Parallel Newton-Krylov-Schwarz Algorithms for the Three-dimensional Poisson-Boltzmann Equation in Numerical Simulation of Colloidal Particle Interactions

Feng-Nan Hwang[a], Shang-Rong Cai[a], Yun-Long Shao[b], Jong-Shinn Wu[c,*]

[a]*Department of Mathematics, National Central University, Jhongli 32001, Taiwan*
[b]*Research and Technology Promotion Center, Mingshin University of Science and Technology, Hsinchu 30401, Taiwan*
[c]*Department of Mechanical Engineering, National Chiao-Tung University, Hsinchu 30050, Taiwan*

Abstract

We investigate fully parallel Newton-Krylov-Schwarz (NKS) algorithms for solving the large sparse nonlinear systems of equations arising from the finite element discretization of the three-dimensional Poisson-Boltzmann equation (PBE), which is often used to describe the colloidal phenomena of an electric double layer around charged objects in colloidal and interfacial science. The NKS algorithm employs an inexact Newton method with backtracking (INB) as the nonlinear solver in conjunction with a Krylov subspace method as the linear solver for the corresponding Jacobian system. An overlapping Schwarz method as a preconditioner to accelerate the convergence of the linear solver. Two test cases including two isolated charged particles and two colloidal particles in a cylindrical pore are used as benchmark problems to validate the correctness of our parallel NKS-based PBE solver. In addition, a truly three-dimensional case, which models the interaction between two charged spherical particles within a rough charged micro-capillary, is simulated to demonstrate the applicability of our PBE solver to handle a problem with complex geometry. Finally, based on the result obtained from a PC cluster of parallel machines, we show numerically that NKS is quite suitable for the numerical simulation of interaction between colloidal particles, since NKS is robust in the sense that INB is able to converge within a small number of iterations regardless of the geometry, the mesh size, the number of processors. With help of an additive preconditioned Krylov subspace method NKS achieves parallel efficiency of 71% or better on up to a hundred processors for a 3D problem with 5 million unknowns.

Key words: Poisson-Boltzmann equation, overlapping Schwarz preconditioning, inexact Newton, finite element, parallel processing, colloidal particles

1. Introduction

Various numerical simulations of physical phenomena, such as like-charge colloidal particle interaction [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11], electrokinetic flows [12, 13], semiconductor device simulation [14, 15, 16], and biomolecule modeling [17, 18, 19, 20, 21] require fast, reliable, and accurate numerical solutions of the Poisson-Boltzmann equation (PBE), which belongs to a class of semilinear elliptic partial differential equations. Since in general finding analytical solutions to PBE with a complicated geometry or general boundary conditions in many applications is not possible, the only remedy is to solve the problem numerically. A general numerical procedure is to replace a continuous PBE with an algebraic nonlinear system of equations by some numerical scheme, such as finite differences [3, 22], finite elements [1, 6, 7, 19, 20], or finite volumes [14, 21]. See Ref. [23] for a comprehensive survey of recent developments on numerical methods, including some novel approaches, for PBE with an emphasis on biophysics applications.

Numerical difficulties arise in solving PBE due to the presence of electric double layers (EDL) around the charged objects [12] and the effect of strong nonlinearity of the exponentially nonlinear term, e.g., the right-hand side of Eq. (1). In this situation, the solution changes dramatically near the boundary imposed by Dirichlet conditions, which is most interesting to engineers and scientists. To resolve the details in the EDL region, parallel processing is necessary for accurate discretization of the PBE using local mesh refinements with a large number of the mesh points, especially in the three-dimensional cases. A great amount of research work in the past was devoted to developing adaptive finite element solvers for the axisymmetric or three-dimensional PBE, which have been implemented by Bowen and Sharif [1], Cortis and Friensner [19, 20], Dyshlovenko [6, 7], Holst and coworkers [21], and Wu and coworkers [24] both in serial and in parallel using different adaptive meshing algorithms.

