GOAL-ORIENTED ADAPTIVITY AND MULTILEVEL PRECONDITIONING FOR THE POISSON-BOLTZMANN EQUATION

BURAK AKSOYLU[1], STEPHEN D. BOND[2], ERIC C. CYR[3], AND MICHAEL HOLST[4]

ABSTRACT. In this article, we develop goal-oriented error indicators to drive adaptive refinement algorithms for the Poisson-Boltzmann equation. Empirical results for the solvation free energy linear functional demonstrate that goal-oriented indicators are not sufficient on their own to lead to a superior refinement algorithm. To remedy this, we propose a problem-specific marking strategy using the solvation free energy computed from the solution of the linear regularized Poisson-Boltzmann equation. The convergence of the solvation free energy using this marking strategy, combined with goal-oriented refinement, compares favorably to adaptive methods using an energy-based error indicator. Due to the use of adaptive mesh refinement, it is critical to use multilevel preconditioning in order to maintain optimal computational complexity. We use variants of the classical multigrid method, which can be viewed as generalizations of the hierarchical basis multigrid and Bramble-Pasciak-Xu (BPX) preconditioners.

CONTENTS

1

1. INTRODUCTION

The Poisson-Boltzmann equation (PBE) is a widely used model for electrostatic interactions of charged bodies in dielectric media, such as molecules, ions, and colloids, and thus is of importance in many areas of science and engineering, including biochemistry, biophysics, and medicine. (See the classical texts [65, 74] for a derivation of the PBE.) The importance of the PBE model is reflected by the popularity of software packages such as APBS [9], CHARMM [34], DelPhi [70], and UHBD [64], within the molecular modeling communities. It provides a high fidelity mean-field description of electrostatic interactions and ionic distributions of solvated biomolecular systems in equilibrium. The partial differential equation itself is challenging to solve numerically due to singularities of different orders at the positions of permanent point charges and the presence of a dielectric interface.

In this article, we develop an adaptive multilevel finite element method for the PBE using *goal-oriented a posteriori error indicators*. This adaptive algorithm, which is a variant of that studied for the PBE in [55, 8, 9], deviates substantially from previous work in that the error indicator is based on a user defined quantity of interest or goal. This is in contrast to traditional residual-based adaptive refinement algorithms (like those developed for the PBE in [39]) that drive-refinement to minimize the global error measured in an energy-norm. The goal-oriented refinement methodology has been successfully employed in a wide range of application areas, including fluids, elasticity, and fluid structure interaction [11]. Despite these successes, we show that this methodology applied directly to the PBE does not necessarily lead to a successful adaptive algorithm. To remedy this issue we propose a novel marking strategy which recovers the performance commonly seen in other applications. This is the first time that this particular goal-oriented refinement strategy has been applied to the PBE specifically, and molecular biophysics in general.

At the core of any adaptive finite element approach are the iterative methods used to solve the discretized equation. However, due to the ill-conditioning of the linear systems arising from the discretization of the PBE, the convergence rate of traditional iterative solvers is significantly deteriorated. To remedy this, we combine modern Bramble-Pasciak-Xu (BPX)-type multilevel preconditioners with the goal-oriented adaptive algorithm mentioned above. When applied to the PBE, our results demonstrate that the overall algorithm is accurate, highly efficient and scalable with respect to the number of levels in the adaptive hierarchy.

An outline of the article is as follows. In section 2, we give a brief overview of the Poisson-Boltzmann equation, and describe the most useful formulations for modeling and numerical simulation, such as the regularized formulations described in [39, 56, 37]. We also discuss the solvation free energy functional corresponding to a given reaction potential, which will form the basis of our goal-oriented error indicators developed later in the article. We describe adaptive finite element methods in section 3, including weak formulation of the regularized PBE, discretization by finite element methods, and adaptive algorithms driven by *a posteriori* error indicators. In section 4, we describe a particular class of error indicators known as *goal-oriented* indicators, and describe several indicators designed for the PBE. In section 5, we discuss a local multigrid algorithm used to precondition an iterative Krylov method for solving the linear systems arising from adaptive mesh refinement. With some care, these methods enable an algorithm whose complexity is close to optimal. The results from a sequence of numerical experiments
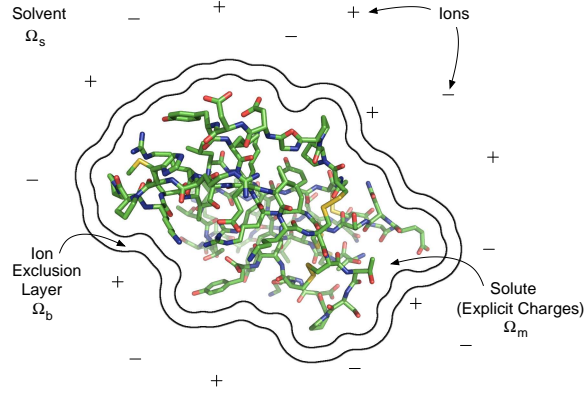
FIGURE 1. Debye-Hückel model of a charged biological structure immersed in a solvent containing mobile ions

using the Finite Element ToolKit (FETK) are presented in section 6. These results highlight the efficacy of the goal-oriented error indicator for the Poisson-Boltzmann problem, as well as the utility of the linear solver strategy combined with the adaptive algorithm, driven by the goal-oriented error indicator. We draw some conclusions in section 7.

2. THE POISSON-BOLTZMANN EQUATION

The Poisson-Boltzmann equation (PBE) is a second-order nonlinear partial differential equation whose solution gives the electrostatic potential, $\phi(x)$, for a solute molecule immersed in an implicitly defined solvent. Using a mean-field approximation, the solvent is treated as a bulk medium where ions are distributed according to the Boltzmann distribution. Figure 1 is a schematic representation of the domain of the PBE, denoted by Ω. The innermost region, Ω_m, contains the explicitly represented solute molecule. The outer region, Ω_s, is the bulk solvent and contains the implicit solvent ions. Between Ω_s and Ω_m is the ion exclusion layer, which separates the solute from the solvent ions, and has a width dependent on the size of the solvent ions. For simplicity, we will assume the solvent ions are small and the ion exclusion layer can be neglected. Hence, the interface between the solute and solvent is a surface, denoted by $\Gamma = \bar{\Omega}_m \cap \bar{\Omega}_s$. The shape of the surface is governed by the short-range repulsive van der Waals interactions, which prevent the solvent from penetrating the solute. The precise definition of the surface varies depending on the model [10].

The PBE for a 1:1 electrolyte (e.g., sodium chloride) is

$$-\nabla \cdot \epsilon(x)\nabla u(x) + \bar{\kappa}^2(x)\sinh\left(u(x)\right) = \frac{4\pi e_c}{k_B T}\sum_{i=1}^{P} q_i \delta(x - x_i), \quad x \in \Omega_m \cup \Omega_s,$$

$$u(\infty) = 0, \tag{2.1}$$

$$\left[\!\!\left[\epsilon(x)\frac{\partial u(x)}{\partial n} \right]\!\!\right] = 0, \quad x \in \Gamma,$$

where $u(x) = e_c\phi(x)/k_B T$ is the dimensionless potential, e_c is the charge of an electron, k_B is Boltzmann's constant, and T is the temperature. Here, $[\!\![\cdot]\!\!]$ denotes the jump across the interface

$$\left[\!\!\left[f(x) \right]\!\!\right] = \lim_{\zeta \to 0} f(x + \zeta n) - f(x - \zeta n) \tag{2.2}$$

and n is the outward pointing normal of $\partial\Omega_m$. The dielectric function $\epsilon(x)$ jumps one or two orders of magnitude at the interface Γ. For example, commonly used values are $\epsilon(\Omega_m) = \epsilon_m = 2$ and $\epsilon(\Omega_s) = \epsilon_s = 80$. The modified Debye-Hückel parameter, $\bar{\kappa}$, has a similar discontinuity, with $\bar{\kappa}(\Omega_m) = 0$ and $\bar{\kappa}(\Omega_s) = \bar{\kappa}_s > 0$. The fixed ions within the solute are represented by a sum of Dirac delta distributions, with fixed charge centers, x_i, and charges $e_c q_i$. This charge distribution induces singularities in the electrostatic potential and has, until recently, proved to be difficult to treat numerically.

To address this issue the PBE is reformulated so that the singularities are explicitly removed [48, 86]. Following [39, 56, 37], this is accomplished by writing the potential as a sum of a singular term u_c and a nonsingular remainder u_r. The singular term is the Coulomb potential

$$u_c(x) = \frac{e_c}{\epsilon_m k_B T} \sum_{i=1}^{P} \frac{q_i}{|x - x_i|}, \tag{2.3}$$

which satisfies the Poisson equation

$$-\nabla \cdot \epsilon_m \nabla u_c(x) = \frac{4\pi e_c}{k_B T} \sum_{i=1}^{P} q_i \delta(x - x_i) \quad \text{for} \quad x \in \Omega,$$
$$u_c(\infty) = 0. \tag{2.4}$$

There are numerous fast algorithms (with linear or near linear complexity) for evaluating u_c on a set of quadrature points, such as fast multipole [49], multilevel summation [72, 51] and particle mesh Ewald [46].

Substituting $u = u_r + u_c$ into the PBE (Eq. 2.1) gives a modified form of the PBE, which was termed in [39] as the Regularized PBE (RPBE):

$$-\nabla \cdot \epsilon(x)\nabla u_r(x) + \bar{\kappa}^2(x) \sinh\left(u_r(x) + u_c(x)\right)$$
$$= \nabla \cdot (\epsilon(x) - \epsilon_m)\nabla u_c(x), \quad x \in \Omega_m \cup \Omega_s,$$
$$u_r(\infty) = 0, \tag{2.5}$$
$$\left[\!\left[\epsilon(x)\frac{\partial u_r(x)}{\partial n}\right]\!\right] = (\epsilon_m - \epsilon_s)\frac{\partial u_c(x)}{\partial n}, \quad x \in \Gamma.$$

Note that, because both $\bar{\kappa}(x)$ and $(\epsilon(x) - \epsilon_m)$ are zero for $x \in \Omega_m$ and the centers of the atoms in the solute are well separated from Γ, the singular function, u_c, is never evaluated near the singularities. This formulation was used in [39] to develop continuous *a priori* L^∞ estimates of solutions, and subsequently to show existence and uniqueness of solutions of the PBE. Furthermore, the authors established discrete *a priori* L^∞ estimates for Galerkin solutions, making possible quasi-optimal *a priori* error estimates, as well as a provably convergent adaptive finite element method for the RPBE.

