten as

$$\underline{\mathbf{U}} - \nabla \mathbf{u}^t = \underline{\mathbf{0}} \ \text{ in } \ \Omega, \tag{3.18}$$

$$-(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p = \mathbf{f} \ \text{ in } \ \Omega, \tag{3.19}$$

$$\nabla \cdot \mathbf{u} = 0 \ \text{ in } \ \Omega, \tag{3.20}$$

$$\mathbf{u} = \mathbf{0} \ \text{ on } \ \Gamma. \tag{3.21}$$

Moreover, we have the equations

$$\nabla \times \underline{\mathbf{U}} = \underline{\mathbf{0}} \ \text{ in } \ \Omega, \tag{3.22}$$

$$\nabla \mathrm{tr} \underline{\mathbf{U}} = 0 \ \text{ in } \ \Omega, \tag{3.23}$$

$$\mathbf{n} \times \underline{\mathbf{U}} = \underline{\mathbf{0}} \ \text{ on } \ \Gamma, \tag{3.24}$$

so that the augmented first-order bvp is (3.18)-(3.24).

3.3.2. FOSLS Minimization Problem for Model Problem II. Define spaces

$$\mathcal{V}_0 = \{\underline{\mathbf{V}} \in H^1(\Omega)^{n^2} \ : \ \mathbf{n} \times \underline{\mathbf{V}} = \underline{\mathbf{0}} \text{ on } \Gamma\}$$

and

$$\mathbf{V}_2 = \mathcal{V}_0 \times H_0^1(\Omega)^n \times (H^1(\Omega) \backslash \mathbb{R}),$$

and FOSLS functional $\mathcal{G}_2(\cdot;\mathbf{f}) : \mathbf{V}_2 \to \mathbb{R}$

$$\begin{aligned}\mathcal{G}_2(\underline{\mathbf{V}},\mathbf{v},q;\mathbf{f}) = {} & \|\mathbf{f} + (\nabla \cdot \underline{\mathbf{V}})^t - \nabla q\|_{0,\Omega}^2 + \|\underline{\mathbf{V}} - \nabla \mathbf{v}^t\|_{0,\Omega}^2 \\ & + \|\nabla \times \underline{\mathbf{V}}\|_{0,\Omega}^2 + \|\nabla \cdot \mathbf{v}\|_{0,\Omega}^2 + \|\nabla \mathrm{tr} \underline{\mathbf{V}}\|_{0,\Omega}^2 \ \ \forall (\underline{\mathbf{V}},\mathbf{v},q) \in \mathbf{V}_2.\end{aligned}$$

Then the FOSLS minimization problem for Model Problem II is: find $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{V}_2$ such that

$$\mathcal{G}_2(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{f}) = \inf_{(\underline{\mathbf{V}}, \mathbf{v}, q) \in \mathbf{V}_2} \mathcal{G}_2(\underline{\mathbf{V}}, \mathbf{v}, q; \mathbf{f}). \tag{3.25}$$

The variational problem associated with (3.25) is: find $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{V}_2$ such that

$$\mathcal{F}_2(\underline{\mathbf{U}}, \mathbf{u}, p; \underline{\mathbf{V}}, \mathbf{v}, q) = f_2(\underline{\mathbf{V}}, \mathbf{v}, q) \ \ \forall (\underline{\mathbf{V}}, \mathbf{v}, q) \in \mathbf{V}_2, \tag{3.26}$$

where bilinear form $\mathcal{F}_2(\cdot;\cdot) : \mathbf{V}_2 \times \mathbf{V}_2 \to \mathbb{R}$ is defined by

$$\begin{aligned}\mathcal{F}_2(\underline{\mathbf{U}}, \mathbf{u}, p; \underline{\mathbf{V}}, \mathbf{v}, q) = {} & ((\nabla \cdot \underline{\mathbf{U}})^t - \nabla p, (\nabla \cdot \underline{\mathbf{V}})^t - \nabla q)_{0,\Omega} + (\underline{\mathbf{U}} - \nabla \mathbf{u}^t, \underline{\mathbf{V}} - \nabla \mathbf{v}^t)_{0,\Omega} \\ & + (\nabla \times \underline{\mathbf{U}}, \nabla \times \underline{\mathbf{V}})_{0,\Omega} + (\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v})_{0,\Omega} + (\nabla \mathrm{tr} \underline{\mathbf{U}}, \nabla \mathrm{tr} \underline{\mathbf{V}})_{0,\Omega},\end{aligned}$$

and linear functional $f_2(\cdot) : \mathbf{V}_2 \to \mathbb{R}$ is given by

$$f_2(\underline{\mathbf{V}}, \mathbf{v}, q) = (\mathbf{f}, (\nabla \cdot \underline{\mathbf{V}})^t - \nabla q)_{0,\Omega}.$$

