BIT Numerical Mathematics (2007) 47: 853–871
DOI: 10.1007/s10543-007-0145-9
Published online: 25 September 2007 – © Springer 2007

A NEW MINIMIZATION PROTOCOL FOR SOLVING NONLINEAR POISSON–BOLTZMANN MORTAR FINITE ELEMENT EQUATION*

DEXUAN XIE[1] and SHUZI ZHOU[2,**]

[1] *Department of Mathematical Sciences, University of Wisconsin, Milwaukee, WI 53211, USA. email: dxie@uwm.edu*

[2] *Department of Applied Mathematics, Hunan University, Changsha, Hunan, P.R. China. email: szzhou@hnu.cn*

Abstract.

The nonlinear Poisson–Boltzmann equation (PBE) is a widely-used implicit solvent model in biomolecular simulations. This paper formulates a new PBE nonlinear algebraic system from a mortar finite element approximation, and proposes a new minimization protocol to solve it efficiently. In particular, the PBE mortar nonlinear algebraic system is proved to have a unique solution, and is equivalent to a unconstrained minimization problem. It is then solved as the unconstrained minimization problem by the subspace trust region Newton method. Numerical results show that the new minimization protocol is more efficient than the traditional merit least squares approach in solving the nonlinear system. At least 80 percent of the total CPU time was saved for a PBE model problem.

AMS subject classification (2000): 65N30, 65H10, 65K10, 92-08.

Key words: Poisson–Boltzmann equation, mortar finite element, nonlinear system, unconstrained minimization, biomolecular simulations.

1 Introduction.

The Poisson–Boltzmann equation (PBE) is one widely-used implicit solvent model for electrostatic interactions of a biomolecular system in an ionic aqueous environment. It has been applied to many important applications in biology and chemistry [2, 4, 18, 24, 25, 31, 33, 34]. In the Poisson–Boltzmann approach, all the solute atoms are described explicitly with point partial charges at atomic positions while the solute and solvent domains are modelled as the continuum

* Received January 4, 2007. Accepted in revised form August 24, 2007. Communicated by Per Lötstedt.

** This work was partially supported by the National Science Foundation, USA, through grant DMS-0241236. The second author was partially supported by NSF of China (grant number 70271019).

regions with low and high dielectric constants, respectively. Several program packages such as DelPhi [30], UHBD [14], and APBS [3] have been developed for solving PBE using different numerical techniques (such as finite difference, and finite element methods) and typical linear and nonlinear iterative solvers (such as the successive over-relaxation method, the conjugate gradient method, the inexact-Newton method, and the multigrid method) [3, 4, 5, 22, 23].

In this paper, we focus on the numerical solution of the PBE nonlinear algebraic system that arises from a mortar finite element discretization. The mortar finite element technique has been shown to be effective to treat the difficulties caused by discontinuous diffusion coefficients, corner point singularities, and singular source terms [1, 6, 26, 35]. In the case of PBE, the domain is naturally decomposed into two subdomains – the solute and solvent domains, and the solute domain is surrounded completely by the solvent domain. The interface between them is the boundary of the solute domain, which may be very irregular since it consists of biomolecular surfaces. Also, PBE contains discontinuous coefficients and singular source terms. Hence, it is of particularly interest to discretize PBE by mortar finite element techniques. A mortar finite element application to the linear PBE problem was analyzed in [11]. In this paper, we intend to develop a new minimization protocol to solve the nonlinear PBE mortar finite element equation efficiently and effectively. Its error estimates will be studied in the future.

Currently, a mortar finite element equation can be formulated in either the constrained mortar space approach [7] or the saddle-point approach based on the unconstrained product space and a Lagrange multiplier space [6]. In this paper, we follow the constrained mortar space approach since it can lead to a nonlinear algebraic system with a positive definite Jacobian matrix, which is required in our new minimization protocol. According to the mortar finite element theory (see [7], for example), we set the solute and solvent regions as the slave and master subdomains, respectively, define the interface finite element space as the restriction of the finite element space of the solute region to the interface, and use a finer mesh size in the solute domain. In this paper, we obtain a mortar finite element equation to the nonlinear PBE in both a variational form and an algebraic form. We then prove that the nonlinear PBE mortar finite element equation has the unique solution and is equivalent to a unconstrained minimization problem. Consequently, the nonlinear PBE mortar finite element equation can be solved naturally as a unconstrained minimization problem.

In order to numerically confirm our theoretical results and demonstrate the performance of the new minimization protocol, we developed a MATLAB program package for solving a PBE model problem. The package was written based on the finite element program PUFFIN [20] and the PDE and optimization toolboxes of MATLAB [27]. While there exist several effective unconstrained optimization algorithms [29], we selected the subspace trust-region Newton method [12, 13]) for solving the PBE unconstrained minimization problem. The subspace trust-region Newton method is a widely-used global convergence

algorithm, which is available in the MATLAB unconstrained minimization solver function *fminunc* as the algorithm for the large scale case.

As comparison, we also solved the nonlinear PBE mortar finite element equation by the MATLAB nonlinear system solver function *fsolve*. In fact, the algorithm in *fsolve* for the large scale case uses the same subspace trust-region Newton method as the one in *fminunc* except that it is applied to solving the merit least squares minimization problem: $\min\{\mathbf{f}(U) \mid U \in R^n\}$, where $\mathbf{f}(U) = \frac{1}{2}\sum_{i=1}^{n}[F_i(U)]^2$ is a widely-used merit function, and the nonlinear system to be solved is in the form $\mathcal{F}(U) = 0$ with $\mathcal{F}(U) = (F_1(U), F_2(U), \ldots, F_n(U))^t$. The merit function $\mathbf{f}(U)$ is also used in another traditional nonlinear solver – the damped inexact Newton method, which was well studied in [22] for solving the PBE nonlinear algebraic system that arises from an adaptive finite element approximation.

The main role of the merit function $\mathbf{f}(U)$ is to make globalization strategies (such as line search and trust region techniques) applicable in solving the nonlinear system such that a numerical solution can be generated from a Newton or inexact Newton method with any initial guess [29]. Without using a globalization strategy, the Newton or inexact Newton method may converge only if an initial guess is "close" enough to the solution. However, since a local minimizer of $\mathbf{f}(U)$ is not necessarily a root of $\mathcal{F}(U)$, some global convergence behaviors of a global convergent Newton or inexact Newton method may be degraded [29]. Moreover, in the merit least squares approach, the gradient vector g_k and Hessian matrix H_k have the forms: $g_k = J_k^t \mathcal{F}(U^{(k)})$ and $H_k = J_k^t J_k$, where J_k denotes the Jacobian matrix of $\mathcal{F}(U)$ at the kth iterate $U^{(k)}$ of the subspace trust-region Newton method. Hence, H_k may become dense even though J_k is sparse. Thus, it becomes expansive to solve the trust region subproblem (see (4.1) for details) by the preconditioned conjugate gradient method (PCG) [19].

In contrast, with our new minimization protocol, we have that $H_k = J_k$ and $g_k = \mathcal{F}(U^{(k)})$; thus, the trust region subproblem can be solved efficiently by PCG, and a numerical minimum point is guaranteed to be a numerical solution of the nonlinear system. Consequently, it can be claimed that the new minimization protocol is more effective and efficient than the traditional merit least squares approach. We confirmed this claim via our MATLAB package for solving the model problem. Numerical results show that the new minimization protocol could save at least 80% of the total CPU time in solving the nonlinear system of PBE compared to the merit least squares approach. We also made numerical experiments using different initial guesses (many of them were selected randomly), confirming that the PBE mortar nonlinear system has a unique solution.

The remainder of the paper is organized as follows. Section 2 introduces and analyzes the PBE mortar finite element approximation and its equivalent functional minimization problem. Section 3 formulates their algebraic forms. Section 4 gives the new minimization protocol. Finally, numerical results are presented in Section 5.