Recently, an alternative 3-term splitting of the PBE has been proposed in [56] which addresses the inherit subtractive cancellation in the reconstruction of the electrostatic potential used by the RPBE. In this article, the authors establish mathematical results for the alternative splitting, including continuous and discrete *a priori* L^∞ estimates, existence and uniqueness of solutions, quasi-optimal *a priori* error estimates, and a convergent adaptive finite element method (AFEM). (Whereas in [39] only AFEM convergence was shown, it was shown in [56] that AFEM is a contraction for the RPBE, using a new AFEM convergence framework for nonlinear problems developed in [57].) The 3-term

splitting decomposes the electrostatic potential into

$$u(x) = \begin{cases} u_3(x) + u_c(x) + u_h(x) & \text{in } \Omega_m \\ u_3(x) & \text{in } \Omega_s \end{cases}, \tag{2.6}$$

where u_c is the Coulomb potential, but here it is restricted to the subdomain Ω_m. The harmonic term, u_h, is defined as the solution to

$$-\nabla^2 u_h(x) = 0 \qquad \text{in } \Omega_m \tag{2.7}$$

$$u_h(x) = -u_c(x) \quad \text{on } \Gamma. \tag{2.8}$$

Applying the definitions of u_c, u_h and substituting into the PBE (Eq. 2.1), one obtains an equation for u_3,

$$-\nabla \cdot (\epsilon(x)\nabla u_3(x)) + \bar{\kappa}^2(x)\sinh(u_3(x)) = 0 \quad \text{in } \Omega, \tag{2.9}$$

$$\left[\!\!\left[\epsilon(x)\frac{\partial u_3(x)}{\partial n} \right]\!\!\right] = \epsilon_m \frac{\partial(u_c(x) + u_h(x))}{\partial n} \quad \text{on } \Gamma, \tag{2.10}$$

$$u_3(\infty) = 0. \tag{2.11}$$

In contrast to the RPBE, this formulation avoids the subtractive cancellation since $u_3 = u$ in Ω_s, and the Coulomb term is not used to reconstruct the potential in the solvent subdomain.

For systems which are not highly charged, the variation in the potential is relatively small, and the hyperbolic sine term is well approximated by its linearization. This approximation, which replaces $\sinh(u)$ with u, results in what is known as the linear Poisson-Boltzmann equation (LPBE), or the linear regularized Poisson-Boltzmann equation (LRPBE) in case of the RPBE. Although this approximation reduces the ionic response of the solvent [10, 47], it can significantly reduce the complexity of many numerical algorithms (e.g., boundary element and boundary integral methods) [63, 25, 62].

One important use for the solution to the PBE is in the calculation of solvation free energies. This quantity measures the thermodynamic work of moving the solute molecule from a vacuum to a solvent environment. The solvation free energy can be written as a sum of nonpolar and polar contributions. The nonpolar term depends on the solvent accessible surface area, excluded volume, and nonpolar forces which are typically assumed to be independent of the electrostatic potential [60, 78]. The polar term, S, is a linear functional of the solution to the RPBE, u_r, (also known as the reaction potential) [60] and can be expressed as

$$S(u_r) = \frac{1}{2}\int_\Omega u_r(x)\sum_{i=1}^P e_c\, q_i \delta(x - x_i)\, dx. \tag{2.12}$$

For the 3-term splitting, the reaction potential is the sum of u_3 and u_h.

3. ADAPTIVE FINITE ELEMENT METHODS

The finite element approach provides a natural framework for dealing with the complex molecular surfaces which arise in the PBE. Although there are modified finite difference methods which address this difficulty [83], finite element methods provide an attractive alternative when paired with an adaptive unstructured mesh designed to conform to the shape of the solute molecule [62]. In this section, we present a general adaptive finite element method for the regularized PBE, including the weak formulation, discretization, solution using an inexact global Newton iteration, and adaptive refinement procedure. For more details on the finite element method, see [73, 32, 26, 45].

3.1. **Weak Forms.** To give a well-defined weak formulation, the nonlinearity involving exponentials must be controlled; in [39, 56], *a priori* L^∞ estimates are obtained for any solution to the RPBE, giving almost everywhere pointwise bounds of the form: $\alpha \leqslant u_r \leqslant \beta$. This leads to working with a well-defined solution space that consists of a non-empty, topologically closed, convex subset of $H^1(\Omega)$:

$$M_e := \{ \, v \in H^1(\Omega) \, : \, \alpha \leqslant v \leqslant \beta \text{ a.e. in } \Omega, v = u - u_c \text{ on } \partial\Omega \, \}. \tag{3.1}$$

It is shown in [39, 56] that there exists a unique solution to either regularized form of the RPBE in $M_e \subset H^1(\Omega)$. The weak formulation is: Find $u_r \in M_e$ such that

$$a(u_r, v) + b(u_r + u_c, v) = L(v) \quad \forall v \in H^1_0(\Omega) \tag{3.2}$$

where

$$a(u, v) = (\epsilon \nabla u, \nabla v) \tag{3.3}$$

$$b(u, v) = (\bar{\kappa}^2 \sinh(u), v) \tag{3.4}$$

$$L(v) = (-(\epsilon - \epsilon_m)\nabla u_c, \nabla v). \tag{3.5}$$

The linear functional $L(\cdot)$ is defined by integrating the right hand side of Eq. 2.5 by parts and applying the jump condition to eliminate the interface terms.

The weak form for the 3-term split regularized PBE requires solving two problems: first for the harmonic term on Ω_m and second for the split potential on the whole domain Ω. Define the solution space to the harmonic problem as $M_h := \{v \in H^1(\Omega_m) : v(x) = -u_c(x) \ \forall x \in \partial\Omega_m\}$. Then the weak form of Eqns. 2.7-2.11 is: Find $(u_h, u_3) \in M_h \times M_e$ such that

$$a(u_3, v) + b(u_3, v) + a_m(u_h, w) = \langle g(u_h), v \rangle \quad \forall(w, v) \in H^1_0(\Omega_m) \times H^1_0(\Omega) \tag{3.6}$$

where $a_m(\cdot, \cdot)$ is the restriction of the bilinear form to the Ω_m subdomain and

$$\langle g(u_h), v \rangle = \int_{\partial\Omega_m} \epsilon_m \frac{\partial(u_c + u_h)}{\partial n} v dx. \tag{3.7}$$

3.2. **Solving.** Due to the hyperbolic sine, the RPBE has a strong nonlinearity. The discretized nonlinear problems defined in Eqns. 3.2 and 3.6 can be solved using an inexact-Newton method [44]. For brevity, we give details for Eq. 3.2. Define the weak residual functional to be

$$\langle R(u_r^h), v \rangle = L(v) - \left(a(u_r^h, v) + b(u_r^h + u_c^h, v) \right). \tag{3.8}$$

Here u_r^h is the discrete solution satisfying the system of nonlinear equations

$$\langle R(u_r^h), v \rangle = 0 \quad \forall v \in V^h \tag{3.9}$$

where V^h is the space of piecewise linear functions defined by the tetrahedral mesh. Linearizing Eq. 3.9 around u_r^h results in

$$\begin{aligned} Jw^h &:= \langle DR(u_r^h)w^h, v \rangle \\ &= \frac{d}{d\epsilon} \left(\langle R(u_r^h + \epsilon w^h), v \rangle \right|_{\epsilon=0} \\ &= -a(w^h, v) - b'(u_r^h + u_c; w^h, v) \quad \forall v \in V^h. \end{aligned} \tag{3.10}$$

In the linear RPBE, $\sinh(u)$ is replaced by u, and $b'(u, v) = (\bar{\kappa}^2 u, v)$. Newton's method defines the nonlinear update vector s^h as the solution to

$$Js^h = -R(u_r^h). \tag{3.11}$$

Given an initial guess $u_r^h \approx u_0$, the updated solution is defined as $u_1 = u_0 + s_h$. This process can be repeated until a desired level of convergence is achieved. An inexact-Newton method uses an iterative solve to find an approximate solution to Eq. 3.11, with a relatively large tolerance for the linear solve when far from the nonlinear solution. However, as the exact solution to Eq. 3.9 is approached, the linear solver tolerance is tightened so that quadratic convergence is achieved.

The computational complexity of the Newton solver is dominated by the method used to solve the N linear algebraic equations [15, 50] within each iteration. Multilevel methods provide an advantage in that they are provably optimal or nearly optimal methods for solving these systems [13, 50, 80]. The presence of geometrically complex discontinuities in the dielectric ϵ and in the Debye-Hückel parameter $\bar{\kappa}$ in the PBE destroy classical multilevel method efficiency, and can even cause divergence. This is analyzed at length for the PBE in [53, 58, 59], and various techniques based on coefficient averaging and algebraic enforcement of variational (Galerkin) conditions are examined. Algebraic multilevel methods have been used successfully for many similar problems; cf. [16, 17, 30, 36, 35, 31, 71, 76, 77]. A fully unstructured algebraic multilevel approach is taken in FETK, more details are provided in Section 5.

Starting from an initial mesh $\mathcal{T}_0$, the adaptive mesh refinement procedure builds a sequence of conforming meshes $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_l$ [39, 56, 43]. This procedure is divided into four steps: SOLVE, ESTIMATE, MARK, and REFINE. In the SOLVE step, a solution is computed on the current mesh. Using this result, the ESTIMATE step computes elementwise error indicators and an estimate of the global error. In a production environment, the procedure terminates if the global error estimate is below some prescribed tolerance. Otherwise, the MARK step selects elements for refinement. This step is crucial to the convergence of the method. Finally, REFINE subdivides the marked elements possibly subdividing additional unmarked elements in order to produce a conforming mesh. The refinement technique used in this article is longest edge bisection [69].