To establish continuity and coercivity bounds for bilinear form $\mathcal{F}_2(\cdot;\cdot)$, assume that domain Ω has a $C^{1,1}$ boundary and the following H^2 regularity result holds:

$$\|\mathbf{u}\|_{2,\Omega} + \|p\|_{1,\Omega} \le \gamma \|\mathbf{f}\|_{0,\Omega}, \tag{3.27}$$

where γ is a positive constant that depends only on domain Ω. Bound (3.27) is established in [14] for problem (3.14)-(3.17). Then the following theorem holds [9]:

THEOREM 3.2. *Assume that domain Ω has a $C^{1,1}$ boundary and that regularity bound (3.27) holds. Then there exists a constant $C > 0$ such that, for any $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{V}_2$,*

$$\frac{1}{C}(\|\underline{\mathbf{U}}\|^2_{1,\Omega} + \|\mathbf{u}\|^2_{1,\Omega} + \|p\|^2_{1,\Omega}) \le \mathcal{G}_2(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \tag{3.28}$$

and

$$\mathcal{G}_2(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \le C(\|\underline{\mathbf{U}}\|^2_{1,\Omega} + \|\mathbf{u}\|^2_{1,\Omega} + \|p\|^2_{1,\Omega}). \tag{3.29}$$

4. Numerical Results. This section presents numerical results for AFACx on adaptively-refined Cartesian and curvilinear coordinate grids. As mentioned earlier, prior to this work, AFACx has been applied only to scalar constant-coefficient diffusion equations on Cartesian grids. The numerical results presented here are for the model problems described in the previous section with inhomogeneous boundary conditions.

4.1. Model Problem I. We discretize the weak-form equations associated with the FOSLS formulation of Model Problem I presented in the previous section. Bilinear finite elements on adaptively-refined curvilinear grids are used for this purpose. The resulting linear systems are solved approximately using several iterations of AFACx.

To analyze the performance of AFACx, three subproblems with known nonzero solutions satisfying (3.1)-(3.2) are constructed. Let

$$\begin{aligned} p_1(x, y) &= \sin(\pi x)\sin(\pi y) \ \ \forall (x, y) \in \Omega, \\ p_2(x, y) &= x(1 - x)y(1 - y) \ \ \forall (x, y) \in \Omega,\end{aligned}$$

and

$$p_3(x,y) = 25e^{-625\pi[(x-x_0)^2+(y-y_0)^2]} \ \forall (x,y) \in \Omega.$$

Then problems I(a), I(b), and I(c) are (3.1)-(3.2) with $\delta = 0$, and $f = -\Delta p_i$ and $\phi = p_i \mid_\Gamma$ for $i = 1, 2,$ and 3, respectively. Numerical experiments were also run for the case $\delta = 1$ producing similar results, and hence are not reported here. Domain Ω is chosen to be either the unit square $\Omega_a \equiv [0,1]^2$ or the quarter annulus region Ω_b depicted in Figure 4.1.