2 PBE mortar finite element approximation.

Let Ω be a bounded domain in R^3 with boundary $\partial\Omega$. It consists of three non-overlapping subdomains Ω_1, Ω_e, and Ω_I satisfying that $\Omega = \Omega_1 \cup \Omega_e \cup \Omega_I$, Ω_1 is surrounded by Ω_e, and Ω_e by Ω_I. We set $\Omega_2 = \Omega_e \bigcup \Omega_I$ and define PBE by

$$\begin{cases} -\nabla \cdot (\epsilon(x)\nabla u) + \kappa(x) \sinh u = f(x) & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where g is a given function, $\epsilon(x)$ and $\kappa(x)$ are the two piecewise constant functions as given below:

$$\epsilon(x) = \begin{cases} \epsilon_1 & \text{for } x \in \Omega_1, \\ \epsilon_2 & \text{for } x \in \Omega_2, \end{cases} \quad \kappa(x) = \begin{cases} 0 & \text{for } x \in \Omega_1 \cup \Omega_e, \\ \bar{\kappa} & \text{for } x \in \Omega_I, \end{cases} \tag{2.2}$$

and the right hand side function $f(x)$ is given by

$$f(x) = \begin{cases} \bar{c} \sum_{i=1}^{n_c} q_i \delta(x - x^i) & \text{for } x \in \Omega_1, \\ 0 & \text{for } x \in \Omega_2. \end{cases} \tag{2.3}$$

Here $x^i \in \Omega_1$ is the position vector of atom i of a biomolecule (e.g., a protein) in Ω_1, q_i is the charge at x^i, n_c is the total number of point charges of the protein, $\bar{c} = 4\pi e_c/(k_B T)$ with e_c as an electron charge constant, k_B Boltzmann's constant, and T the absolute temperature, and $\delta(x)$ denotes the Delta-function, which is defined as 1 at the origin and zero otherwise. According to the Debye–Hückel theory of continuum molecular electrostatics [32], a solution u of PBE gives an electrostatic potential in the region of Ω, $\epsilon(x)$ indicates the permittivity in both the solute region Ω_1 and the solvent region Ω_2, and κ describes the distribution of ions in region Ω. Here Ω_e is the ionic exclusive region while Ω_I the ionic region. For simplicity, we set $g = 0$ in this paper. The case of nonzero g can be easily treated in the way as shown in [9].

Let $v|_S$ denote the restriction of v onto region S, and Γ the interface between Ω_1 and Ω_2. To discretize PBE by a mortar technique, we define two independent conforming finite element spaces, V_{Ω_1} and V_{Ω_2}, based on two independent conforming triangulations $\mathcal{T}_{1,h_1}$ and $\mathcal{T}_{2,h_2}$ of Ω_1 and Ω_2, respectively, where h_i denotes the mesh size of Ω_i for $i = 1$ and 2. The finite element space on the interface, denoted by Λ_{h_1}, is defined as the restriction of V_{Ω_1} to the interface Γ. The mortar condition (or a week matching condition across the interface) is assumed as below:

$$\int_\Gamma (v|_{\Omega_1} - v|_{\Omega_2}) w ds = 0, \quad \forall w \in \Lambda_{h_1}. \tag{2.4}$$

According to the mortar finite element theory (see [7], for example), we set $h_1 < h_2$ and refer to Ω_1 and Ω_2 as the slave and master subdomain, respectively, since the coefficient ϵ_2 on Ω_2 is larger than the coefficient ϵ_1 on Ω_1.

Let V_h be the product function space of V_{Ω_1} and V_{Ω_2}. The constrained mortar finite element space, denoted by $\tilde{V}_h$, is defined as a subspace of V_h of functions that satisfy the mortar condition (2.4). Here $h = max\{h_1, h_2\}$. We then can obtain the mortar finite element equation of PBE as follows:

Find $u_h \in \tilde{V}_h$ such that

$$a(u_h, v) + \int_\Omega \kappa(x) v \sinh u_h dx = \int_\Omega f v dx, \quad \forall v \in \tilde{V}_h, \tag{2.5}$$

where $a(u, v)$ is a symmetric bilinear functional defined by

$$a(u, v) = \int_{\Omega_1} \epsilon_1 \nabla u \cdot \nabla v dx + \int_{\Omega_2} \epsilon_2 \nabla u \cdot \nabla v dx. \tag{2.6}$$

From [7] it can imply that there exists a positive constant α such that

$$a(v, v) \geq \alpha \|v\|^2_{V_h}, \quad \forall v \in \tilde{V}_h, \tag{2.7}$$

where $\|\cdot\|^2_{V_h}$ denotes a norm in V_h. Hence, the corresponding stiffness matrix obtained from a set of basis functions of $\tilde{V}_h$ is symmetric positive definite. In particular, we obtain a new proof to directly prove the following theorem with respect to the case of PBE.

THEOREM 2.1. *Let $a(u, v)$ be the bilinear functional defined in* (2.6). *Then*

$$a(v, v) > 0, \quad \forall v \in \tilde{V}_h \text{ and } v \neq 0.$$

PROOF. Suppose that there exists a nonzero $v^0 \in \tilde{V}_h$ such that $a(v^0, v^0) = 0$. For clarity, we denote by $v^{0,1}$ and $v^{0,2}$ the restrictions of v^0 onto the closures of Ω_1 and Ω_2, respectively. Since $v^0 \in \tilde{V}_h$, we have that $v^0 = 0$ on $\partial\Omega$, and the mortar condition (2.4) holds, i.e.,

$$\int_\Gamma (v^{0,1} - v^{0,2}) w ds = 0, \quad \forall w \in \Lambda_{h_1}. \tag{2.8}$$

Since the two terms of $a(v^0, v^0)$ in (2.6) are nonnegative, they must be zero when $a(v^0, v^0) = 0$. That is,

$$\int_{\Omega_1} \epsilon_1 \nabla v^{0,1} \cdot \nabla v^{0,1} dx = 0, \quad \text{and} \quad \int_{\Omega_2} \epsilon_2 \nabla v^{0,2} \cdot \nabla v^{0,2} dx = 0. \tag{2.9}$$

Since $v^{0,2} \in H^1(\Omega_2)$, and $v^{0,2} = 0$ on $\partial\Omega$, from [9] we know that $v^{0,2}$ satisfies the inequality

$$\int_{\Omega_2} \epsilon_2 \nabla v^{0,2} \cdot \nabla v^{0,2} dx \geq \gamma \|v^{0,2}\|^2_{H^1(\Omega_2)}, \tag{2.10}$$

where γ is a positive constant. Hence, applying the second expression of (2.9) to the above inequality yields $v^{0,2} = 0$.

As a result, the mortar condition (2.8) becomes

$$\int_\Gamma v^{0,1} w ds = 0, \quad \forall w \in \Lambda_{h_1},$$

and from which we get that $v^{0,1} = 0$ on Γ. Note that Γ is exactly the boundary $\partial\Omega_1$ of Ω_1. Hence, $v^{0,1} \in H_0^1(\Omega_1)$. Thus, from [9] we can know that there exists a positive constant, β, such that

$$\int_{\Omega_1} \epsilon_1 \nabla v^{0,1} \cdot \nabla v^{0,1} dx \geq \beta \|v^{0,1}\|^2_{H^1(\Omega_1)}. \tag{2.11}$$

Therefore, applying the first expression of (2.9) to the above inequality immediately yields $v^{0,1} = 0$.

Since both $v^{0,1}$ and $v^{0,2}$ are zero, we get that $v^0 = 0$. This is a contradiction to the assumption that $v^0 \neq 0$. Therefore, $a(v, v)$ must be positive for all nonzero $v \in \tilde{V}_h$. □

In general, for a nonlinear problem given in the form

$$\text{Find } u \in H \text{ such that} \quad (B(u), v) = 0, \quad \forall v \in H,$$

where H is a Hilbert space, it is difficult to find a functional, $\mathcal{J}(v)$, such that its G-derivative $\mathcal{J}'(u)$ satisfies

$$(\mathcal{J}'(u), v) = (B(u), v), \quad \forall v \in H.$$

However, in the case of PBE, we find such a functional and obtain the following theorem.