On the other hand, although in general a nonlinear solver is the kernel of a scientific software package and plays an important role in the whole process of numerical simulations, to the best of our knowledge, very few studies available in the literature focus on the development of parallel nonlinear iterative algorithms for solving PBE and the investigation of their parallel performances on a PC cluster. Some of these studies include: (a) Sirotkin and Tarvainen [16] proposed two families of parallel domain decomposition algorithms for a general singularly perturbed semilinear elliptic problem. One is a red-black block type alternating

[*]Corresponding author. Tel: +886-3-573-1693; Fax: +886-3-572-0634
Email addresses: hwangf@math.ncu.edu.tw (Feng-Nan Hwang), 942201002@cc.ncu.edu.tw (Shang-Rong Cai), ylshao@must.edu.tw (Yun-Long Shao), chongsin@faculty.nctu.edu.tw (Jong-Shinn Wu)

Schwarz method and the other one is a two-level Schwarz iterative method, where an overlapping additive Schwarz iterative method and an alternating Schwarz method on the overlapping region are used at the first level and the second level, respectively. An inexact Newton method in conjunction with an incomplete Cholesky preconditioned conjugate gradient method is employed as a subdomain solver. (b) Li [14] employed a parallel monotone iterative method for solving the semi-linear elliptic PDE with applications in semiconductor device simulation and his parallel code achieved about 80% on up to 16 processors for the problem with about 500 thousand unknowns. In fact, this method can be viewed as a special case of an inexact Newton method, where the Jacobian systems are solved by a point-wise Jacobi method with one inner iteration. The advantage of this method that it is able to monotonically converge to a desired solution with a larger radius of convergence starting from an arbitrary initial guess and it is easily parallelized. However, the method possesses only a linear rate of convergence, in particular the number of monotone iterations typically required for convergence is large. (c) Baker et al. [17] showed 80% parallel efficiency of their adaptive Poisson-Boltzmann solver (APBS) for the biomolecule calculation on up to 32 processors. The APBS is publicly available and employs an inexact Newton method as a nonlinear solver and an algebraic multigrid method as a Jacobian solver. Later, in [18] they reported results on large scale calculations for the linearized PBE using 686 processors.

A well-known interesting open problem till now in colloidal science is to provide a theoretical explanation for the experimentally-observed attraction between two charged particles at a long range distance under a flat plane [25]. Some attempts using numerical approaches have been tried in the past, e.g., [2]. However, it has been proved (for instance, see [8, 9, 10]) that the results [2] for the attraction of the like-charged identical particles in a confined cylindrical pore are erroneous. No such attraction is possible within the PBE theory. Later the absence of the attraction in this system was demonstrated numerically in [3, 7]. Since our focus is on parallel algorithm developments, we restrict our discussion on the cases in which the PBE theory is valid.

The aim of our research is to develop some parallel domain decomposition based algorithm for solving large sparse nonlinear systems of equations arising from the finite element discretization of the three-dimensional PBE, which is used for modeling colloidal phenomena and investigate its parallel performance on a parallel PC cluster. As an application in colloidal science, colloidal dispersion requires knowledge of the electrostatic potential distribution, which can then be used to calculate other physical quantities, such as particle-particle interaction force. Features of particle interaction are of great importance for the stability and properties of colloidal dispersions.

Our proposed parallel algorithm is based on a Newton-Krylov-Schwarz (NKS) algorithm [26, 27], which is a general framework for solving large nonlinear systems of equations arising from the discretization of PDEs in computational science and engineering. As its name suggests, the algorithm consists of three important components. The inexact Newton method with backtracking (INB) [28, 29] is used as a nonlinear solver. In each Newton step, an overlapping Schwarz-type preconditioner is employed to accelerate the convergence rate of Krylov subspace methods, which are used for solving linear Jacobian systems. The Schwarz-type preconditioner is ideal for parallel computing, since all subdomain problems are independent of each other and can be solved in parallel.

The remainder of this paper is organized as follows. A standard Galerkin finite element discretization of the PBE used for modeling the colloidal particle interaction is given first, followed by a detailed description of the parallel NKS algorithm for solving the resulting nonlinear algebraic system of equations. Then, the results of parallel code validation with previous published numerical solutions are presented and a realistic three-dimensional problem that involves two like-charge particles in a rough micro-capillary is simulated to demonstrate its applicability for complex geometry. Next the parallel performance of this PBE solver is discussed. Finally, the paper is summarized with some important conclusions and future directions of the research are given.

2. A model problem and its finite element formulation

Under the assumption that the charge density obeys an equilibrium Boltzmann distribution [12], the distribution of electrostatic potential around charged colloidal particles in a symmetric electrolyte (z:z) solution can be found by solving the three-dimensional PBE in the dimensionless form:

$$\begin{cases} \frac{\partial^2 \overline{\psi}}{\partial \overline{x_1}^2} + \frac{\partial^2 \overline{\psi}}{\partial \overline{x_2}^2} + \frac{\partial^2 \overline{\psi}}{\partial \overline{x_3}^2} = \sinh(\overline{\psi}) & \text{in} \quad \Omega \subset R^3, \\ \overline{\psi} = g_D & \text{on} \quad \Gamma_D, \\ \dfrac{\partial \overline{\psi}}{\partial n} = 0 & \text{on} \quad \Gamma_N, \end{cases} \tag{1}$$