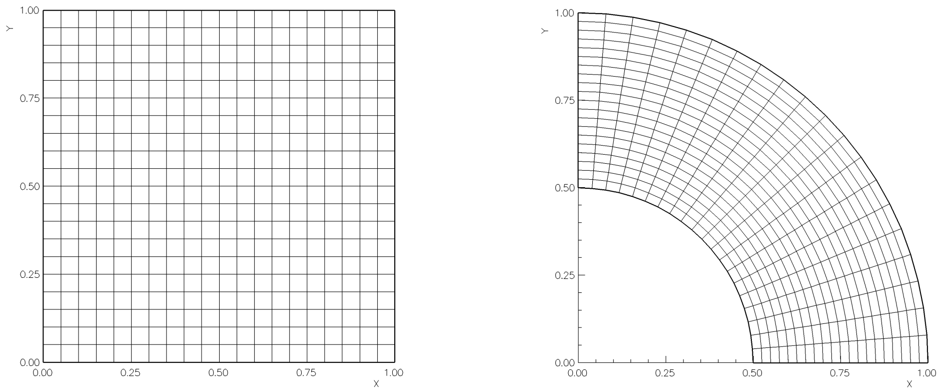

FIG. 4.1. *Domains* Ω_a *and* Ω_b *with uniform* 20×20 *grids.*

4.1.1. Asymptotic Convergence Factors for AFACx. To assess the efficiency of AFACx, we first measure asymptotic convergence factors. This is done by computing the ratios of the L^2 norms of successive residuals after many iterations of AFACx when these ratios settle to a constant limiting value. In practice, several grid patches can exist on each refinement level. However, for simplicity, our numerical experiments use only one uniform subgrid patch on each refinement level, with 20 grid lines in each coordinate direction on the coarsest grid, as shown in Figure 4.1. The effect of varying the number of refinement levels, and the number of restricted and fine grid smoothing steps on the asymptotic convergence factors is studied. To do this effectively, placement of the refinement patches is fixed in advance. The initial guess on each level is set to zero and point-Gauss-Seidel smoothing is used on every level but the coarsest, the current iterate values at each node being updated separately in turn. A direct solver capable of handling multiple right sides is used to solve the linear systems on the coarsest level. The initial cost of performing a LU factorization for the direct solve is amortized over multiple iterations of AFACx. The experiments were run on SUN Ultra-10 workstations in double-precision arithmetic.

Tables 4.1 and 4.2 present asymptotic convergence factors for problems I(a)-I(c) measured after 20 AFACx iterations. We study asymptotic convergence factors as the number of refinement levels (n), the number of restricted grid smoothing steps (ν_1), and the number of fine grid smoothing steps (ν_2) are varied. The numerical results show that asymptotic convergence factors do not degrade as the number of refinement levels is increased, confirming the theoretical results established in [21].

$(\nu_1,\ \nu_2) \rightarrow$	(1, 1)	(2, 1)	(2, 2)	(3, 1)	(3, 2)	(3, 3)	(4, 1)	(4, 2)	(4, 3)	(4, 4)
n=1	0.30	0.30	0.33	0.30	0.33	0.32	0.30	0.32	0.30	0.32
n=2	0.34	0.34	0.33	0.34	0.33	0.32	0.34	0.33	0.34	0.31
n=3	0.36	0.36	0.33	0.36	0.33	0.32	0.36	0.33	0.36	0.32
n=4	0.34	0.36	0.33	0.36	0.33	0.32	0.36	0.33	0.36	0.32
n=5	0.35	0.36	0.33	0.36	0.33	0.32	0.36	0.33	0.36	0.31
n=6	0.40	0.36	0.33	0.36	0.33	0.32	0.36	0.33	0.37	0.31
n=7	0.44	0.37	0.33	0.36	0.33	0.32	0.36	0.33	0.38	0.32
n=8	0.46	0.38	0.33	0.36	0.33	0.32	0.36	0.33	0.39	0.32
n=9	0.48	0.40	0.33	0.37	0.33	0.32	0.36	0.33	0.40	0.32
n=10	0.49	0.40	0.33	0.37	0.33	0.33	0.36	0.33	0.41	0.32

TABLE 4.1

Asymptotic convergence factors for point-Gauss-Seidel-based AFACx, problems I(a)-I(c), $\Omega = \Omega_a$.

$(\nu_1,\ \nu_2) \rightarrow$	(1, 1)	(2, 1)	(2, 2)	(3, 1)	(3, 2)	(3, 3)	(4, 1)	(4, 2)	(4, 3)	(4, 4)
n=1	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=2	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=3	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=4	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=5	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=6	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=7	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=8	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=9	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44
n=10	0.62	0.62	0.43	0.62	0.43	0.47	0.62	0.43	0.47	0.44

TABLE 4.2

Asymptotic convergence factors for point-Gauss-Seidel-based AFACx, problems I(a)-I(c), $\Omega = \Omega_b$.

n : number of refinement levels
ν_1: number of restricted grid smoothing steps at each level
ν_2: number of fine grid smoothing steps at each level

Moreover, increasing the number of restricted and fine grid smoothing steps does not significantly improve the performance of AFACx. This suggests that, to attain reasonable convergence factors, it is sufficient to perform a few smoothing steps on the restricted grid patches and a few smoothing steps on the fine grid patches.