THEOREM 2.2. *Let $\mathcal{J}(v)$ be a functional defined by*

$$\mathcal{J}(v) = \frac{1}{2} a(v, v) + \int_\Omega \kappa(x) \cosh v dx - \int_\Omega f v dx. \tag{2.12}$$

Then, the minimization problem

$$\text{Find } u \in \tilde{V}_h \text{ such that} \quad \mathcal{J}(u) = \min\{\mathcal{J}(v) \mid v \in \tilde{V}_h\} \tag{2.13}$$

has a unique solution. Moreover, it is equivalent to the PBE mortar finite element equation (2.5).

PROOF. Clearly, $\mathcal{J}(v)$ is Gâteaux differentiable. Its first and second G-derivatives at u, $\mathcal{J}'(u)$ and $\mathcal{J}''(u)$, can be found as below:

$$(\mathcal{J}'(u), v) = a(u, v) + \int_\Omega \kappa(x) v \sinh u dx - \int_\Omega f v dx, \quad \forall v \in \mathbf{V}_h, \tag{2.14}$$

and

$$(\mathcal{J}''(u)v, v) = a(v, v) + \int_\Omega \kappa(x) v^2 \cosh u dx, \quad \forall v \in \mathbf{V}_h. \tag{2.15}$$

From the above expression, together with Theorem 2.1 and the definition of $\kappa(x)$ in (2.2), it is easy to see that $(\mathcal{J}''(u)v, v) > 0$ for all nonzero $v \in \tilde{V}_h$. Thus, $\mathcal{J}(v)$ is a strictly convex functional in $\tilde{V}_h$.

Furthermore, since the delta function $\delta(x - x^i)$ is a linear functional in the sense that $\int_\Omega \delta(x - x^i)v dx = v(x^i)$ for all $v \in V_h$, the function f defined in (2.3) can be regarded as a linear functional of $\tilde{V}_h$, and satisfy that

$$|(f, v)| \le \|f\|_{\tilde{V}_h^*} \|v\|_{\tilde{V}_h}, \quad \forall v \in \tilde{V}_h, \tag{2.16}$$

where $\tilde{V}_h^*$ denotes the dual space of $\tilde{V}_h$. Here $\|f\|_{\tilde{V}_h^*}$ is bounded since $\tilde{V}_h$ is finite dimensional. Hence, a combination of (2.16) and (2.7) with the nonnegativity of $\kappa(x)$ immediately gives that

$$\mathcal{J}(v) \ge a(v, v) - (f, v) \ge \alpha \|v\|_{\tilde{V}_h}^2 - \|f\|_{\tilde{V}_h^*} \|v\|_{\tilde{V}_h}, \quad \forall v \in \tilde{V}_h.$$

From the above inequality it implies that $\mathcal{J}(v) \to \infty$ as $\|v\|_{\tilde{V}_h} \to \infty$. Thus, $\mathcal{J}(v)$ is coercive. Therefore, according to [17, Proposition 1.2], we claim that the unconstrained minimization problem (2.13) has a unique solution .

Finally, the equivalence between (2.5) and (2.13) is followed from (2.14) [15]. □

3 Formulations of nonlinear algebraic problems.

In this section, we formulate the algebraic expressions of the PBE mortar finite element equation (2.5), the minimization problem (2.13), and the mortar condition (2.4). In addition, some details on computing the source and nonlinear terms of the PBE mortar nonlinear algebraic system are discussed.

Let $\{\varphi_j\}_{j=1}^N$ be a set of basis functions of the product space V_h based on the N mesh nodes, x_j for $j = 1, 2, \ldots, N$, of the triangulations $\mathcal{T}_{1,h_1}$ and $\mathcal{T}_{2,h_2}$ (except of the mesh points lying on the boundary $\partial\Omega$). Then, each function u of V_h can be expressed in the form $u = \sum_{j=1}^N u_j \varphi_j$ such that the finite element equation of (2.5) within V_h can be expressed as a system of nonlinear algebraic equations:

$$\sum_{i=1}^N a(\varphi_i, \varphi_j)u_i + s_j(u_1, u_2, \ldots, u_N) = f_j, \quad j = 1, 2, \ldots, N, \tag{3.1}$$

where u_j is a numerical approximation of $u(x_j)$, $s_j(u_1, u_2, \ldots, u_N)$ is a nonlinear function of the unknowns u_1, u_2, …, and u_N, which is defined by

$$s_j(u_1, u_2, \ldots, u_N) = \int_\Omega \kappa(x)\varphi_j \sinh\Big(\sum_{i=1}^N u_i\varphi_i\Big) dx,$$

and $f_j = \int_\Omega f\varphi_j dx$. Clearly, the nonlinear function s_j can be simplified as

$$s_j = \int_{\tau^j} \kappa(x)\varphi_j \sinh\Big(\sum_{i=1}^N u_i\varphi_i\Big) dx,$$

where τ^j denotes the support set of φ_j, and can be evaluated approximately by a numerical quadrature. For example, $s_j \approx \kappa(x_j)|\tau^j|\sinh(u_j)$, where $|\tau^j|$ denotes the size of τ^j. From the definition of delta function we can get the following formula for evaluating f_j:

$$f_j = \begin{cases} \bar{c}\sum_{i=1}^{n_c} q_i \varphi_j(x^i) & \text{if } \varphi_j \in V_{\Omega_1}; \\ 0 & \text{otherwise,} \end{cases}$$

where $j = 1, 2, \ldots, N$, and x^i is the position vector of atom i of the protein.

To write the nonlinear system (3.1) in a 4-block matrix form, we assume that the N mesh nodes are labelled globally in the following ordering: first on the nodes in Ω_1, then on Γ_{Ω_1}, next in $\tilde{\Omega}_2$, and finally on Γ_{Ω_2}. We then construct four subspaces of V_h as follows:

$$\mathcal{V}_{\Omega_1} = Span\{\varphi_j | x_j \in \Omega_{1,h_1}\}, \quad \mathcal{V}_{\Gamma_{\Omega_1}} = Span\{\varphi_j | x_j \in \Gamma_{\Omega_1,h_1}\}, \tag{3.2}$$

$$\mathcal{V}_{\Omega_2} = Span\{\varphi_j | x_j \in \Omega_{2,h_2}\}, \quad \text{and} \quad \mathcal{V}_{\Gamma_{\Omega_2}} = Span\{\varphi_j | x_j \in \Gamma_{\Omega_2,h_2}\}. \tag{3.3}$$

For convenience, we rewrite their basis functions as $\{\varphi_j\}_{j=1}^{n_1}$, $\{\tilde{\varphi}_j\}_{j=1}^{n_2}$, $\{\hat{\varphi}_j\}_{j=1}^{l}$, and $\{\bar{\varphi}_j\}_{j=1}^{m}$, where n_1, n_2, l and m are the dimensions of $\mathcal{V}_{\Omega_1}, \mathcal{V}_{\Omega_2}, \mathcal{V}_{\Gamma_{\Omega_1}}$ and $\mathcal{V}_{\Gamma_{\Omega_2}}$ respectively. Thus, the restrictions of a function v of V_h onto these four subspaces, $v|_{\Omega_1}, v|_{\Omega_2}, v|_{\Gamma_{\Omega_1}}$, and $v|_{\Gamma_{\Omega_2}}$ can be expressed as

$$v|_{\Omega_1} = \sum_{j=1}^{n_1} v_j \varphi_j, v|_{\Omega_2} = \sum_{j=1}^{n_2} \tilde{v}_j \tilde{\varphi}_j, v|_{\Gamma_{\Omega_1}} = \sum_{j=1}^{l} \hat{v}_j \hat{\varphi}_j, v|_{\Gamma_{\Omega_2}} = \sum_{j=m}^{m} \bar{v}_j \bar{\varphi}_j. \tag{3.4}$$