4. Error Indicators

In this section we present *a posteriori* error indicators for use in adaptive refinement. These estimators are typically developed by considering the residual of the weak form. For example, given a finite element solution $u_r^h \in V^h$, the weak residual for the linear RPBE is (compare to Eq. 3.8)

$$\langle R(u_r^h), v \rangle = L(v) - (\bar{\kappa}^2 u_c, v) - \left(a(u_r^h, v) + (\bar{\kappa}^2 u_r^h, v) \right) \tag{4.1}$$

for a given $v \in V$. For the remainder of this article, we restrict our attention to two classes of error indicators: energy-based and goal-oriented. The first class estimates the error in the energy norm, although this idea can be generalized to other norms, (e.g., the H^1 norm). The second class, called goal-oriented, focuses on estimating the error in a user specified quantity of interest or goal functional. In the following sections, we derive error indicators from both classes for the linear RPBE. For the goal-oriented indicator, the solvation free energy is used as the target functional.

4.1. **Energy Norm Indicators.** A standard *a posteriori* error indicator is based on bounding the error in the energy norm. It is easily derived by breaking the weak residual into its elementwise components and integrating by parts over each element [1]. This technique was used in [40] to derive the following estimator for the linear RPBE

$$\eta_K^2(u^h) = h_K^2 \|r_K\|_{L^2(K)}^2 + \frac{1}{4} h_{\partial K} \|r_{\partial K}\|_{L^2(\partial K)}^2, \tag{4.2}$$

where

$$
\begin{aligned}
r_K(x) =& (\nabla \cdot (\epsilon(x) - \epsilon_m)\nabla u_c(x) - \bar{\kappa}^2(x)u_c(x)) \\
& - (-\nabla \cdot \epsilon(x)\nabla u_r^h(x) + \bar{\kappa}^2(x)u_r^h(x)) \qquad \forall K \in \mathcal{T}
\end{aligned}
\tag{4.3}
$$

and

$$
r_{\partial K}(x) = n_K \cdot \left[\!\left[(\epsilon(x) - \epsilon_m)\nabla u_c(x) + \epsilon(x)\nabla u_r^h(x) \right]\!\right]_{n_K} \qquad \forall K \in \mathcal{T}.
\tag{4.4}
$$

This indicator gives a bound on the error measured in the energy norm ($\|\!|v|\!\|^2 := a(v,v) + (\bar{\kappa}^2 v, v)$)

$$
\|\!|u_r - u_r^h|\!\|^2 \le \sum_{K \in \mathcal{T}} \eta_K^2(u_r^h).
\tag{4.5}
$$

Bounding the error in other norms is possible. For example, in [39, 56] a similar *a posteriori* error indicator for the RPBE was shown to bound the error measured in the H^1 norm. Other efforts have focused on formulating the RPBE as a first-order system least squares (FOSLS) problem, which has a natural error estimate [22, 38].

4.2. **Goal-Oriented Indicators.** Key to the development of goal-oriented error indicators is relating the weak residual to the error in the goal functional. For symmetric linear problems, a direct application of the Riesz representation theorem shows that there exists a dual function that when paired with the weak residual gives the error in the goal [21]. The challenge is to approximate this function and utilize that approximation to develop error indicators. However, for nonlinear problems, like Eq. 2.5, the definition of the dual function is not so clear. The first part of this section discusses a strategy for defining the dual function for both the nonlinear RPBE and the three term splitting. Using this definition of the dual, two goal-oriented error indicators are proposed for the linear RPBE utilizing the solvation free energy as the quantity of interest.

Following [21], a dual function for the nonlinear RPBE can be defined by considering the constrained minimization problem

$$
u_r = \underset{u_r^* \in M_e}{\arg\min}\, S(u_r^*) \quad \text{subject to} \quad a(u_r, v) + b(u_r + u_c, v) = L(v) \ \ \forall v \in H_0^1(\Omega),
\tag{4.6}
$$

where $a(\cdot, \cdot)$, $b(\cdot, \cdot)$ and $L(\cdot)$ are specified in Eq. 3.5, and S is the goal functional. Note that because the RPBE constraint determines the solution uniquely, the minimization problem has the same solution. However, specifying the minimization provides an additional mathematical framework to define the dual function. To see this, consider the Lagrangian associated with the minimization problem

$$
\Theta(u_r, w) = S(u_r) + \big(L(w) - (a(u_r, w) + b(u_r + u_c, w))\big),
\tag{4.7}
$$

where the Lagrange multiplier, $w \in H_0^1(\Omega)$, is also the dual function. Taking the first variation of Θ with respect to u gives the dual problem:

$$
\text{Find } w \in H_0^1(\Omega), \text{ such that } \langle DR(u_r)v, w \rangle = -S(v), \quad \forall v \in H_0^1(\Omega),
\tag{4.8}
$$

where $\langle DR(\cdot)\cdot, \cdot \rangle$ was defined in Eq. 3.10. As discussed above, if $b(\cdot, \cdot)$ is linear in the first argument then Eq. 4.8 simplifies to $a(v, w) + (\bar{\kappa}^2 v, w) = S(v) \ \ \forall v \in H_0^1(\Omega)$. For the linear problem the error in a goal functional $S(\cdot)$, like the solvation free energy, is simply expressed in terms of the weak residual

$$
\begin{aligned}
S(u_r - u_r^h) &= a(u_r - u_r^h, w) + (\bar{\kappa}^2(u_r - u_r^h), w) \\
&= L(w) - a(u_r^h, w) - (\bar{\kappa}^2(u_r^h + u_c), w).
\end{aligned}
\tag{4.9}
$$

Thus, if w is known, the error in $S(\cdot)$ is easily calculated. In the nonlinear case the error in the goal satisfies

$$S(u_r - u_r^h) = L(w) - a(u_r^h, w) - b'(u_r^h + u_c; u_r^h, w) + E, \qquad (4.10)$$

where E is quadratic in the error in u_r^h [21].

For the three term splitting of the PBE, again we setup a constrained minimization problem to define the dual. Using the notation from the weak form in Eq. 3.6, the corresponding Lagrangian is

$$\Theta_{3\text{-term}}(u_3, u_h; w_3, w_h) = S(u_3) + S_m(u_h) + \\ \langle g(u_h), w_3 \rangle - (a(u_3, w_3) + b(u_3, w_3) + a_m(u_h, w_h)) \quad (4.11)$$

where $w_3 \in H_0^1(\Omega)$ and $w_h \in H_0^1(\Omega_m)$ are dual functions. The functional $S_m(\cdot)$ is a restriction of the original goal functional to the Ω_m domain. Taking the first variation of $\Theta_{3\text{-term}}$ with respect to u_3 and u_h gives the dual problem

$$\begin{aligned} a(v, w_3) + b'(u_3; v, w_3) &= S(v) & \forall v \in H_0^1(\Omega), \\ a_m(v, w_h) &= S_m(v) + \langle g'(u_h; v), w_3 \rangle & \forall v \in H_0^1(\Omega_m) \end{aligned} \qquad (4.12)$$

where

$$\langle g'(u_h; v), w_3 \rangle = \int_{\partial \Omega_m} \epsilon_m \frac{\partial v}{\partial n} w_3 dx. \qquad (4.13)$$

4.2.1. *Goal-Oriented Error Indicators for the Linear RPBE.* To make the application of the dual functions in error indicators more concrete, we present two goal-oriented indicators for the linearized RPBE. As an example we will focus on accurate computation of the solvation free energy (see Eq. 2.12). Unfortunately, $S(\cdot)$ is not bounded on $H_0^1(\Omega)$ due to the inclusion of delta distributions. A common approach to circumventing this issue is to use a *mollified* version of the functional [1, 11]. In this case, the mollified solvation free energy is

$$S(u_r) \approx S^\sigma(u_r) = \frac{1}{2} \int u_r(x) \sum_{i=1}^{P} e_c \, q_i \theta(|x - x_i|, \sigma) \, dx, \qquad (4.14)$$

where θ is a locally supported function defined such that

$$\lim_{\sigma \to 0} \int \theta(|x|, \sigma) f(x) \, dx = \int \delta(x) f(x) \, dx = f(0). \qquad (4.15)$$

One possible choice for θ is the step function

$$\theta(r, \sigma) = \begin{cases} B_\sigma^{-1} & r \le \sigma, \\ 0 & r > \sigma, \end{cases} \qquad (4.16)$$

where B_σ is the volume of a ball of radius σ.

A simple error indicator suggested by Eq. 4.9 is to first solve the dual problem using the same approximation space as the primal. This approximate dual could then be substituted for w in Eq. 4.9 to compute the value of the indicator. However, if the same finite dimensional space is used for solving both the dual problem and the primal problem, then, because of Galerkin orthogonality, Eq. 4.9 will be zero (see [11]). A remedy is to instead solve the dual problem using a finer approximation space, $U^h \subset H_0^1(\Omega)$. One convenient choice is to maintain the same mesh and use higher order polynomials for U^h. In the examples below, V^h is the space of piecewise linear polynomials and U^h is

the space of piecewise quadratics. Let the finer resolution solution of the dual problem be denoted $w^{h,2}$. Substituting $w \approx w^{h,2}$ into Eq. 4.9 yields

$$
\begin{aligned}
S^\sigma(u_r - u_r^h) \approx{}& L(w^{h,2} - P_{h,2}^h w^{h,2}) \\
& - \left(a(u_r^h, w^{h,2} - P_{h,2}^h w^{h,2}) + (\bar{\kappa}^2(u_r^h + u_c), w^{h,2} - P_{h,2}^h w^{h,2}) \right) \quad (4.17)
\end{aligned}
$$

Where $P_{h,2}^h$ is a convenient projection (e.g., nodal injection) of the fine space U^h onto V^h. The choice of the projection operator will affect the quality of the indicator. Decomposing the error into its elementwise contributions gives

$$
\begin{aligned}
S^\sigma(u_r - u_r^h) \approx{}& \sum_K L_K(w^{h,2} - P_{h,2}^h w^{h,2}) - a_K(u^h, w^{h,2} - P_{h,2}^h w^{h,2}) \\
& - (\bar{\kappa}^2(u_r^h + u_c), w^{h,2} - P_{h,2}^h w^{h,2})_K \leq \sum_K \eta_K(u^h, w^{h,2}), \quad (4.18)
\end{aligned}
$$

where the subscript K indicates the restrictions of the linear functional, bilinear functional or inner product to element K and

$$
\begin{aligned}
\eta_K(u^h, w^{h,2}) = \int_K \Big| & -(\epsilon(x) - \epsilon_m)\nabla u_c(x) \cdot \nabla(w^{h,2}(x) - w^h(x)) \\
& - \bar{\kappa}^2(x)u_c(x)(w^{h,2}(x) - w^h(x)) \\
& - \epsilon(x)\nabla u_r^h(x) \cdot \nabla(w^{h,2}(x) - w^h(x)) \\
& - \bar{\kappa}^2(x)u_r^h(x)(w^{h,2}(x) - w^h(x)) \Big| \, dx.
\end{aligned} \quad (4.19)
$$

Here, the absolute value of the integrand in Eq. 4.18 has been taken over each element. In the numerical experiments in section 6, Eq. 4.19 is referred to as the the "goal-quadratic" error estimator used by the various adaptive refinement marking strategies.