4.1.2. Discretization Errors. To measure discretization errors, we use problems I(a)-I(c) again with f and ϕ chosen for each problem as described at the beginning of this section. Since the gradient variables $\mathbf{U} = \nabla p$ are computed during the FOSLS solution process, the exact values ∇p_i for $i = 1, 2$ and 3 are also computed. A sufficient number of AFACx iterations (typically 20 to 30 iterations) are used to find the approximate solution on composite-grids with various levels of discretization. Many iterations are used to ensure that the errors properly reflect the discretization accuracy. Figures 4.2-4.5 give an example of a series of uniformly refined composite-grids that are used in our computations. Note that by uniform refinement of a composite-grid we mean refinement of each of the uniform component-grid patches (in this case three) whose union forms the composite-grid. Tables 4.3 and 4.4 show the discretization errors and their convergence factors on uniformly refined composite-grids with different numbers of refinement levels. The data presented in these tables shows that discretization error on the composite-grids is $O(\overline{h}^2)$ in the L^2 norm, where $\overline{h}$ represents a local measure of the quasi-uniform composite-grid mesh spacing.

A sequence of uniformly refined composite-grids, each with 3 refinement levels.

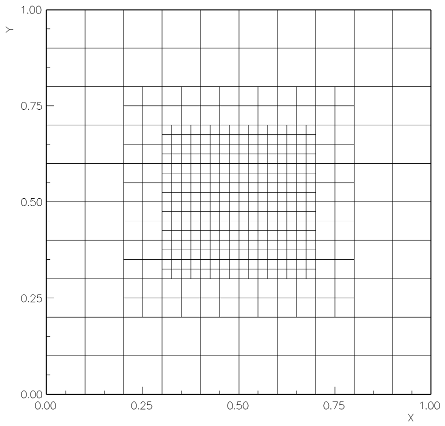

FIG. 4.2. *Composite-Grid 1*

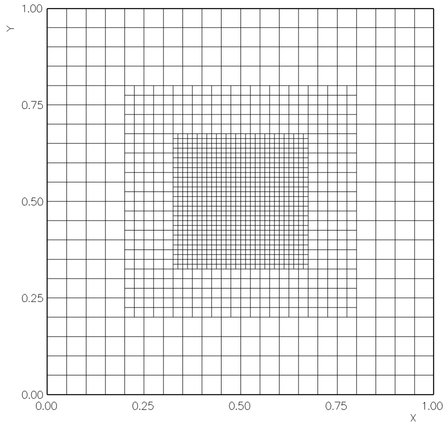

FIG. 4.3. *Composite-Grid 2*

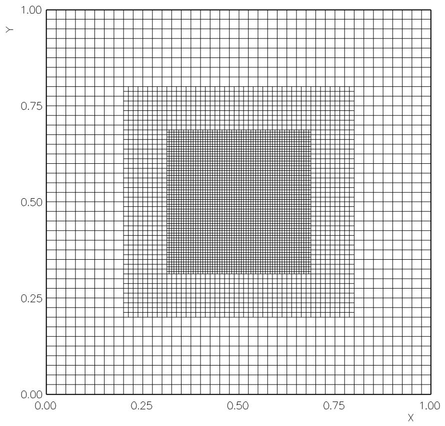

FIG. 4.4. *Composite-Grid 3*

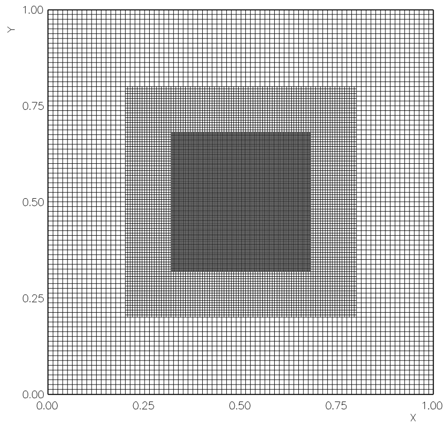

FIG. 4.5. *Composite-Grid 4*

4.2. Model Problem II. The variational equations derived from the FOSLS formulation of problem (3.14)-(3.17) described in the previous section are discretized using bilinear finite elements on adaptively refined meshes. Multiple iterations of AFACx are then used to solve the resulting linear system approximately. As for Model Problem I, a known nonzero solution satisfying (3.14)-(3.17) is constructed.