Hence, the corresponding vector V of function v can have the 4-block form

$$V = \begin{pmatrix} v_{\Omega_1} \\ v_{\Gamma_{\Omega_1}} \\ v_{\Omega_2} \\ v_{\Gamma_{\Omega_2}} \end{pmatrix},$$

where

$$v_{\Omega_1} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n_1} \end{pmatrix}, \quad v_{\Gamma_{\Omega_1}} = \begin{pmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \vdots \\ \hat{v}_l \end{pmatrix}, \quad v_{\Omega_2} = \begin{pmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \vdots \\ \tilde{v}_{n_2} \end{pmatrix}, \quad \text{and} \quad v_{\Gamma_{\Omega_2}} = \begin{pmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \vdots \\ \bar{v}_m \end{pmatrix}.$$

Using the above block vector partition, we can express the nonlinear system (3.1) in the 4-block form

$$\mathcal{F}(U) = 0 \quad \text{with } \mathcal{F}(U) = AU + S(U) - F, \tag{3.5}$$

where U, $S(U)$, and F are the three column vectors with u_j, s_j, and f_j as the jth entries, respectively, for $j = 1, 2, \ldots, N$, A is a symmetric matrix of $N \times N$

with the (i,j)th entry being $a(\varphi_i,\varphi_j)$ for $i,j=1,2,\ldots,N$, and their block forms are given by

$$U=\begin{pmatrix} U_{\Omega_1}\\ U_{\Gamma_{\Omega_1}}\\ U_{\Omega_2}\\ U_{\Gamma_{\Omega_2}}\end{pmatrix},\quad F=\begin{pmatrix} f_{\Omega_1}\\ f_{\Gamma_{\Omega_1}}\\ f_{\Omega_2}\\ f_{\Gamma_{\Omega_2}}\end{pmatrix},\quad S(V)=\begin{pmatrix} S_{\Omega_1}\\ S_{\Gamma_{\Omega_1}}\\ S_{\Omega_2}\\ S_{\Gamma_{\Omega_2}}\end{pmatrix},$$

$$A=\begin{pmatrix} A_{\Omega_1} & A_{\Omega_1,\Gamma_{\Omega_1}} & 0 & 0\\ A^t_{\Omega_1,\Gamma_{\Omega_1}} & A_{\Gamma_{\Omega_1}} & 0 & 0\\ 0 & 0 & A_{\Omega_2} & A_{\Omega_2,\Gamma_{\Omega_2}}\\ 0 & 0 & A^t_{\Omega_2,\Gamma_{\Omega_2}} & A_{\Gamma_{\Omega_2}}\end{pmatrix}.$$

The following lemma gives the algebraic form of the mortar condition (2.4).

LEMMA 3.1. *Let $\{\psi_j\}_{j=1}^l$ be a set of basis functions of the interface finite element space Λ_{h_1}. Then, the mortar condition* (2.4) *can be expressed in the matrix form*

$$Mv_{\Gamma_{\Omega_1}}-Wv_{\Gamma_{\Omega_2}}=0, \tag{3.6}$$

where M and W are the two matrices of $l\times l$ and $l\times m$, respectively, with entries m_{ji} and w_{jk} being defined by

$$m_{ij}=\int_\Gamma \psi_i\hat{\varphi}_j ds \quad \text{and} \quad w_{ik}=\int_\Gamma \psi_i\bar{\varphi}_k ds \tag{3.7}$$

for $i,j=1,2,\ldots,l$ and $k=1,2,\ldots,m$. Moreover, M is nonsingular.

PROOF. By using the expressions of $v|_{\Gamma_{\Omega_1}}$ and $v|_{\Gamma_{\Omega_2}}$ in (3.4), the bilinear functional in (2.4) with $v\in V_h$ can be written as

$$\int_\Gamma (v|_{\Gamma_{\Omega_1}}-v|_{\Gamma_{\Omega_2}})w ds=\int_\Gamma \Big(\sum_{i=1}^l \hat{v}_i\hat{\varphi}_i-\sum_{k=l}^m \bar{v}_k\bar{\varphi}_k\Big)w ds,\quad \forall w\in\Lambda_{h_1}.$$

Selecting $w=\psi_j$ and using the expressions of m_{ij} and w_{kj} in (3.7) give that

$$\begin{aligned}\sum_{i=1}^l \hat{v}_i\int_\Gamma \hat{\varphi}_i\psi_j ds-\sum_{k=l}^m \bar{v}_k\int_\Gamma \bar{\varphi}_k\psi_j ds\\ =\sum_{i=1}^l m_{ji}\hat{v}_i-\sum_{k=l}^m w_{jk}\bar{v}_k\end{aligned}$$

for $j=1,2,\ldots,l$. Clearly, the above expressions can be written as the matrix form $Mv_{\Gamma_{\Omega_1}}-Wv_{\Gamma_{\Omega_2}}$. Thus, the mortar condition (2.4) is expressed in the matrix form (3.6).

We next prove the non-singularity of matrix M. Let M_j denote the jth row of M. We consider their linear combination of zero: $\sum_{j=1}^l c_jM_j=0$, which is

equivalent to $\sum_{j=1}^{l} c_j m_{ji} = 0$ for $i = 1, 2, \ldots, l$, or equivalently,

$$\int_\Gamma \hat{\varphi}_i \sum_{j=1}^{l} c_j \psi_j ds = 0 \quad \text{for } i = 1, 2, \ldots, l.$$

Here the definition of w_{ji} in (3.7) has been used. Multiplying $\hat{v}_i$ to the both sides of the above equations and adding them together for all i immediately give

$$\int_\Gamma v \sum_{j=1}^{l} c_j \psi_j ds = 0, \quad \forall v \in V_h,$$

where the fact $v|_{\Gamma_{\Omega_1}} = \sum_{i=l}^{l} \hat{v}_i \hat{\varphi}_i$ has been used. Hence, from the above equation it implies $\sum_{j=1}^{l} c_j \psi_j ds = 0$. Since $\{\psi_j\}_{j=1}^{l}$ is a set of basis functions, all c_j must be zero. This shows that the matrix M has a rank of l. Hence, M is nonsingular. □

Corresponding to Theorem 2.2, we obtain the algebraic forms of (2.5) and (2.13) and their equivalence in the following theorem.

THEOREM 3.2. *Let R^n be the Euclidean vector space with $n = N - l$, and $T = M^{-1}W$ with the matrices M and W being defined in* (3.7)*. Then, the PBE mortar finite element equation* (2.5) *and the corresponding variational problem* (2.13) *can be formulated, respectively, as the nonlinear algebraic system* (3.8) *and the unconstrained minimization problem* (3.9) *as given below:*

$$\tilde{\mathcal{F}}(\tilde{U}) = \mathbf{0} \quad \textit{with} \quad \tilde{\mathcal{F}}(\tilde{U}) = \tilde{A}\tilde{U} + \tilde{S}(\tilde{U}) - \tilde{F} \in R^n, \tag{3.8}$$

and

$$\textit{Find } \tilde{U} \in R^n \textit{ such that } \quad \tilde{J}(\tilde{U}) = \min\{\tilde{J}(\tilde{V}) \mid \tilde{V} \in R^n\}, \tag{3.9}$$

where $\tilde{V} = (\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n)^t$, $\tilde{J}(\tilde{V})$ is defined by

$$\tilde{J}(\tilde{V}) = \frac{1}{2}\tilde{V}^t \tilde{A}\tilde{V} + \int_{\Omega_I} \kappa(x) \cosh\Big(\sum_{x_i \in \Omega_I} \tilde{v}_i \varphi_i\Big) dx - \tilde{F}^t \tilde{V}, \tag{3.10}$$