where $\overline{x_1} = x_1/\kappa^{-1}$, $\overline{x_2} = x_2/\kappa^{-1}$, $\overline{x_3} = x_3/\kappa^{-1}$, and $\overline{\psi} = \dfrac{ze\psi}{k_bT}$. Here, $\kappa^2 = \dfrac{2n_0z^2e^2}{\varepsilon\varepsilon_0k_bT}$, where z is the valence of the ions, n_0 is the average number concentration, T is the absolute temperature, ε the permittivity ratio, and ε_0 is the vacuum permittivity. Note that κ is the so-called Debye-Huckel parameter and its inverse represents the characteristic length of the electrical double layer. Two types of boundary conditions are imposed: a Dirichlet-type boundary condition on Γ_D and a Neumann-type boundary condition which is assumed be zero on

Γ_N. To simplify the notation, we drop the bars for all variables throughout the paper.

To discretize the PBE (1), we use the standard Galerkin finite element method [30] on a given unstructured tetrahedral mesh, $\mathcal{T}^h = \{K\}$ with an element diameter h. Let $\Psi^h \subset H^1(\Omega)$ be a linear or quadratic finite element space for ψ:

$$\Psi^h = \{\psi \in (C^0(\Omega) \cap H^1(\Omega) : \ \psi|_K \in P_m(K), \ K \in \mathcal{T}^h \ \},$$

where $C^0(\Omega)$ is the space consisting of all continuous functions defined on Ω, $H^1(\Omega)$ is a Sobolev space defined by

$$H^1(\Omega) = \{\psi : \int_\Omega (|\nabla\psi|^2 + \psi^2)\, d\mathbf{x} < \infty\},$$

and $P_m(K)$ is the space of all polynomial functions of degree m defined on an element K. Here $m = 1$ or 2. We define the weighting and trial function spaces Ψ_0^h and Ψ_g^h as

$$\Psi_0^h = \{\psi \in \Psi^h : \psi = 0 \text{ on } \Gamma_D\} \text{ and } \Psi_g^h = \{\psi \in \Psi^h : \ \psi = g_D \text{ on } \Gamma_D\},$$

respectively, and then the Galerkin finite element formulation for PBE (1) can be written as: Find $\psi^h = \varphi + g \in \Psi_g^h$ such that

$$a(\psi^h, \varphi) + (f_\rho(\psi^h), \varphi) = 0 \quad \forall \varphi \in \Psi_0^h, \tag{2}$$

with

$$a(\psi, \varphi) = \int_\Omega \nabla\psi \cdot \nabla\varphi \, d\mathbf{x}$$

and

$$(f_\rho(\psi), \varphi) = \int_\Omega \sinh(\psi)\varphi \, d\mathbf{x}$$

Let $\{N_l\}_{l=1}^{Nen}$ be a set of global shape functions for Ψ_0^h, where N_{en} is the total number of nodes, excluding the Dirichlet boundary nodes. Similarly, $\{N_m^g\}_{m=1}^{N_{en}^g}$ is a set of shape functions associated with the Dirichlet boundary nodes. Then, the approximate ψ^h can be written in term of the global shape functions and nodal values given by

$$\psi^h = \sum_{l=1}^{N_{en}} \beta_l N_l + \sum_{m=1}^{N_{en}^g} g_m N_m^g,$$

where g_m is the nodal interpolate of g at $\mathbf{x}_m \in \Gamma_D$, and φ^h is selected to be $N_i, i = 1, ..., N_{en}$. Substituting these two functions into the finite element formulation (2) yields a large sparse nonlinear algebraic system of equations, $F : R^{N_{en}} \to R^{N_{en}}$ such that

$$F(x) = 0, \tag{3}$$

where $F = (F_1, F_2, ..., F_{N_{en}})^T$ and $x = (\beta_1, \beta_2, ..., \beta_{N_{en}})^T$ with

$$F_i(x) = \sum_{l=1}^{N_{en}} \beta_l a(N_l, N_i) + (f_\rho(\sum_{l=1}^{N_{en}} \beta_l N_l + \sum_{m=1}^{N_{en}^g} g_m N_m^g), N_i) + \sum_{m=1}^{N_{en}^g} g_m a(N_m^g, N_i),$$

for $i = 1, 2, ..., N_{en}$.