	Problem	I(a)		I(b)		I(c)	
N_r	N	L^2 norm	factor	L^2 norm	factor	L^2 norm	factor
1	10	0.09675	-	0.00747	-	1164.21	-
	20	0.02153	0.22261	0.00168	0.22575	296.529	0.25470
	40	0.00498	0.23152	0.00038	0.23116	11.1261	0.03752
	80	0.00119	0.23951	0.00009	0.23485	5.74593	0.51643
2	10	0.09286	-	0.00723	-	194.369	-
	20	0.02049	0.22066	0.00165	0.22824	6.70822	0.03451
	40	0.00472	0.23037	0.00038	0.23373	3.44755	0.51392
	80	0.00112	0.23884	0.00009	0.23611	0.84999	0.24655
3	10	0.09261	-	0.00721	-	4.51524	-
	20	0.02042	0.22059	0.00164	0.22849	2.01107	0.44539
	40	0.00470	0.23015	0.00038	0.23379	0.53124	0.26416
	80	0.00112	0.23884	0.00009	0.23615	0.12788	0.24072
4	10	0.09258	-	0.00721	-	1.43648	-
	20	0.02042	0.22060	0.00164	0.22851	0.31874	0.22189
	40	0.00470	0.23013	0.00038	0.23379	0.07937	0.24902
	80	0.00112	0.23884	0.00009	0.23615	0.01927	0.24281

N_r: number of refinement levels on each composite-grid
N: number of grid lines in each direction on the coarsest uniform grid

TABLE 4.3

Discretization errors and convergence factors for problems I(a)-I(c) on different composite-grids, $\Omega = \Omega_a$.

Let

$$\mathbf{u}_1(x,y) = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} (x(1-x))^2(2y-6y^2+4y^3) \\ -(2x-6x^2+4x^3)(y(1-y))^2 \end{pmatrix}$$

and

$$p_1(x,y) = x^2 - y^2 \ \ \forall (x,y) \in \Omega.$$

We then construct $\mathbf{f} = -\Delta \mathbf{u_1} + \nabla p_1$ and $\Phi = \mathbf{u}_1|_\Gamma$. The exact gradient matrix $\underline{\mathbf{U}} = \nabla \mathbf{u}_1$ is also constructed. Then $(\mathbf{u}_1, p_1)$ satisfies (3.14)-(3.17) for data $\mathbf{f}$ and Φ.

4.2.1. Asymptotic Convergence Factors. Asymptotic convergence factors for AFACx applied to this problem are measured by computing the ratio of the L^2 norms of the residuals at successive steps after a sufficient number of iterations. The domain, Ω, the number of refinement levels, the number of fine grid and restricted grid smooths are all varied to study their effects on the asymptotic convergence factors. To perform numerical experiments without introducing effects due to the specific re-gridding and error estimation algorithms used, we fix the placement of the refinement patches. In addition, for simplicity, as before, each refinement level has only one grid patch. The experiments were again run on SUN Ultra-10 workstations using double precision arithmetic.

Tables 4.6 and 4.7 present the asymptotic convergence factors for AFACx applied to the Stokes model problem after 30 iterations. A point-Gauss-Seidel smoother is used, where at each node in turn, the current iterate values for each variable are

	Problem	I(a)		I(b)		I(c)	
N_r	N	L^2 norm	factor	L^2 norm	factor	L^2 norm	factor
1	10	0.08737	-	0.00791	-	455.326	-
	20	0.02197	0.25154	0.00207	0.25878	107.223	0.23548
	40	0.00536	0.24397	0.00052	0.25221	11.9773	0.11170
	80	0.00131	0.24525	0.00013	0.25035	4.18633	0.34952
2	10	0.07474	-	0.00734	-	71.8310	-
	20	0.01814	0.24273	0.00193	0.26356	7.39164	0.10290
	40	0.00438	0.24175	0.00049	0.25608	2.51185	0.33982
	80	0.00107	0.24513	0.00012	0.25238	0.61129	0.24336
3	10	0.07455	-	0.00747	-	5.19679	-
	20	0.01805	0.24212	0.00169	0.22693	1.46525	0.28195
	40	0.00436	0.24180	0.00039	0.23216	0.38212	0.26078
	80	0.00107	0.24516	0.00009	0.23474	0.09156	0.23961
4	10	0.07449	-	0.00731	-	1.14432	-
	20	0.01803	0.24216	0.00193	0.26431	0.51540	0.45039
	40	0.00436	0.24179	0.00049	0.25613	0.14948	0.29002
	80	0.00106	0.24519	0.00012	0.25246	0.04507	0.30156

N_r: number of refinement levels on each composite-grid

N: number of grid lines in each direction on the coarsest uniform grid

TABLE 4.4

Discretization errors and convergence factors for problems I(a)-I(c) on different composite-grids, $\Omega = \Omega_b$.