and $\tilde{U}, \tilde{A}, \tilde{F}$, and $\tilde{S}(\tilde{U})$ are defined by

$$\tilde{U} = \begin{pmatrix} u_{\Omega_1} \\ u_{\Omega_2} \\ u_{\Gamma_{\Omega_2}} \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} A_{\Omega_1} & 0 & A_{\Omega_1,\Gamma_{\Omega_1}} T \\ 0 & A_{\Omega_2} & A_{\Omega_2,\Gamma_{\Omega_2}} \\ T^t A^t_{\Omega_1,\Gamma_{\Omega_1}} & A^t_{\Omega_2,\Gamma_{\Omega_2}} & T^t A_{\Gamma_{\Omega_1}} T + A_{\Gamma_{\Omega_2}} \end{pmatrix}, \tag{3.11}$$

$$\tilde{F} = \begin{pmatrix} f_{\Omega_1} \\ \mathbf{0} \\ T^t f_{\Gamma_{\Omega_1}} + f_{\Gamma_{\Omega_2}} \end{pmatrix}, \quad \textit{and} \quad \tilde{S}(\tilde{U}) = \begin{pmatrix} \mathbf{0} \\ S_{\Omega_2}(\bar{B}\tilde{U}) \\ \mathbf{0} \end{pmatrix}. \tag{3.12}$$

Moreover, both (3.8) *and* (3.9) *have the unique solution and are equivalent.*

PROOF. Since the inverse M^{-1} exists, equation (3.6) can be rewritten as

$$u_{\Gamma_{\Omega_1}} = T u_{\Gamma_{\Omega_2}} \quad \text{with} \quad T = M^{-1}W. \tag{3.13}$$

Thus, we can set the transform matrix $\bar{B}$ such that $U = \bar{B}\tilde{U}$, where

$$U = \begin{pmatrix} u_{\Omega_1} \\ u_{\Gamma_{\Omega_1}} \\ u_{\Omega_2} \\ u_{\Gamma_{\Omega_2}} \end{pmatrix}, \quad \text{and} \quad \bar{B} = \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & T \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}.$$

Multiplying the transpose of $\bar{B}$, $\bar{B}^t$, on the both sides of (3.5) and replacing U by $\bar{B}\tilde{U}$ gives the expression of $\tilde{\mathcal{F}}(\tilde{U})$ with $\tilde{A} = \bar{B}^t A \bar{B}$, $\tilde{S}(\tilde{U}) = \bar{B}^t S(\bar{B}\tilde{U})$, and $\tilde{F} = \bar{B}^t F$. By matrix-vector operations, we then immediately obtain the block form of $\tilde{A}$ in (3.11), and the block forms of $\tilde{F}$ and $\tilde{S}(\tilde{U})$ as below:

$$\tilde{F} = \begin{pmatrix} f_{\Omega_1} \\ f_{\Omega_2} \\ T^t f_{\Gamma_{\Omega_1}} + f_{\Gamma_{\Omega_2}} \end{pmatrix}, \quad \text{and} \quad \tilde{S}(\tilde{U}) = \begin{pmatrix} S_{\Omega_1}(\bar{B}\tilde{U}) \\ S_{\Omega_2}(\bar{B}\tilde{U}) \\ T^t S_{\Gamma_{\Omega_1}}(\bar{B}\tilde{U}) + S_{\Gamma_{\Omega_2}}(\bar{B}\tilde{U}) \end{pmatrix}.$$

The above forms are then simplified to the forms of (3.12) since from the definitions of $\kappa(x)$ and $f(x)$ in (2.2) and (2.3) it is easy to see that $f_{\Omega_2} = 0$ and $S_{\Omega_1}(U) = S_{\Gamma_{\Omega_1}}(U) = S_{\Gamma_{\Omega_2}}(U) = 0$.

Similarly, we can obtain (3.9). Clearly, $\tilde{J}(\tilde{V})$ is a twice continuously differentiable function of n real variables $\tilde{v}_j$ for $j = 1, 2, \ldots, n$. Its gradient vector $\nabla\tilde{J}$ and Hessian matrix $\nabla^2\tilde{J}$ can be easily found as below:

$$\nabla\tilde{J}(\tilde{V}) = \tilde{\mathcal{F}}(\tilde{V}), \quad \text{and} \quad \nabla^2\tilde{J}(\tilde{V}) = \nabla\tilde{\mathcal{F}}(\tilde{V}), \tag{3.14}$$

where $\tilde{\mathcal{F}}(\tilde{V})$ is given in (3.8), and $\nabla\tilde{\mathcal{F}}(\tilde{V})$ is the Jacobian matrix of $\tilde{\mathcal{F}}(\tilde{V})$. Hence, from the above identities it follows the equivalence between (3.8) and (3.9). □

The formulation of the PBE mortar nonlinear system (3.8) depends on the transform matrix T of (3.13). To reduce the computing cost, the basis functions can be selected based on the dual Lagrange multiplier approach [35] to satisfy the bi-orthogonality relation $m_{ii} = \int_\Gamma \hat{\varphi}_i ds$ and $m_{ij} = 0$ if $i \neq j$, where $i, j = 1, 2, \ldots, l$. In this way, T becomes a sparse scaled mass matrix.

The Jacobian matrix $\nabla\tilde{\mathcal{F}}(\tilde{V})$ can be found as below:

$$\nabla\tilde{\mathcal{F}}(\tilde{V}) = \tilde{A} + \nabla\tilde{S}(\tilde{V}), \tag{3.15}$$

where $\nabla\tilde{S}(\tilde{V})$ is a symmetric matrix of $N \times N$ with entry s_{ij} being defined by

$$s_{ij} = \int_{\Omega_I} \kappa(x)\varphi_i\varphi_j \cosh\Big(\sum_{x_i \in \Omega_I} \tilde{v}_k\varphi_k\Big) dx \quad \text{for } i, j = 1, 2, \ldots, N. \tag{3.16}$$

From the definition of $\kappa(x)$ in (2.2) we see that the above s_{ij} is nonzero only if the corresponding mesh nodes x_i and x_j of basis functions φ_i and φ_j are located in the ionic region Ω_I. In particular, if s_j is approximated by the numerical quadrature, $s_j \approx \kappa(x_j)|\tau^j|\sinh(\tilde{v}_j)$, $\nabla\tilde{S}(\tilde{V})$ is reduced to the diagonal matrix with the nonzero diagonal entries $s_{jj} = \kappa(x_j)|\tau^j|\cosh(\tilde{v}_j)$ for $j = 1, 2, \ldots, n$.

4 New minimization protocol for solving PBE.

Based on Theorem 3.2, we propose to solve the nonlinear system (3.8) through solving the unconstrained minimization problem (3.9) by a widely-used global convergent optimization algorithm, the subspace trust-region Newton method [8, 10, 12, 13, 28]. By this new minimization protocol, a numerical solution of (3.8) can be obtained efficiently and effectively with any initial guess.