The error indicator in Eq. 4.19 requires solving a dual problem which is substantially larger than the primal problem. To alleviate this issue, we develop a second goal-oriented error estimator that finds an approximation to the dual in V^h (which is the same space as the primal problem). The error in the goal is estimated by solving many local elementwise boundary value problems. The technique proposed here is similar to the development of the *equilibrated residual method* for computing goal-oriented estimators [66, 67]. However, the less accurate but simpler *element residual method* (ERM), as discussed in [1], is used. Using the parallelogram law, the error in the linear functional can be rewritten [1, 66, 67] as

$$
\begin{aligned}
S^\sigma(u_r - u_r^h) &= a(u_r - u_r^h, w - w^h) + (\bar{\kappa}^2(u_r - u_r^h), w - w^h) \\
&= \frac{1}{4}\left\|\!\left\|(u_r - u_r^h) + (w - w^h)\right\|\!\right\|^2 - \frac{1}{4}\left\|\!\left\|(u_r - u_r^h) - (w - w^h)\right\|\!\right\|^2
\end{aligned} \quad (4.20)
$$

where again $\|\!\|v\|\!\|^2 = a(v,v) + (\bar{\kappa}^2 v, v)$ is the square of the energy norm. Define elementwise error functions $\phi_K = u_r - u_r^h|_K$ and $\psi_K = w - w^h|_K$ to be computed by the element residual method. The error in the primal problem on element K is approximated by the solution to

$$
a_K(\phi_K, v) + (\bar{\kappa}^2 \phi_K, v)_K = \langle R(u_r^h), v\rangle_K + \int_{\partial K} f_K^u v(x)\, ds \quad \forall v \in H^1(K), \quad (4.21)
$$

where

$$
\langle R(u_r^h), v\rangle_K = L_K(v) - a_K(u_r^h, v) - (\bar{\kappa}^2(u_r^h + u_c), v)_K, \quad (4.22)
$$

and

$$f_K^u = \left(\epsilon(x)\nabla u_r^h(x) + (\epsilon(x) - \epsilon_m)\nabla u_c(x)\right) \cdot n_K. \qquad (4.23)$$

Similarly, the error in the dual problem on element K is approximated by the solution to

$$a_K(v, \psi_K) + (\bar{\kappa}^2 v, \psi_K)_K = S_K^\sigma(v) - a_K(v, w^h) - (\bar{\kappa}^2 v, w^h)_K$$
$$+ \int_{\partial_K} f_K^w v(x) \, ds \quad \forall v \in H^1(K), \qquad (4.24)$$

where

$$f_K^w = \epsilon(x)\nabla w^h(x) \cdot n_K. \qquad (4.25)$$

For a derivation of these equations see [40]. Explicitly stated, the "goal-linear" error estimator used in the numerical experiments in section 6 is given by

$$\eta_K(u_r^h, w^h) = \frac{1}{4}\|\phi_K + \psi_K\|_K^2 - \frac{1}{4}\|\phi_K - \psi_K\|_K^2. \qquad (4.26)$$

5. MULTILEVEL PRECONDITIONING

As the mesh is refined, the conditioning of the linear system deteriorates, and preconditioning is necessary to accelerate the convergence of iterative solvers (e.g., the conjugate gradient method). The challenge in designing an efficient preconditioner is balancing the cost of applying the preconditioner with its effectiveness in improving the conditioning of the underlying system.

In a given finite element mesh, at level j, we denote the set of nodes and its cardinality by $\mathcal{N}_j$ and N_j, respectively. We call the set of nodes introduced precisely at level j the *fine* nodes, and denote them $\mathcal{N}_j^f$. As the mesh is refined, $\mathcal{N}_j^f$ is appended to $\mathcal{N}_{j-1}$, leading to the following hierarchy of nodes:

$$\mathcal{N}_j = \mathcal{N}_{j-1} \bigcup \mathcal{N}_j^f, \quad j = 1, \ldots, J,$$

where $N_J = N$.

In the local mesh refinement setting, the way the coarse and fine nodes are processed plays a central role in determining the overall efficiency of a preconditioner. If the computational cost per level can be maintained proportional to $N_j - N_{j-1}$, or slightly larger, then total cost will be order N, and the resulting preconditioner is said to have optimal computational complexity per iteration. If the resulting preconditioned system has a bounded condition number (independent of problem size), a solution can be obtained using an iterative method with a bounded number of iterations. Hence, the combination of optimal per iteration complexity with a bounded condition number leads to a solver with optimal overall complexity.

In this article, we restrict the presentation of local multilevel preconditioning to a purely geometric (node based) perspective because the computational complexity is exactly governed by the number of nodes processed by the preconditioner at each level. The local multilevel preconditioners of interest can be classified into two groups: multiplicative local multigrid (MG) [18, 20, 28, 68] and additive local MG [28, 29]. The additive local MG preconditioner is often called the Bramble-Pasciak-Xu (BPX) preconditioner in the literature. In this article, we report on only the multiplicative variants. We use the term "classical" to refer to the application of a preconditioner in the uniform refinement setting. There is abundant literature on MG preconditioners [33, 75, 80, 81] and local MG (e.g., see [19, 79] for a review).

Proofs demonstrating the optimality of classical MG and classical additive MG preconditioners rely on a geometric increase in the number of nodes per level. This is

because the cost per iteration of these classical preconditioners is proportional to N_j (not $N_j - N_{j-1}$) per level, which results in suboptimal complexity if $\sum_{j=1}^{J} c_j N_j$ is not $\mathcal{O}(N)$. This occurs frequently in the local refinement setting due to the slow increase in the number of nodes between levels.

The family of hierarchical basis (HB) preconditioners, developed by Bank, Dupont, and Yserentant [12, 14, 82], maintain a per-level cost proportional to $N_j - N_{j-1}$, by only processing (smoothing) the fine nodes at each level. Although the cost per iteration is optimal, HB preconditioners do not achieve a uniformly bounded condition number, and suffer from $\mathcal{O}(J^2)$ and $\mathcal{O}(2^J)$ iterations in two- and three-dimensions, respectively. To address this deficiency, we investigate local MG preconditioners which process a larger set of nodes, $\mathcal{X}_j$, but still maintain a cost which is proportional to $N_j - N_{j-1}$ at each level. Hence, we seek a set $\mathcal{X}_j$ such that

$$\mathcal{N}_j^f \subset \mathcal{X}_j \subset \mathcal{N}_j,$$

with cardinality, X_j, which is proportional to $N_j - N_{j-1}$. At the same time, $\mathcal{X}_j$, should be large enough that the resulting system has a bounded condition number, leading to a solver with optimal overall complexity. Aksoylu and Holst [6] showed that this is possible even for three-dimensional local refinement routines.

In the local mesh refinement setting, Aksoylu, Bond, and Holst [2] studied the implementation and algebraic aspects (e.g., matrix representations) of multilevel preconditioners. Subsequent articles provide a comprehensive overview of local MG preconditioners with various emphases: for a theoretical treatment, see [3, 4, 5]; for optimality analysis in three-dimensional local refinement routines, see [6]; for surface mesh applications in computer graphics, see [7].

5.1. **Local multigrid preconditioners.** As mentioned in the previous section, the fundamental difference between classical and local MG preconditioners is the smoothing operation. In classical MG, the smoother acts on all degrees of freedom on every level. In contrast, local MG only smooths a small subset, typically a neighborhood, of the fine degrees of freedom. Pseudo-code for a local MG V-cycle is provided in Algorithm 5.1.