$n \rightarrow$	1		2		3		4	
N	L^2 norm	factor	L^2 norm	factor	L^2 norm	factor	L^2 norm	factor
10	0.0074	-	0.0087	-	0.0092	-	0.0092	-
20	0.0019	0.260	0.0021	0.243	0.0022	0.238	0.0022	0.237
40	0.0004	0.217	0.0005	0.217	0.0005	0.218	0.0005	0.217
80	0.0001	0.222	0.0001	0.224	0.0001	0.224	0.0001	0.224
160	2.e-05	0.235	2.e-05	0.240	-	-	-	-

n : number of refinement levels

N : number of grid lines in each coordinate direction on coarsest uniform grid

TABLE 4.5

Discretization errors (L^2 norm) and their convergence factors for model problem II on different composite-grids, $\Omega = \Omega_a$.

updated separately. Experiments with block smoothers did not show any significant improvement in convergence rates, and hence are not reported here. The coarsest level was solved approximately using a fast transpose-free QMR solver from the PETSc library.

The numerical results indicate no degradation in asymptotic convergence factors for AFACx as the number of refinement levels is increased. Increasing the number of smoothing steps on the restricted and fine grid patches does not improve the convergence factors significantly.

Model Problem II: Computed approximations to the velocity $\mathbf{u} = (u,\ v)^t$ *and the pressure* p.

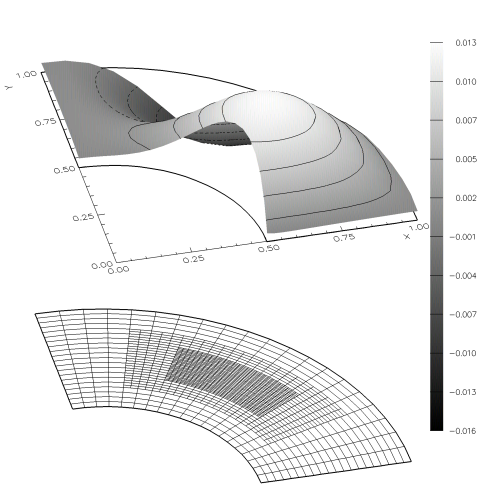

FIG. 4.6. *velocity (u component)*

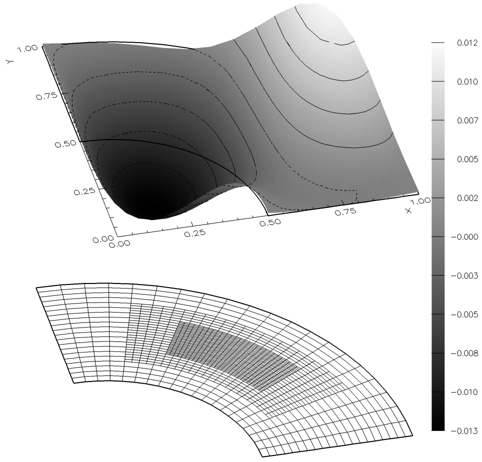

FIG. 4.7. *velocity (v component)*

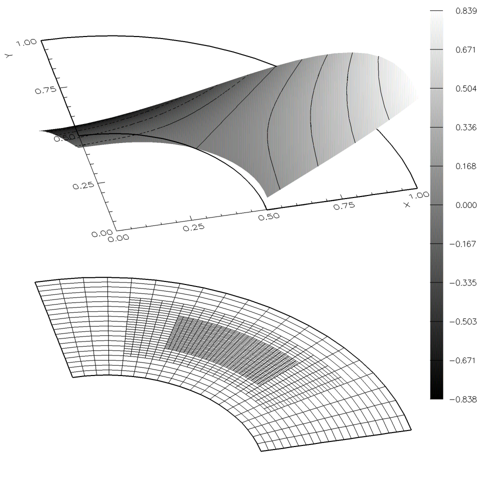

FIG. 4.8. *pressure (p)*

Model Problem II: Computed velocity gradient components $\nabla u = (u_x, u_y)^t$ *and* $\nabla v = (v_x, v_y)^t$.