The subspace trust region Newton iterates $\{\tilde{U}^{(k)}\}$ for solving (3.9) is defined in the form

$$\tilde{U}^{(k+1)} = \tilde{U}^{(k)} + \mathbf{s}^k, \quad k = 0, 1, 2, \ldots,$$

where $\tilde{U}^{(0)}$ is a given initial guess, and $\mathbf{s}^k$ denotes the kth trial step vector, which satisfies that $\tilde{J}(\tilde{U}^{(k)} + \mathbf{s}^k) < \tilde{J}(\tilde{U}^{(k)})$, and is selected by PCG for solving the trust region subproblem:

$$\min\left\{\frac{1}{2}\mathbf{s}^t H_k \mathbf{s} + \mathbf{s}^t \mathbf{g}_k + \mathbf{f}_k \mid \mathbf{s} \in \mathcal{N}\right\}, \tag{4.1}$$

where $\mathbf{f}_k = \tilde{J}(\tilde{U}^{(k)})$, $\mathbf{g}_k = \nabla\tilde{J}(\tilde{U}^{(k)})$, $H_k = \nabla^2\tilde{J}(\tilde{U}^{(k)})$, and $\mathcal{N}$ is a trust region. In the subspace trust region Newton method, $\mathcal{N}$ is defined as below:

$$\mathcal{N} = \{\mathbf{s} \in \mathit{Span}\{\mathbf{g}_k, \mathbf{u}_k\} \mid \|\mathbf{s}\| \le \delta_k\},$$

where $\delta_k > 0$ is the trust region radius, $\|\cdot\|$ denotes the Euclidean norm, and $\mathbf{u}_k$ is the jth iterate $s^{(j)}$ of PCG if it is well defined and satisfies the termination rule $\|\mathbf{g}_k + H_k s^{(j)}\| < \mathit{TolPCG}\|\mathbf{g}_k\|$; otherwise, $\mathbf{u}_k$ is a negative curvature direction s produced from PCG that satisfies $s^t H_k s < 0$. Here *TolPCG* is a given tolerance for controlling the accuracy of PCG iterates. Clearly, the dominant work is to determine the two-dimensional subspace $\mathit{Span}\{\mathbf{g}_k, \mathbf{u}_k\}$, which is spanned linearly by two vectors $\mathbf{g}_k$ and $\mathbf{u}_k$. Once this subspace is computed, solving (4.1) becomes trivial. See [8, 10] for the details of the subspace trust region Newton method.

We denote by J_k as the Jacobian matrix $\nabla\tilde{\mathcal{F}}(\tilde{U}^{(k)})$. Because of (3.14), we see that $H_k = J_k$ and $\mathbf{g}_k = \tilde{\mathcal{F}}(\tilde{U}^{(k)})$. Thus, the trust region subproblem (4.1) can be solved efficiently by PCG since J_k is a sparse symmetric positive definite matrix. Due to that $\mathbf{g}_k = \tilde{\mathcal{F}}(\tilde{U}^{(k)})$, a numerical minimizer of $\tilde{J}(\tilde{V})$ satisfying that $\|\mathbf{g}_k\| < \epsilon$ (e.g., $\epsilon = 10^{-6}$) is also a numerical solution of the nonlinear algebraic system (3.8) with the error $\|\tilde{\mathcal{F}}(\tilde{U}^{(k)})\|$ less than ϵ.

As comparison, we also consider the traditional merit least squares approach: Find a numerical solution of the nonlinear system (3.8) through solving the

following merit least squares minimization problem:

$$\text{Find } U \in R^n \text{ such that} \quad \mathbf{f}(U) = \min\{\mathbf{f}(V) | V \in R^n\}, \tag{4.2}$$

where $\mathbf{f}(V) = \frac{1}{2}\|\tilde{\mathcal{F}}(V)\|_2^2$ with the vector function $\tilde{\mathcal{F}}$ being given in (3.8). When the subspace trust region Newton method is used to solve (4.2), the trust region subproblem (4.1) is defined with $\mathbf{f}_k = \frac{1}{2}[\tilde{\mathcal{F}}(\tilde{U}^{(k)})]^t\tilde{\mathcal{F}}(\tilde{U}^{(k)})$, $\mathbf{g}_k = J_k^t\tilde{\mathcal{F}}(\tilde{U}^{(k)})$, and $H_k = J_k^t J_k$. Clearly, H_k may become dense even though J_k is sparse, and may have a larger condition number than J_k. Thus, the cost for solving (4.1) by PCG may increase significantly. Hence, our new minimization protocol is expected to be more effective and efficient than the traditional merit least squares approach in solving the PBE nonlinear algebraic system (3.8).

5 Numerical results.

To illustrate our minimization protocol, we developed a MATLAB program package for solving a two-dimensional (2D) model problem of PBE with domains $\Omega = (0,1)\times(0,1)$, $\Omega_1 = (1/4, 3/4)\times(1/4, 3/4)$, and $\Omega_2 = \Omega - \Omega_1$. The values of $\epsilon(x)$ and $\kappa(x)$ were set as $\epsilon_1 = 2$, $\epsilon_2 = 78$, and $\bar{\kappa} = 60$. The n charge positions $\{x^i | i = 1, 2, \ldots, n\}$ of the right hand side function (2.3) were set as the centers of the n triangles selected randomly from the triangulation $\mathcal{T}_{1,h_1}$. The charge q_i at position x^i was set as a random value between -1 and 1 to mimic an atomic charge of a protein. In addition, we set the electron charge constant $e_c = 1$ and the temperature $T = 298K$.

The MATLAB program package was written based on the finite element MATLAB program PUFFIN [20] and the PDE toolbox of MATLAB [27]. The PBE mortar nonlinear system (3.8) and its corresponding unconstrained minimization problem (3.9) were formulated based on two independent linear finite element spaces V_{Ω_1} and V_{Ω_2}. The interface finite element space Λ_{h_1} was set as a linear finite element space inherited from V_{Ω_1}. For test purpose, all the involved matrices including the Hessian matrix and the Jacobian matrix were simply stored in 2D matrix arrays.

Note that the subspace trust region Newton method described in the previous section has been implemented in the MATLAB library, and used in the MATLAB unconstrained minimization solver function *fminunc* for solving a large scale unconstrained minimization problem and the MATLAB nonlinear solver function *fsolve* for solving a large scale nonlinear system based on the merit least square approach. Hence, we simply called *fminunc* and *fsolve* to solve the minimization problem (3.9) and the nonlinear system (3.8), respectively, in the package.

In the tests, we provided both *fminunc* and *fsolve* with the same calculation routines on $\tilde{\mathcal{F}}(\tilde{U}^{(k)})$ and $\nabla\tilde{\mathcal{F}}(\tilde{U}^{(k)})$. For simplicity, we used the same parameters for all the tests, which included the parameter options of *fsolve* and *fminunc* (e.g., diagonal preconditioning), the termination tolerance on unknowns $TolX = 10^{-8}$, the termination tolerance on the function value $TolFun = 10^{-12}$, and the termination tolerance on the PCG inner loop $TolPCG = 10^{-3}$. To test the effect of an

accuracy of the PCG solution on the performance of the subspace trust region Newton method, some tests were repeated with $TolPCG = 10^{-6}$.

The tests with $h_1 = 1/32$, $h_2 = 1/16$, and the total charge number $n_c = 470$ were done on a PC computer (Dell Latitude D610 with 1.86 GHz Pentium 4 and 1 GB of RAM) at the University of Wisconsin-Milwaukee. The results are reported in Figures 5.1 to 5.8 and Tables 5.1 and 5.2.

Figure 5.1 displays the mortar triangulation mesh of domain Ω generated by our package. The solution of the corresponding PBE model problem is visualized in Figure 5.2. From the figure we see that the electrostatic potential energy in Ω_1 dominated over the whole region Ω. Although the potential in the solvent region Ω_2 is relatively small, it does nonlinearly change toward the interface Γ as shown in Figure 5.3.

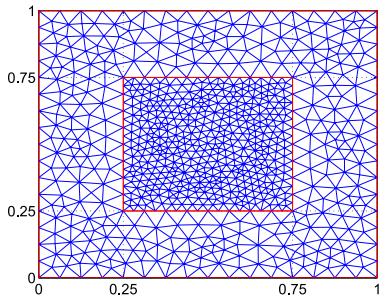

Figure 5.1: The mortar triangular mesh of domain Ω with grid sizes $h_1 = 1/32$ and $h_2 = 1/16$.