Algorithm 5.1. *Local multigrid V-Cycle:*
$u_{[j]} = Vcycle(u_{[j]}, f_{[j]}, j)$
  0) *If $j = 1$, solve $A_{[j]}u_{[j]} = f_{[j]}$*   *coarsest level solve*
        and return $u_{[j]}$;
  1) $u_{[j]} = Smooth(u_{[j]}, A_{[j]}, f_{[j]}, \mathcal{X}_j, s_1)$;   *smooth s_1 times on $\mathcal{X}_j$*
  2) $r_{[j-1]} = I_{[j]}^{[j-1]} \left(f_{[j]} - A_{[j]}u_{[j]} \right)$;   *restrict residual*
  3) $e_{[j-1]} = 0$;   *set $e_{[j-1]}$ to zero*
  4) $e_{[j-1]} = Vcycle(e_{[j-1]}, r_{[j-1]}, j-1)$;   *coarse-grid recursion*
  5) $u_{[j]} = u_{[j]} + I_{[j-1]}^{[j]}e_{[j-1]}$;   *add interpolated correction*
  6) $u_{[j]} = Smooth(u_{[j]}, A_{[j]}, f_{[j]}, \mathcal{X}_j, s_2)$;   *smooth s_2 times on $\mathcal{X}_j$*
  7) *return $u_{[j]}$*

The set of nodes used by the underlying method's smoothing operation defines the type of preconditioner. In this article, we introduce three different sets of nodes to be used in the local MG preconditioner. On a given refinement level, the *marked region* is the set of elements which have been marked for refinement. The *refinement region* is union of the newly introduced elements as a result of the refinement and closure procedure. Figure 2 depicts a two-dimensional refinement region formed by a local refinement routine which

consists of quadrasection closed by bisection (the so-called *red-green* refinement). The sets of nodes used to define the preconditioners are as follows:
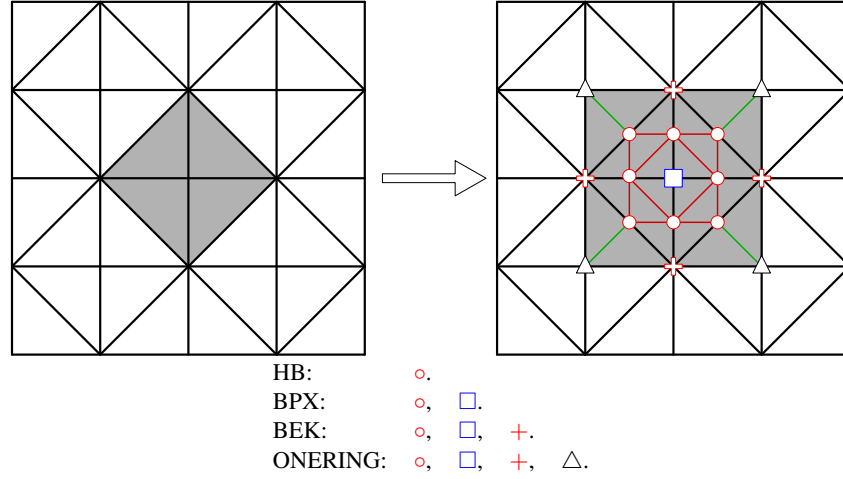

FIGURE 2. An example of red-green refinement (quadrasection-bisection) in two dimensions. On the left, the mesh before refinement, with the marked region shaded in gray. On the right, the mesh after refinement (red edges) and closure (green edges), with the refinement region shaded in gray. The labels indicate which nodes are assigned to each of the preconditioner smoothing sets, $\mathcal{X}_j$.

- $\mathcal{X}_j$-HB: The set of fine nodes on level j, i.e., $\mathcal{N}_j^f$ [12, 14, 82].
- $\mathcal{X}_j$-BPX: The set of nodes whose corresponding basis functions have support entirely contained in the *refinement region* [28].
- $\mathcal{X}_j$-BEK: The set of nodes whose corresponding basis functions have non-empty intersection with the *marked region*. This set is named after the Bornemann-Erdmann-Kornhuber type refinement [23] routine. It consists of fine nodes and their immediate neighboring coarse nodes in the *marked region*. For P1 elements, this set can be inferred from the nonzero pattern of the prolongation operator.
- $\mathcal{X}_j$-ONERING: The set of nodes whose corresponding basis functions have non-empty intersection with the *refinement region* [2, 24, 27, 41]. This set consists of fine nodes and their immediate neighboring coarse nodes in the *refinement region*. For P1 elements, this set can be inferred from the nonzero pattern of the coarse-fine subblock of the stiffness matrix.

As an example, we have labeled the nodes in Figure 2 to show which nodes are in each of the sets described above. The set corresponding to the classical multigrid preconditioner contains all of the nodes on each level.

We should note that the practical implementation of the various local MG preconditioners varies significantly depending on the particular preconditioner. Special care must be taken in order to achieve optimal computational as well as storage complexities. The implementation aspects of how to construct optimal complexity preconditioners are studied in more detail in [2].

6. NUMERICAL EXPERIMENTS

6.1. **Adaptive Refinement.** In this section, the effectiveness of goal-oriented mesh refinement is compared to refinement using the energy-based error indicator. Of interest is
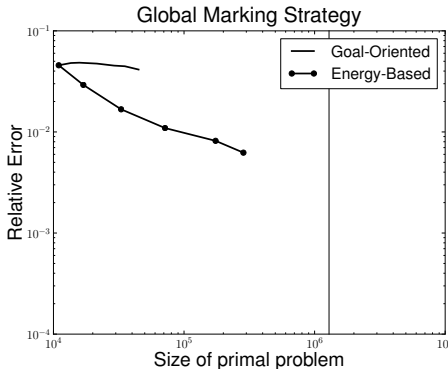
FIGURE 3. Convergence of the solvation free energy for Fasciculin-1 with both goal-oriented and energy-based indicators using the global marking strategy.

computing the solvation free energy of the 921-atom Fasciculin-1 protein [61], using the solution of the linear RPBE. All tests were performed using FETK [54].

In order to solve the RPBE, a definition of the molecular surface and a mesh conforming to that surface is needed. Various definitions of the molecular surface have been proposed in the literature, and the particular value obtained for the solvation free energy will depend strongly on the surface geometry [10]. However, the performance of the algorithms proposed here are insensitive to the choice of molecular surface, as long as it is sufficiently smooth, and the underlying mesh is conforming. Historically, generation of the mesh conforming to the surface was a great impediment to using finite elements for solving the PBE, and only recently, with the development of tools like PDB2PQR and GAMer, has molecular meshing become a routine task. The first step is to prepare the structure using PDB2PQR [42], which adds missing hydrogens, assigns charges, and specifies a radius for each atom in the protein. Next, the resulting PQR file is passed to GAMer [56, 52, 84, 85], which produces a tetrahedral mesh conforming to the shape of the protein. Finally, this mesh is used by FETK to solve the linear RPBE, and compute the solvation free energy.

6.1.1. *Global Marking Strategy.* The adaptive refinement algorithm creates a sequence of grids $\mathcal{T}_0, \mathcal{T}_1, \ldots \mathcal{T}_l \ldots$ based on an error indicator. Critical to this algorithm is the third step MARK. The goal of this step is to select elements for refinement. There are several different choices for marking strategies [1, 11]. The strategy used here is to mark all elements in the lth refinement level that satisfy

$$\eta_K > \gamma \max_{T \in \mathcal{T}_l} \eta_T \quad \forall K \in \mathcal{T}_l, \tag{6.1}$$

where $\gamma \in (0, 1)$. This criteria yields no refinement for $\gamma = 1$ and uniform refinement for $\gamma = 0$.

The convergence of the solvation free energy for Fasciculin-1 using the global marking strategy with both energy-based (Eq. (4.2)) and goal-oriented indicators (Eq. (4.19)) can be seen in Fig. 3. The figure shows the relative error in the solvation free energy as a function of the number of unknowns in the primal problem. The error in the solvation free energy is estimated by computing a high resolution solution to the PBE using uniform mesh refinement. The vertical line near 1.5×10^6 unknowns marks the size of this reference solution. Notice that this figure differs from traditional finite element convergence plots which show error as a function of element radius. Since the resolution of

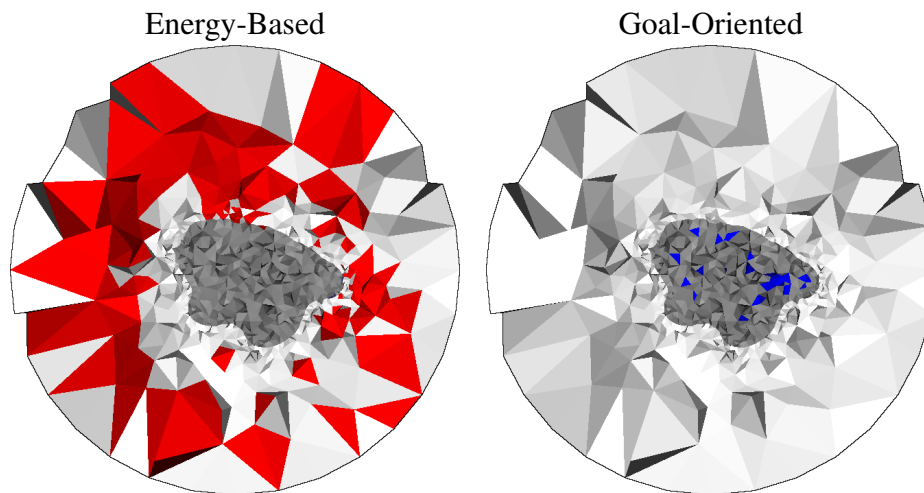Energy-Based   Goal-Oriented

FIGURE 4. A cut-away of the 3D mesh surrounding Fasciculin-1. The colors indicate the distribution of marked elements using the *global* marking strategy with either the energy-based indicator (left) or the goal-oriented indicator (right). Red and blue are marked elements in the solvent and solute subdomains, respectively.

an adaptively refined mesh can greatly vary over the spatial domain, the element radius is not an appropriate measure of the resolution of the mesh. Furthermore, the order of accuracy of the approximation is based on the type of basis functions used, and thus is the same for both uniform and adaptive refinement. The benefit of adaptive refinement is that the mesh can be refined in regions that heavily contribute to the error, resulting in higher accuracy with fewer total degrees of freedom.

In Fig. 3, the lower line shows the convergence of the solvation free energy for several levels of adaptive refinement using the energy-based indicator with the global marking strategy. This scheme makes steady progress to the correct solvation free energy. On the other hand, the upper line shows the results using goal-oriented indicators with the global marking strategy. This scheme performs very poorly.

The reason for the poor performance can be explained by looking at Fig. 4. The images are cut-aways of the 3D mesh, colored to indicate the distribution of marked elements. The left image shows elements selected by the global marking strategy using the energy-based indicator. While the image on the right uses the goal-oriented indicator. The white elements are unmarked elements in the solvent subdomain and the gray elements are unmarked elements in the solute subdomain. Elements colored red are marked solvent elements, while blue elements are marked solute elements. Notice for the energy-based indicator only elements in the solvent subdomain are marked. This indicates that the reaction potential in the solute domain is relatively well approximated compared to the solution in the solvent subdomain. The distribution of the elements marked using a goal-oriented indicator is focused on a few locations in the inner subdomain around the solute atoms. This explains the poor convergence for the goal-oriented indicator in Fig. 3. Since the strategy does not indicate there is error in the solvent domain, no refinement takes place.