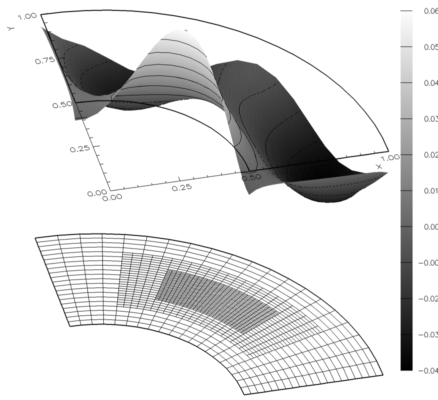

FIG. 4.9. u_x

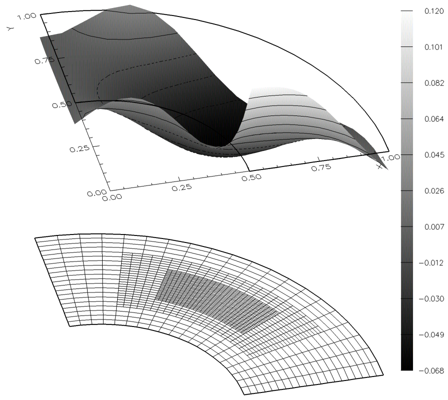

FIG. 4.10. u_y

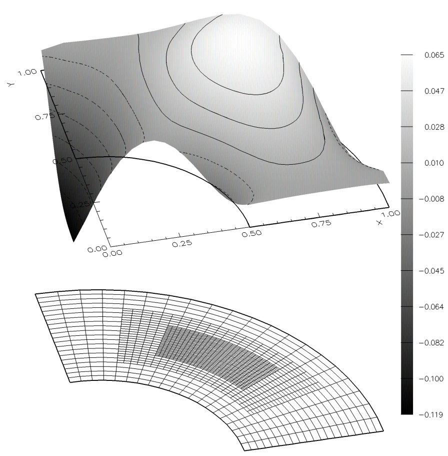

FIG. 4.11. v_x

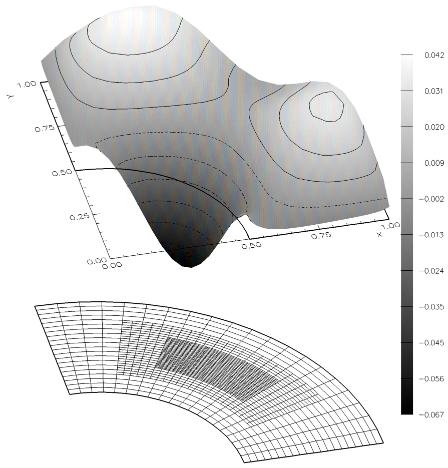

FIG. 4.12. v_y

$(\nu_1,\ \nu_2) \rightarrow$	(1, 1)	(2, 1)	(2, 2)	(3, 1)	(3, 2)	(3, 3)	(4, 1)	(4, 2)	(4, 3)	(4, 4)
n=1	0.81	0.80	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.53
n=2	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=3	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=4	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=5	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=6	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=7	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=8	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=9	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62
n=10	0.80	0.79	0.65	0.79	0.65	0.53	0.80	0.65	0.53	0.62

TABLE 4.6

Asymptotic convergence factors for point-Gauss-Seidel-based AFACx, problem II, $\Omega = \Omega_a$.

$(\nu_1,\ \nu_2) \rightarrow$	(1, 1)	(2, 1)	(2, 2)	(3, 1)	(3, 2)	(3, 3)	(4, 1)	(4, 2)	(4, 3)	(4, 4)
n=1	0.91	0.91	0.85	0.91	0.85	0.79	0.91	0.85	0.78	0.72
n=2	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=3	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=4	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=5	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=6	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=7	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=8	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=9	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71
n=10	0.91	0.91	0.84	0.91	0.84	0.77	0.91	0.84	0.77	0.71

TABLE 4.7

Asymptotic convergence factors for point-Gauss-Seidel-based AFACx, problem II, $\Omega = \Omega_b$.

n : number of refinement levels
$\nu_1,\ \nu_2$: number of restricted grid and fine grid smoothing steps

4.2.2. Discretization Errors. To measure discretization errors we solve problem II with $\mathbf{f}$ and Φ chosen as described at the beginning of this section. The exact gradients $\nabla \mathbf{u}_1$ are also computed. The approximate solution is then computed on composite-grids with various levels of discretization. This is done using a sufficient number of iterations (approximately 30) of AFACx to ensure that the errors properly reflect discretization accuracy.

Table 4.5 shows the discretization errors and their convergence factors on uniformly refined composite-grids with different numbers of refinement levels for $\Omega = \Omega_a$. The results show that the discretization error on the composite-grids is $O(\overline{h}^2)$ in the L^2 norm, where $\overline{h}$ represents a local measure of the quasi-uniform composite-grid mesh spacing.

4.3. Software and Implementation. The AFACx solver used for performing the numerical experiments was implemented in C++ and uses the AMR++ adaptive mesh refinement library [23] developed as part of the Overture framework [5]. Solvers from the PETSc library [1] are also used to perform global coarse grid solves within AFACx.