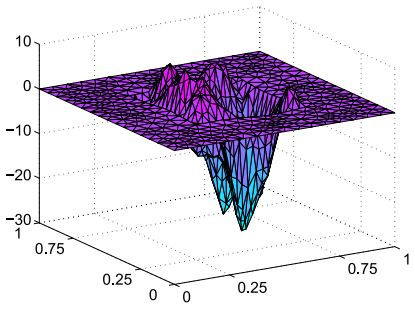

Figure 5.2: The numerical solution of the PBE model problem.

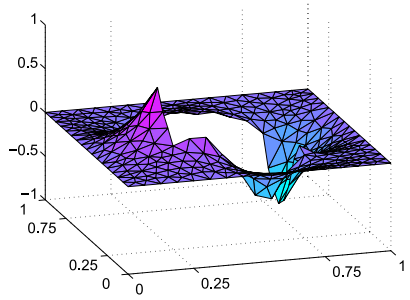

Figure 5.3: The part of the PBE numerical solution in the solvent region Ω_2.

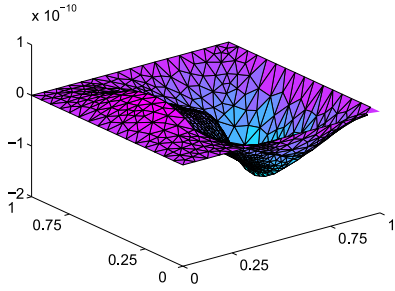

Figure 5.4: The errors between the two numerical solutions produced by solving (3.9) and (4.2) with $\tilde{U}^{(0)} = \tilde{A}^{-1}\tilde{F}$.

Tables 5.1 and 5.2 compare the performance of the subspace trust region Newton method for solving the PBE minimization problem (3.9) with that for solving the merit least squares problem (4.2). From the tables we see that our PBE minimization protocol has significantly speeded up the searching process of a PBE

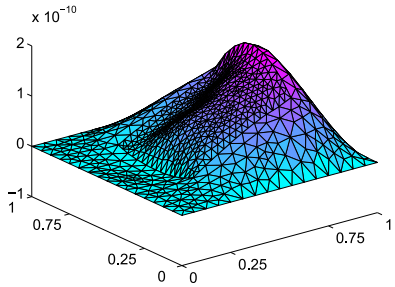

Figure 5.5: The errors between the two numerical solutions produced by solving (3.9) with $\tilde{U}^{(0)} = 0$ and $\tilde{U}^{(0)} = \tilde{A}^{-1}\tilde{F}$.

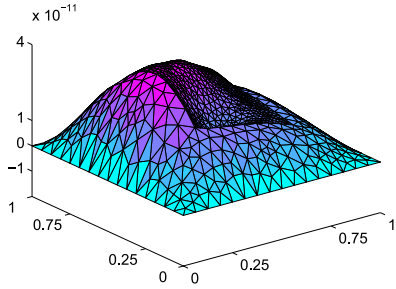

Figure 5.6: The errors between the two numerical solutions produced by solving (3.9) with $\tilde{U}^{(0)} = \tilde{A}^{-1}\tilde{F}$ and a randomly selected initial guess.

Table 5.1: Performance of the subspace trust region Newton method for solving the PBE minimization problem (3.9). Here *TolPCG* is the PCG termination tolerance, ITE denotes the total number of iterations, $\mathbf{s}$ is a trial step vector, and #PCG denotes the total number of PCG iterations.

With the initial guess $\tilde{U}^{(0)} = \tilde{A}^{-1}\tilde{F}$ (CPU in seconds)						
TolPCG	ITE	$\tilde{J}$	$\|\nabla\tilde{J}\|$	$\|\mathbf{s}\|$	#PCG	CPU
10^{-6}	6	-7173.55	7.25×10^{-11}	1.6×10^{-9}	393	5.39
10^{-3}	6	-7173.55	7.25×10^{-11}	1.6×10^{-9}	245	4.82
With the initial guess $\tilde{U}^{(0)} = 0$ (CPU in minutes)						
TolPCG	ITE	$\tilde{J}$	$\|\nabla\tilde{J}\|$	$\|\mathbf{s}\|$	#PCG	CPU
10^{-6}	224	-7173.55	1.16×10^{-3}	9.66×10^{-9}	12537	2.79
10^{-3}	224	-7173.55	6.80×10^{-4}	6.14×10^{-9}	7560	2.54

Table 5.2: Performance of the subspace trust region Newton method for solving the merit least squares problem (4.2). Here $\mathbf{f}$ is the merit function and $\mathbf{g}$ its gradient vector.

With the initial guess $\tilde{U}^{(0)} = \tilde{A}^{-1}\tilde{F}$ (CPU in seconds)						
TolPCG	ITE	$\mathbf{f}$	$\|\mathbf{g}\|$	$\|\mathbf{s}\|$	#PCG	CPU
10^{-6}	5	8.6×10^{-17}	2.7×10^{-7}	5.3×10^{-8}	5716	36.70
10^{-3}	5	8.5×10^{-17}	2.08×10^{-6}	5.3×10^{-8}	3434	23.62
With the initial guess $\tilde{U}^{(0)} = 0$ (CPU in minutes)						
TolPCG	ITE	$\mathbf{f}$	$\|\mathbf{g}\|$	$\|\mathbf{s}\|$	#PCG	CPU
10^{-6}	215	5.22×10^{-18}	7.47×10^{-8}	1.43×10^{-8}	199488	20.88
10^{-3}	216	5.22×10^{-18}	1.57×10^{-7}	1.67×10^{-8}	79115	10.14

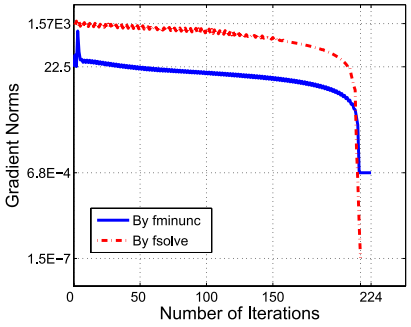

Figure 5.7: Comparison of the gradient norms of the subspace trust region Newton method for solving the PBE minimization problems (3.9) with that for the merit least squares problem (4.2). Here $\tilde{U}^{(0)} = 0$.

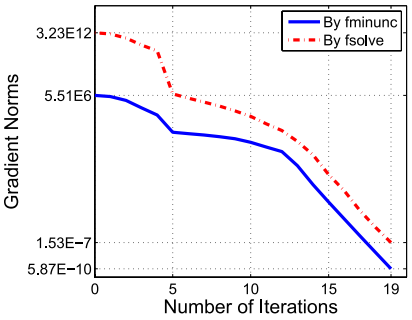

Figure 5.8: Comparison of the gradient norms of the subspace trust region Newton method for solving (3.9) with that for (4.2). Here $\tilde{U}^{(0)}$ is a randomly selected initial guess.

solution: it took only at most 20% of the total CPU time spent by solving the traditional merit least squares problem. Most of the CPU time savings came from the inner PCG loop for solving the trust region subproblem (4.1). From the tables we also see that using $\tilde{U}^{(0)} = \tilde{A}^{-1}\tilde{F}$ is a good choice, with which 96% of the total CPU time was saved compared the case with an initial guess of zero.

Figures 5.7 and 5.8 compare the global convergence behaviors (in terms of gradient norms) of the subspace trust region Newton methods for solving (3.9) and (4.2). Here the initial guess was respectively set as zero and a randomly selected initial guess. Each component of the randomly selected initial guess was a random number between 0 and 10. The starting and ending values of the gradient norm are marked out in the figures for easy to compare. From the figures we see that the subspace trust region Newton method has a superlinear convergence rate at the last few iterations. We noted that the gradient norm was not small enough in the case of solving (3.9) with the initial guess of zero due to the iteration satisfying the termination rule: $\|\tilde{U}^{(k)} - \tilde{U}^{(k-1)}\| = \|\mathbf{s}^{k-1}\| < TolX$, where we set $TolX = 10^{-8}$ for all the tests. This rule is one of the practical termination rules in *fminunc* since it guarantees the solution $\tilde{U}^{(k)}$ to have an enough accuracy.