To gain further insight into why the global goal-oriented strategy only marks elements in the solute domain, we refer the reader to Fig. 5. This image shows a cut-away with elements colored by their approximate *signed contribution* to the error in the goal. This
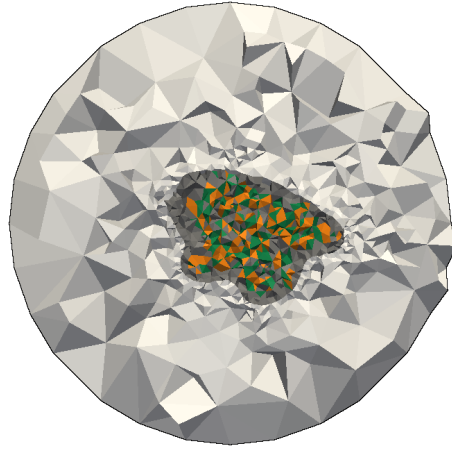
FIGURE 5. A cut-away of the 3D mesh surrounding Fasciculin-1. The colors indicate the positive (green) and negative (orange) estimated elementwise contributions to the error in the solvation free energy (see eq. (4.17)).

is constructed using a quadratic approximation of the dual and the signed elementwise contributions from Eq. (4.17). Note that because these are error contributions and not indicators, they take both positive and negative values. The element contributions range in value between -0.6928 and 1.2779. In the figure, positive element contributions greater than 0.005 are colored green and negative contributions less than -0.005 are colored orange. From the image it is clear that the contributions in the solute subdomain have relatively large magnitude, and they oscillate in sign. The result of this oscillation is that the majority of these contributions cancel when integrated over the entire solute domain. However, the error estimator in Eq. (4.18) uses the absolute value, which results in an overestimation of the error attributed to the solute domain. As a result, the solute domain is over refined, unless steps are taken to modify the marking (or error estimation) strategy.

6.1.2. *Split Marking Strategy.* To improve on the convergence of goal-oriented refinement, a second marking strategy is employed. This is a domain dependent marking strategy that attempts to spread the refinement over solvent and solute regions of the domain. The strategy relies on splitting the mesh into two subsets $\mathcal{T}_l^s \subset \mathcal{T}_l$ and $\mathcal{T}_l^m \subset \mathcal{T}_l$, where $\mathcal{T}_l^s$ and $\mathcal{T}_l^m$ contain the elements in the solvent and solute domains respectively. Stated concisely, the marking strategy is

$$\text{Mark all} \quad \begin{matrix} K \in \mathcal{T}_l^s \\ K \in \mathcal{T}_l^m \end{matrix} \quad \text{such that} \quad \begin{matrix} \eta_K > \gamma \max\limits_{T \in \mathcal{T}_l^s} \eta_T \\ \eta_K > \gamma \max\limits_{T \in \mathcal{T}_l^m} \eta_T \end{matrix} \tag{6.2}$$

where $\gamma \in (0, 1)$.

The split-marking strategy marks a significantly different group of elements, especially for the goal-oriented indicators. The color coding of Fig. 6 is the same as in Fig. 4. Again the energy-based refinement selects elements primarily in the solvent subdomain. However, because split-marking forces refinement in both subdomains, a few elements along the interface in the solute domain are also selected. In contrast, split-marking using a goal-oriented indicator marks a few elements in the solvent subdomain, while also marking elements surrounding the solute atoms.

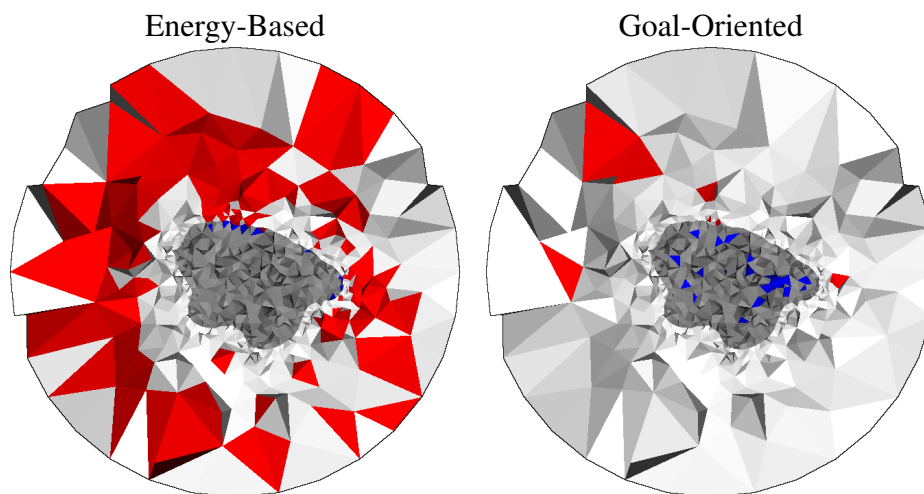Energy-Based   Goal-Oriented

FIGURE 6. A cut-away of the 3D mesh surrounding Fasciculin-1. The colors indicate the distribution of marked elements using the *split* marking strategy with either the energy-based indicator (left) or the goal-oriented indicator (right). Red and blue are marked elements in the solvent and solute subdomains, respectively.
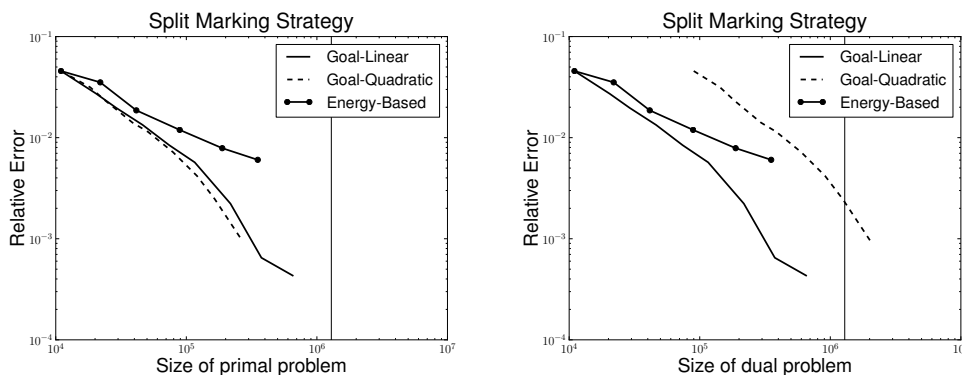

FIGURE 7. Convergence of the solvation free energy for Fasciculin-1 with both goal-oriented and energy-based indicators using the split marking strategy. The convergence measured against the size of the primal/dual problem is on the left/right.

Figure 7 shows the relative error of the solvation free energy in Fasciculin-1 as a function of problem size for several indicators using the split-marking strategy. In the figure, there are two plots. The plot on the left shows the relative error in the solvation free energy as a function of the number of unknowns in the primal problem. The second plot, with the exception of the energy-based refinement strategy, shows the relative error as a function of the size of the dual problem. For energy-based refinement, since no dual is needed, the horizontal axis is the size of the primal problem.

In the figure, the split marking strategy using the energy-based indicator converges steadily. This is similar to the convergence of energy-based refinement using global marking. On the other hand, there is a dramatic improvement in the convergence of both the linear and quadratic goal-oriented refinement techniques, with the rate even increasing slightly as the size of the problem increases. Compare this with the poor results for goal-oriented refinement with the global marking strategy from Fig. 3. The
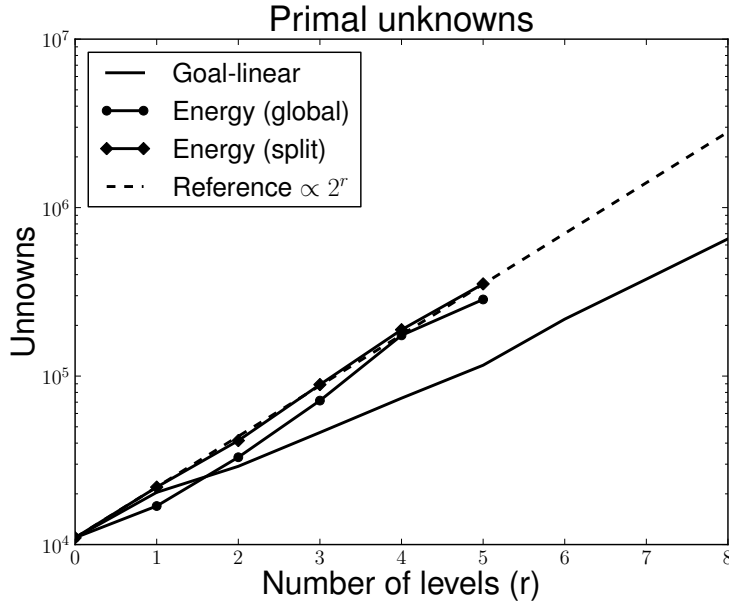
FIGURE 8. Number of primal unknowns as a function of the number of
levels of refinement. The reference line grows proportional to 2^r.

improvement comes from the split marking strategy explicitly taking into account the
error in the solvent and trying to control where it is large.

The second plot in Fig. 7 shows that solving the dual problem using piecewise qua-
dratic elements has substantial additional cost. Although, it is likely that this cost would
be mitigated by the additional work needed when solving the nonlinear RPBE (see
Eq. 2.5). In contrast, the goal-oriented strategy using an indicator constructed from a
linear dual problem (and split marking) does not suffer from the same problem, and is
the most efficient method when the total cost is taken into account.

6.2. **Performance of Preconditioners.** As was discussed in section 5, classical multi-
grid (MG) preconditioners perform best in the uniform refinement setting, where there is
a rapid geometric growth in the number of unknowns as the mesh is refined. In the lo-
cal refinement setting, this growth is much slower, and is frequently subgeometric when
the refinement is concentrated in the neighborhood of a low dimensional feature (e.g., a
point or a line). As a result, the per-iteration complexity of classical MG may fail to scale
linearly (or suffer from a large scaling constant) as the number of unknowns increases. In
contrast, many local MG preconditioners do not have the same restriction, and maintain
optimal per-iteration complexity in both the local and uniform refinement settings.