REFERENCES

[1] S. BALAY, W. D. GROPP, L. C. MCINNES, AND B. F. SMITH, *PETSc Users Manual*, Tech. Report ANL-95/11 - Revision 2.1.0, Argonne National Laboratory, 2001.
[2] M. BERGER, *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, PhD thesis, Stanford University, 1982.

[3] M. Berndt, T. Manteuffel, and S. McCormick, *Local Error Estimates and Adaptive Refinement for First-Order System Least Squares (FOSLS)*, Electronic Transactions on Numerical Analysis, 6 (1997), pp. 35–43.

[4] A. Brandt, *Multi-Level Adaptive Solutions to Boundary-Value Problems*, Mathematics of Computation, 31 (1977), pp. 333–390.

[5] D. Brown, G. Chesshire, W. Henshaw, and D. Quinlan, *Overture: An Object Oriented Software System for Solving Partial Differential Equations in Serial and Parallel Environments*, in Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, 1997.

[6] Z. Cai, T. Manteuffel, and S. McCormick, *First-Order System Least-Squares for Second-Order Partial Differential Equations: Part I*, SIAM Journal of Numerical Analysis, 31 (1994), pp. 1785–1799.

[7] ———, *First-Order System Least-Squares for Velocity-Vorticity-Pressure form of the Stokes Equations, with Applications to Linear Elasticity*, Electronic Transactions on Numerical Analysis, 3 (1995), pp. 150–159.

[8] ———, *First-Order System Least-Squares for Second-Order Partial Differential Equations: Part II*, SIAM Journal of Numerical Analysis, 34 (1997), pp. 425–454.

[9] ———, *First-Order System Least-Squares for the Stokes Equations, with Applications to Linear Elasticity*, SIAM Journal of Numerical Analysis, 34 (1997), pp. 1727–1741.

[10] J. Fiard, T. Manteuffel, and S. McCormick, *First-Order System Least-Squares (FOSLS) for Convection-Diffusion Problems: Numerical Results*, SIAM Journal of Scientific Computing, 19 (1998), pp. 1958–1979.

[11] D. Gilbarg and N. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer-Verlag, 1977.

[12] V. Girault and P. Raviart, *Finite Element Methods for Navier-Stokes Equations, Theory and Algorithms*, no. 5 in Springer Series in Computational Mathematics, Springer-Verlag, 1986.

[13] L. Hart and S. McCormick, *Asynchronous Multilevel Adaptive Methods for Solving Partial Differential Equations: Basic Ideas*, Parallel Computing, 12 (1989), pp. 131–144.

[14] O. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach, New York, 1963.

[15] S. McCormick, *Fast Adaptive Composite Grid (FAC) Methods: Theory for the Variational Case*, in Defect Correction Methods: Theory and Applications, K. Bohmer and H. Stetter, eds., vol. 5 of Computations Supplementation, Springer-Verlag, 1984, pp. 115–122.

[16] ———, *Multilevel Adaptive Methods for Partial Differential Equations*, no. 6 in Frontiers in Applied Mathematics, SIAM, 1989.

[17] ———, *Multilevel Projection Methods for Partial Differential Equations*, CBMS-NSF, SIAM, 1992.

[18] S. McCormick, S. McKay, and J. Thomas, *Computational Complexity of the Fast Adaptive Composite Grid (FAC) Method*, Applied Numerical Mathematics, 6 (1989), pp. 315–327.

[19] S. McCormick and D. Quinlan, *Asynchronous Multilevel Adaptive Methods for Solving Partial Differential Equations on Multiprocessors: Performance Results*, Parallel Computing, 12 (1989), pp. 145–156.

[20] B. Philip, *Asynchronous Fast Adaptive Composite Grid Methods for Elliptic Problems on Adaptively-Refined Curvilinear Grids*, PhD thesis, University of Colorado at Boulder, August 2001.

[21] B. Philip, S. McCormick, B. Lee, and D. Quinlan, *Asynchronous Fast Adaptive Composite-Grid Methods: Theoretical Results*, submitted to SIAM Journal of Scientific Computing, (2002).

[22] D. Quinlan, *Adaptive Mesh Refinement for Distributed Parallel Architectures*, PhD thesis, University of Colorado at Denver, 1993.

[23] D. Quinlan, *AMR++ Manual*, Tech. Report LA-UR-97-4325, Los Alamos National Laboratory, 1997.