To confirm that the PBE problem (3.8) has the unique solution, we made many tests using different initial guesses (most of them selected randomly). It was found that the differences between any two computed solutions were less than $O(10^{-10})$. As demonstrations, three of them were displayed in Figures 5.4–5.6. These test results confirm what is claimed in Theorem 3.2.

Finally, we made tests with $h_1 = 1/64$, $h_2 = 1/32$, and $n_c = 1500$ to demonstrate the performance of our new minimization protocol on a finer mesh. In this case, the nonlinear algebraic system consisted of 3042 equations and 3042 unknowns, which resulted in a 3042×3042 Jacobian matrix. To speed up the calculations, we made these tests on a powerful Linux workstation (a Dell Preci-

Table 5.3: Comparison of the performance of the new minimization protocol (*fminunc*) with that of the traditional least squares approach (*fsolve*). Here the model problem was set with $h_1 = 1/64$, $h_2 = 1/32$, $n_c = 1500$, and $n = 3042$.

Solve PBE minimization problem (3.9) by *fminunc* (CPU in minutes)						
$\tilde{U}^{(0)}$	ITE	$\tilde{J}$	$\|\nabla\tilde{J}\|$	$\|\mathbf{s}\|$	#PCG	CPU
$\tilde{A}^{-1}\tilde{F}$	7	-24930.3	1.09×10^{-8}	2.39×10^{-10}	601	0.713
0	618	-24930.3	3.36×10^{-9}	1.53×10^{-3}	9365	53.18
Solve PBE nonlinear system (3.8) by *fsolve*						
$\tilde{U}^{(0)}$	ITE	$\mathbf{f}$	$\|\mathbf{g}\|$	$\|\mathbf{s}\|$	#PCG	CPU
$\tilde{A}^{-1}\tilde{F}$	6	2.76×10^{-17}	2.10×10^{-7}	4.90×10^{-7}	41026	9.93
0	618	1.61×10^{-15}	6.27×10^{-7}	1.12×10^{-6}	396180	469

sion 490 with dual core Intel Xeon processor 2GHz and 4GB main memory) . The results were given in Table 5.3. From it we see that using our new minimization protocol saved more CPU time for a larger nonlinear system. For example, in the case of using the initial guess $\tilde{U}^{(0)} = \tilde{A}^{-1}\tilde{F}$, our new minimization protocol speeded up by 14 times the performance of the traditional merit lease-squares approach. In the future, we plan to develop a new program package with sparse matrix techniques to further improve the performance of our PBE minimization protocol in the large scale case.

Acknowledgments.

We are indebted to the two anonymous referees and the reviewing editor for valuable comments and helpful suggestions.

REFERENCES

1. Y. Achdou, Y. Maday, and O. Widlund, *Iterative substructuring preconditioners for mortar element methods in two dimensions*, SIAM J. Numer. Anal., 36 (1999), pp. 551–580.
2. N. A. Baker, *Improving implicit solvent simulations: a Poisson-centric view*, Curr. Opin. Struc. Biol., 15 (2005), pp. 137–143.
3. N. Baker, D. Sept, M. Holst, and J. A. McCammon, *The adaptive multilevel finite element solution of the Poisson–Boltzmann equation on massively parallel computers*, IBM J. Res. Develop., 45 (2001), pp. 427–438.
4. N. A. Baker, D. Sept, S. Joseph, M. Holst, and J. A. MoCammon, *Electrostatics of nanosystems: Application to microtubules and the ribosome*, Proc. Nat. Acad. Sci. USA, 98 (2001), pp. 10037–10041.
5. R. Bank and M. Holst, *A new paradigm for parallel adaptive meshing algorithms*, SIAM Rev., 45 (2003), pp. 291–323.

6. F. B. Belgacem, *The mortar finite element method with Lagrange multipliers*, Numer. Math., 84 (1999), pp. 173–197.

7. C. Bernardi, Y. Maday, and A. Patera, *A new nonconforming approach to domain decomposition: The mortar element method*, in H. Brezis et al., eds., Nonlinear Partial Differential Equations and Their Applications, vol. XI, Pitman Res. Notes. Math. Ser., 299, Longman Sci. Tech., Harlow, 1994, pp. 13–51.

8. M. A. Branch, T. F. Coleman, and Y. Li, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, SIAM J. Sci. Comput., 21 (1999), pp. 1–23.

9. S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, 2nd edn., Springer, New York, 2002.

10. R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Progr., 40 (1988), pp. 247–263.

11. W. Chen, Y. Shen and Q. Xia, *A mortar finite element approximation for the linear Poisson–Boltzmann equation*, Appl. Math. Comput., 164 (2005), pp. 11–23.

12. T. F. Coleman and Y. Li, *An interior, trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optimization, 6 (1996), pp. 418–445.

13. T. F. Coleman and Y. Li, *On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds*, Math. Progr., 67 (1994), pp. 189–224.

14. M. E. Davis, J. D. Madura, B. A. Luty, and J. A. McCammon, *Electrostatics and diffusion of molecules in solution: Simulations with the University of Houston Browian dynamics program*, Comp. Phys. Comm., 62 (1991), pp. 187–197.

15. L. Debnath and P. Mikusinnski, *Introduction to Hilbert Spaces with Applications*, Academic Press, New York, 1990.

16. S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

17. I. Ekeland and R. Témam, *Convex Analysis and Variational Problems*, SIAM, Philadelphia, 1999.

18. K. E. Forsten, R. E. Kozack, D. A. Lauffenburger, and S. Subramaniam, *Numerical solution of the nonlinear Poisson–Boltzmann equation for a membrane-electrolyte system*, J. Phys. Chem., 98 (1994), pp. 5580–5586.

19. G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd edn., John Hopkins University Press, Baltimore, MD, 1996.

20. J. Hoffman and A. Logg, PUFFIN Web page: http://www.fenics.org/puffin/, 2004.

21. J. Hoffman, J. Jansson, A. Logg, and G. N. Wells, DOLFIN Web page: http://www.fenics.org/dolfin, 2003.

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

23. 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 (2000), pp. 1319–1342.

24. B. Honig and A. Nicholls, *Classical electrostatics in biology and chemistry*, Science, 268 (1995), pp. 1144–1149.

25. B. Al-Lazikani, J. Jung, Z. Xiang, and B. Honig, *Protein structure prediction*, Current Option in Chemical Biology, 5 (2001), pp. 51–56.

26. B. P. Lamichhane and B. I. Wohlmuth, *Mortar finite elements for interface problems*, Computing, 72 (2004), pp. 333–348.

27. The MathWorks, Inc., *MATLAB Partial Differential Equation Toolbox 1.0.8* Web page: http://www.mathworks.com/products/pde/, 2005.

28. J. J. Moré and D. C. Sorensen, *Computing a Trust Region Step*, SIAM J. Sci. Stat. Comput., 3 (1983), pp. 553–572.

29. J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2000.

30. W. Rocchia, E. Alexov, and B. Hoing, *Extending the applicability of the nonlinear Poisson–Boltzmann equation: Multiple dielectric constants and multivalent ions*, J. Phys. Chem. B, 105 (2001), pp. 6507–6514.

31. B. Roux and T. Simonson, *Implicit solvent models*, Biophys. Chem., 78 (1999), pp. 1–20.

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

33. Y. N. Vorobjev and H. A. Scheraga, *A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent*, J. Comput. Chem., 18 (1997), pp. 569–583.

34. J. Wagoner and N. A. Baker, *Solvatin forces on biomolecular structures: a comparison of explicit solvent and Poisson–Boltzmann models*, J. Comput. Chem., 25 (2004), pp. 1623–1629.

35. B. I. Wohlmuth, *A V-cycle multigrid approach for mortar finite elements*, SIAM J. Numer. Anal., 42 (2005), pp. 2476–2405.