In Figure 8, the number of primal unknowns is shown as a function of the refinement
level for three different refinement strategies. For the energy-based marking/refinement
strategy, the growth in the number of unknowns is geometric, but subuniform. For goal-
oriented refinement the growth is even slower, but still geometric. The dashed reference
line shows a geometric growth rate proportional to 2^r.

As a measure of the relative locality of each preconditioner, we compute the ratio of
the number of unknowns processed by the smoother, shown in Figure 9. Classical MG is
a global method, so the ratio of smoothed to total unknowns is 1 in both the energy based
and goal oriented refinement cases. As expected, the BEK preconditioner consistently
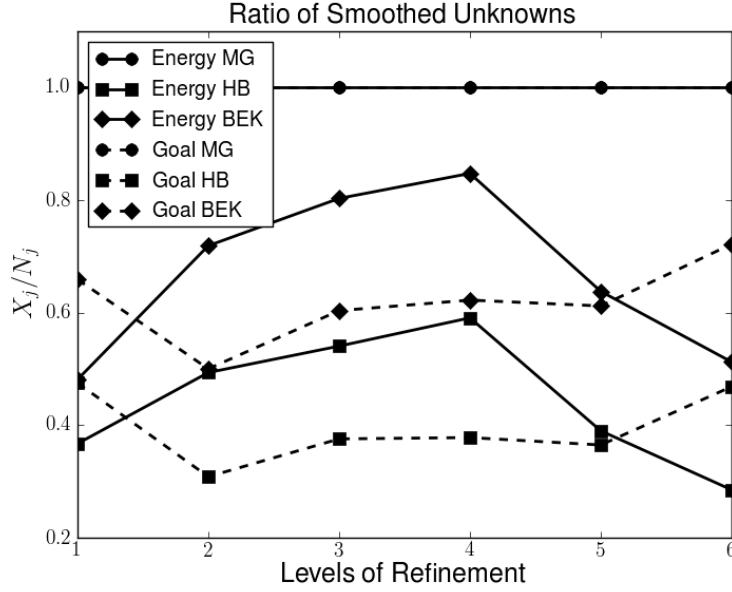
FIGURE 9. Ratio of the number smoothed unknowns, X_j, to total unknowns, N_j, on each level for different multilevel preconditioners
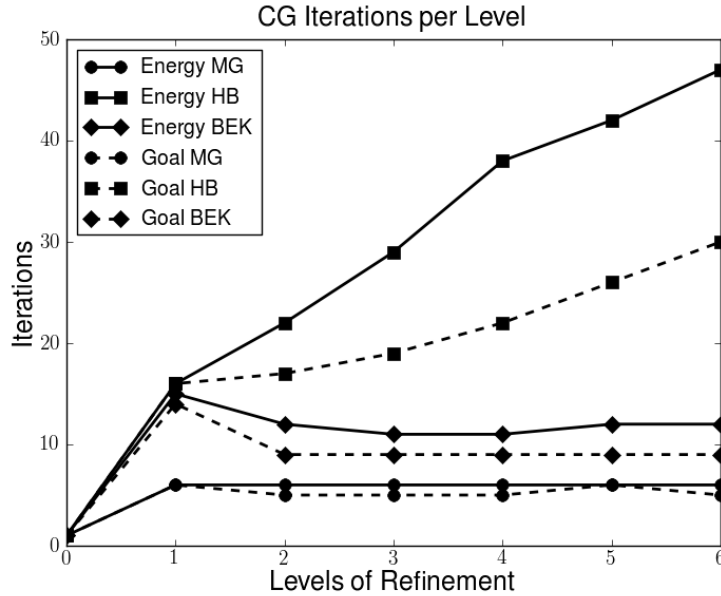

FIGURE 10. Conjugate gradient method iteration counts for the preconditioners used.

smooths more unknowns than HB, but fewer than MG. This is because the set processed by the BEK is a superset of the set processed by HB.

By design, the local multilevel preconditioners considered here maintain an optimal per-iteration complexity in both the local and global refinement settings. The primary challenge for these preconditioners is achieving a bounded condition number, independent of problem size. It can be shown that for a given tolerance, the number of conjugate gradient (CG) iterations can be bounded by a function of the condition number. Hence,

if the condition number is bounded, so is the number of CG iterations. In Figure 10, we report the number of CG iterations as a function of refinement level for MG, BEK, and HB. As predicted by theory, application of the HB preconditioner leads to a slow growth in the number of CG iterations as the mesh is refined regardless of indicator type. In contrast, for both goal-oriented and energy-based refinement the iteration count is bounded for both the classical MG and BEK preconditioners. The iteration counts for BEK are modestly higher than classical MG, but the work per level is reduced, since BEK smooths only a fraction of the unknowns smoothed by classical MG. For this reason, BEK is a compelling alternative to classical MG and HB.

7. CONCLUSION

In this article, we developed goal-oriented error indicators for accurate computation of the solvation free energy from solutions of the regularized Poisson-Boltzmann equation. We found that due to oscillations and imbalanced cancellation in the error contributions, global marking strategies based on goal-oriented error indicators were not viable for driving adaptive mesh refinement. To address this problem, we developed a split marking strategy based on considering each subdomain individually. In numerical experiments, we calculated the solvation free energy for a 921-atom Fasciculin-1 protein. Through these experiments, we showed that the new marking strategy, combined with goal-oriented refinement, is more efficient than energy-based refinement in the context of solvation free energy calculations.

The use of adaptive mesh refinement puts a greater burden on the preconditioner to maintain optimal runtime efficiency. To address this issue, we investigated the use of local multigrid methods, which have a lower per-iteration complexity compared to classical global multigrid. In particular, the BEK variant proved to be a compelling alternative to classical multigrid since it has optimal per-iteration complexity, while still maintaining a bounded iteration count as the mesh is refined. The result is an iterative solver with an optimal overall complexity, scaling linearly with problem size.

8. ACKNOWLEDGMENTS

REFERENCES

[1] M. Ainsworth and J. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. John Wiley & Sons, Inc., New York, 2000.

[2] B. Aksoylu, S. Bond, and M. Holst. An odyssey into local refinement and multilevel preconditioning III: Implementation and numerical experiments. *SIAM J. Sci. Comput.*, 25(2):478–498, 2003.

[3] B. Aksoylu, S. Bond, and M. Holst. Implementation and theoretical aspects of the BPX preconditioner in the three dimensional local mesh refinement setting. Technical report, UT-Austin ICES Report 04-50, 2004.

[4] B. Aksoylu and M. Holst. An odyssey into local refinement and multilevel preconditioning I: Optimality of the BPX preconditioner. Technical report, UT-Austin ICES Report 05-03, 2005.

[5] B. Aksoylu and M. Holst. An odyssey into local refinement and multilevel preconditioning II: Stabilizing hierarchical basis methods. Technical report, UT-Austin ICES Report 05-04, 2005.

[6] B. Aksoylu and M. Holst. Optimality of multilevel preconditioners for local mesh refinement in three dimensions. *SIAM J. Numer. Anal.*, 44(3):1005–1025, 2006.

[7] B. Aksoylu, A. Khodakovsky, and P. Schröder. Multilevel solvers for unstructured surface meshes. *SIAM J. Sci. Comput.*, 26(4):1146–1165, 2005.

[8] N. Baker, M. Holst, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II: refinement at solvent accessible surfaces in biomolecular systems. *J. Comput. Chem.*, 21:1343–1352, 2000.

[9] N. Baker, D. Sept, S. Joseph, M. Holst, and J. A. McCammon. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proc. Natl. Acad. Sci. USA*, 98:10037–10041, 2001.

[10] N. A. Baker, D. Bashford, and D. A. Case. Implicit solvent electrostatics in biomolecular simulation. In B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schutte, and R. Skeel, editors, *New Algorithms for Macromolecular Simulation*, volume 49 of *Lecture Notes in Computational Science and Engineering*, pages 263–295. Springer-Verlag, 2006.

[11] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhauser, Boston, 2003.

[12] R. E. Bank. Hierarchical bases and the finite element method. *Acta Numerica*, 5:1–43, 1996.

[13] R. E. Bank and T. F. Dupont. An optimal order process for solving finite element equations. *Math. Comput.*, 36(153):35–51, 1981.

[14] R. E. Bank, T. F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numer. Math.*, 52:427–458, 1988.

[15] R. E. Bank and D. J. Rose. Analysis of a multilevel iterative method for nonlinear finite element equations. *Math. Comput.*, 39(160):453–465, 1982.

[16] R. E. Bank and J. Xu. The hierarchical basis multigrid method and incomplete LU decomposition. In D. Keyes and J. Xu, editors, *Seventh International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 163–173. AMS, 1994.

[17] R. E. Bank and J. Xu. An algorithm for coarsening unstructured meshes. *Numer. Math.*, 73:1–36, 1996.

[18] P. Bastian. Locally refined solution of unsymmetric and nonlinear problems. In *Proc. of the 8th GAMM Seminar*, volume 46 of *Notes on Numerical Fluid Mechanics*, pages 12–21. Vieweg, 1993.

[19] P. Bastian, W. Hackbusch, and G. Wittum. Additive and multiplicative multi-grid – A comparison. *Computing*, 60:345–364, 1998.

[20] P. Bastian and G. Wittum. On robust and adaptive multigrid methods. In P. Wesseling and P. Hemker, editors, *Proc. of the 4th European Multigrid Conference*. Birkhäuser, 1994.

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

[22] S. D. Bond, J. H. Chaudhry, E. C. Cyr, and L. N. Olson. A first-order systems least-squares finite element method for the Poisson-Boltzmann equation. *J. Comput. Chem.*, 31(8):1625–1635, 2010.

[23] F. Bornemann, B. Erdmann, and R. Kornhuber. Adaptive multilevel methods in three space dimensions. *Internat. J. Numer. Methods Engrg.*, 36:3187–3203, 1993.

[24] F. Bornemann and H. Yserentant. A basic norm equivalence for the theory of multilevel methods. *Numer. Math.*, 64:455–476, 1993.

[25] A. H. Boschitsch and M. O. Fenley. Hybrid boundary element and finite difference method for solving the nonlinear Poisson-Boltzmann equation. *J. Comput. Chem.*, 25(7):935–955, 2004.

[26] D. Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, second edition, 2001.

[27] J. H. Bramble and J. E. Pasciak. New estimates for multilevel algorithms including the V-cycle. *Math. Comput.*, 60(202):447–471, 1993.

[28] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu. Convergence estimates for product iterative methods with applications to domain decomposition. *Math. Comput.*, 57:1–21, 1991.

[29] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55(191):1–22, 1990.

[30] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comp.*, 19:23–56, 1986.

[31] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and its Applications*. Cambridge Univ. Press, 1984.

[32] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, Berlin, second edition, 2002.

[33] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM Books, Philadelphia, second edition, 2000.

[34] B. R. Brooks, C. L. Brooks, III, A. D. Mackerell, Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. CHARMM: The biomolecular simulation program. *J. Comput. Chem.*, 30(10):1545–1614, 2009.

[35] T. F. Chan, S. Go, and L. Zikatanov. Lecture notes on multilevel methods for elliptic problems on unstructured meshes. Technical report, Dept. of Mathematics, UCLA, 1997.

[36] T. F. Chan, B. Smith, and J. Zou. Overlapping Schwarz methods on unstructured meshes using non-matching coarse grids. Technical Report CAM 94-8, Department of Mathematics, UCLA, 1994.

[37] J. H. Chaudhry, S. D. Bond, and L. N. Olson. Finite element approximation to a finite-size modified Poisson-Boltzmann equation. *J. Sci. Comput.*, 47(3):347–364, 2011.

[38] J. H. Chaudhry, S. D. Bond, and L. N. Olson. A weighted adaptive least-squares finite element method for the Poisson-Boltzmann equation. 2011. submitted.

[39] L. Chen, M. Holst, and J. Xu. The finite element approximation of the nonlinear Poisson-Boltzmann Equation. *SIAM J. Numer. Anal.*, 45(6):2298–2320, 2007.

[40] E. C. Cyr. *Numerical Methods for Computing the Free-Energy of Coarse-Grained Molecular Systems*. PhD thesis, University of Illinois at Urbana-Champaign, 2008.

[41] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.*, 63:315–344, 1992.

[42] T. J. Dolinsky, J. E. Nielsen, J. A. McCammon, and N. A. Baker. PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Res.*, 32:W665–W667, 2004.

[43] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.

[44] S. C. Eisenstat and H. F. Walker. Globally convergent inexact newton methods. *SIAM J. Optim.*, 4(2):393–422, 1994.

[45] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Sciences*. Springer, Berlin, first edition, 2004.

[46] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen. A smooth particle mesh Ewald method. *J. Chem. Phys.*, 103(19):8577–8593, 1995.

[47] F. Fogolari, P. Zuccato, G. Esposito, and P. Viglino. Biomolecular electrostatics with the linearized Poisson-Boltzmann equation. *Biophys. J.*, 76(1):1–16, January 1999.

[48] M. K. Gilson, M. E. Davis, B. A. Luty, and J. A. McCammon. Computationn of electrostatic forces on solvated molecules using the Poisson-Boltzmann equation. *J. Phys. Chem.*, 97:3591–3600, 1993.

[49] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.

[50] W. Hackbusch. *Multi-grid Methods and Applications*. Springer-Verlag, Berlin, Germany, 1985.

[51] D. J. Hardy. *Multilevel Summation for the Fast Evaluation of Forces for the Simulation of Biomolecules*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2006.

[52] T. Hayashi, M. E. Martone, Z. Yu, A. Thor, M. Doi, M. Holst, M. H. Ellisman, and M. Hoshijima. Three-dimensional reconstruction reveals new details of membrane systems for calcium signaling in the heart. *J. Cell. Sci.*, 122(7):1005–1013, April 2009.

[53] M. Holst. The Poisson-Boltzmann equation: Analysis and multilevel numerical solution (Monograph based on the Ph.D. Thesis: *Multilevel Methods for the Poisson-Boltzmann Equation*). Technical report, Applied Mathematics and CRPC, California Institute of Technology, 1994.

[54] M. Holst. Adaptive numerical treatment of elliptic systems on manifolds. *Advances in Computational Mathematics*, 15(1–4):139–191, 2001.

[55] M. Holst, N. Baker, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: algorithms and examples. *J. Comput. Chem.*, 21:1319–1342, 2000.

[56] M. Holst, J. A. McCammon, Z. Yu, Y. C. Zhou, and Y. Zhu. Adaptive finite element modeling techniques for the Poisson-Boltzmann equation. *Comm. Comput. Phys.*, 11:179–214, 2012.

[57] M. Holst, G. Nagy, and G. Tsogtgerel. Rough solutions of the Einstein constraints on closed manifolds without near-CMC conditions. *Comm. Math. Phys.*, 288(2):547–613, 2009.

[58] M. Holst and F. Saied. Multigrid solution of the Poisson-Boltzmann equation. *J. Comput. Chem.*, 14(1):105–113, 1993.

[59] M. Holst and F. Saied. Numerical solution of the nonlinear Poisson-Boltzmann equation: Developing more robust and efficient methods. *J. Comput. Chem.*, 16(3):337–364, 1995.

[60] W. Im, D. Beglov, and B. Roux. Continuum solvation model: computation of electrostatic forces from numerical solutions to the Poisson-Boltzmann equation. *Comput. Phys. Comm.*, 111:59–75, 1998.

[61] M. H. le Du, P. Marchot, P. E. Bougis, and J. C. Fontecilla-Camps. 1.9-Å resolution structure of fasciculin 1, an anti-acetylcholinesterase toxin from green mamba snake venom. *J. Biol. Chem.*, 267(31):22122–22130, 1992.

[62] B. Lu, Y. Zhou, M. Holst, and J. A. McCammon. Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications. *Comm. Comp. Phys.*, 3(5):973–1009, 2008.

[63] B. Lu, Y. C. Zhou, G. A. Huber, S. D. Bond, M. J. Holst, and J. A. McCammon. Electrodiffusion: A continuum modeling framework for biomolecular systems with realistic spatiotemporal resolution. *J. Chem. Phys.*, 127(13):135102+ (17 pages), 2007.

[64] J. D. Madura, J. M. Briggs, R. C. Wade, M. E. Davis, B. A. Luty, A. Ilin, J. Antosiewicz, M. K. Gilson, B. Bagheri, L. R. Scott, and J. A. McCammon. Electrostatics and diffusion of molecules in solution: simulations with the University of Houston Brownian Dynamics program. *Comput. Phys. Comm.*, 91(1–3):57–95, 1995.

[65] D. A. McQuarrie. *Statistical Mechanics*. Harper and Row, New York, NY, 1973.

[66] J. T. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Comput. Math. Appl.*, 41:735–756, 2001.

[67] S. Prudhomme and J. T. Oden. On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors. *Comput. Meth. Appl. Mechanics Engrg.*, 176(1-4):313–331, 1999.

[68] M. C. Rivara. Algorithms for refining triangular grids for adaptive and multigrid techniques. *Int. J. Numer. Meth. Eng.*, 20(4):745–756, 1984.

[69] M. C. Rivara. Design and data structure of fully adaptive, multigrid, finite-element software. *ACM Transactions on Mathematical Software*, 10(3):242–264, 1984.

[70] W. Rocchia, S. Sridharan, A. Nicholls, E. Alexov, A. Chiabrera, and B. Honig. Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects. *J. Comput. Chem.*, 23(1):128–137, 2002.

[71] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelphia, PA, 1987.

[72] R. D. Skeel, I. Tezcan, and D. J. Hardy. Multiple grid methods for classical molecular dynamics. *J. Comput. Chem.*, 23(6):673–684, 2002.

[73] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice Hall, 1973.

[74] C. Tanford. *Physical Chemistry of Macromolecules*. John Wiley & Sons, New York, NY, 1961.

[75] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, 2001.

[76] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. Technical Report UCD/CCM 34, Center for Computational Mathematics, University of Colorado at Denver, 1994.

[77] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. Technical Report UCD/CCM 36, Center for Computational Mathematics, University of Colorado at Denver, 1995.

[78] J. A. Wagoner and N. A. Baker. Assessing implicit models for nonpolar mean solvation forces: The importance of dispersion and volume terms. *Proc. Natl. Acad. Sci. USA*, 103(22):8331–8336, 2006.

[79] G. Wittum. Multi-grid methods – An introduction. In W. Hergert, A. Ernst, and M. Däne, editors, *Computational Materials Science*, volume 642 of *Lect. Notes Phys.*, pages 283–311. Springer-Verlag, Berlin, 2004.

[80] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, 1992.

[81] I. Yavneh. Why multigrid methods are so efficient. *Comput. Sci. Engrg.*, 8(6):12–22, 2006.

[82] H. Yserentant. Old and new convergence proofs for multigrid methods. *Acta Numerica*, 2:285–326, 1993.

[83] S. Yu, W. Geng, and G. W. Wei. Treatment of geometric singularities in implicit solvent models. *J. Chem. Phys.*, 126:244108+ (13 pages), 2007.

[84] Z. Yu, M. Holst, Y. Cheng, and J. A. McCammon. Feature-preserving adaptive mesh generation for molecular shape modeling and simulation. *J. Mol. Graph. Model.*, 26:1370–1380, 2008.

[85] Z. Yu, M. Holst, and J. A. McCammon. High-fidelity geometric modeling for biomedical applications. *Finite Elem. Anal. Des.*, 44(11):715–723, 2008.

[86] Z. Zhou, P. Payne, M. Vasquez, N. Kuhn, and M. Levitt. Finite-difference solution of the Poisson-Boltzmann equation: Complete elimination of self-energy. *J. Comput. Chem.*, 17:1344–1351, 1996.

E-mail address: baksoylu@etu.edu.tr

E-mail address: sdbond@sandia.gov

E-mail address: eccyr@sandia.gov

E-mail address: mholst@math.ucsd.edu