3. A Newton-Krylov-Schwarz algorithm

3.1. A description of the algorithm

We employ a NKS algorithm for solving large nonlinear systems of equations (3). To be more specific, we describe NKS as follows. Let $x^{(0)}$ be a given initial guess and assume $x^{(k)}$ is the current approximation of x^*. Then a new approximation $x^{(k+1)}$ can be computed via the following steps:

Step 1: Find a Newton direction $s^{(k)}$ by solving approximately the following preconditioned Jacobian system using a Krylov subspace method, such as GMRES [31],

$$J_k M_k^{-1} y = -F(x^{(k)}), \text{ with } s^{(k)} = M_k^{-1} y, \tag{4}$$

where J_k is the Jacobian of F evaluated at $x^{(k)}$ and M_k^{-1} is called a right preconditioner. We will discuss the construction of M_k^{-1} based on a Schwarz framework further in the later subsection.

Step 2: Obtain the new approximation $x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}$, where $\lambda^{(k)} \in (0, 1]$ is a step length of the Newton direction.

In NKS, the accuracy of the solution to the Jacobian systems is controlled by the parameter, η_k to force the condition

$$\|F(x^{(k)}) + J_k s^{(k)}\|_2 \leq \eta_k \|F(x^{(k)})\|_2$$

to be satisfied. η_k is often referred to as the forcing term. If the chosen forcing term is small enough, the algorithm reduces to the exact Newton algorithm. The step length, $\lambda^{(k)} \subset (0, 1]$, is selected so that

$$f(x^{(k)} + \lambda^{(k)} s^{(k)}) \leq f(x^{(k)}) + \alpha \lambda^{(k)} \nabla f(x^{(k)})^T s^{(k)}, \tag{5}$$

where the merit function $f : R^{Nen} \rightarrow R$ is defined as $\|F(x)\|_2^2/2$, and the parameter α is used to ensure that the reduction of f is sufficient. Here, typically $\alpha = 10^{-4}$. When the condition (5) does not hold for the case of $\lambda^{(k)} = 1$, a cubic linearsearch technique [28] is employed to determine the step length $\lambda^{(k)}$.

3.2. Calculation of Jacobian matrices

The Jacobian matrices are the key components in NKS. We can write the Jacobian matrices in the form of

$$J_k = J^L + J_k^N,$$

where J^L and J_k^N are the linear part and nonlinear part of the Jacobian, respectively. The matrix J^L can be explicitly written as $J^L = (J_{ij}^L)$ with elements $J_{ij}^L = a(N_j, N_i)$. Since J^L is unchanged during Newton iteration, it can be formed once and then reused throughout the computation. On the other hand, the element of the matrix J_k^N can be computed through

$$(J_{ij}^N)_k = (f_\rho'(\sum_{l=1}^{N_{en}} \beta_l^{(k)} N_l + \sum_{m=1}^{N_{en}^g} g_m N_m^g) N_j, N_i),$$

where $f_\rho'(x) = \cosh(x) \geq 0$ for $x \geq 0$. Note that the resulting Jacobian systems are symmetric and positive definite since it is easily seen that $(J_{ij})_k = (J_{ij})_k$ and for $\varphi^h = \sum_{i=1}^{N_{en}} \beta_i N_i \in \Psi_0^h$ and $\psi^h = \sum_{l=1}^{N_{en}} \beta_l^{(k)} N_l + \sum_{m=1}^{N_{en}^g} g_m N_m^g$ at the kth Newton iteration, we have

$$\begin{aligned}
\sum_{i=1}^{N_{en}} \sum_{j=1}^{N_{en}} \beta_i (J_{ij})_k \beta_j &= a(\sum_{j=1}^{N_{en}} \beta_j N_j, \sum_{i=1}^{N_{en}} \beta_i N_i) \\
&+ (f_\rho'(\sum_{l=1}^{N_{en}} \beta_l^{(k)} N_l + \sum_{m=1}^{N_{en}^g} g_m N_m^g) \sum_{j=1}^{N_{en}} \beta_j N_j, \sum_{i=1}^{N_{en}} \beta_i N_i) \\
&= a(\varphi^h, \varphi^h) + (f_\rho'(\psi^h)\varphi^h, \varphi^h) \geq 0
\end{aligned}$$

The equality holds only if $\varphi^h = 0$.

3.3. A parallel one-level overlapping Schwarz preconditioner

In this section we introduce a parallel additive Schwarz preconditioner for the Jacobian systems. Our preconditioner is an extension of the overlapping Schwarz preconditioner for the linear elliptic PDEs [32, 33, 34]. Let $\{\Omega_i^h, i = 1, ..., np\}$ be a non-overlapping subdomain partition with boundary $\partial\Omega_i^h$ and assume the union of these non-overlapping subdomains covers the entire domain Ω and its mesh $\mathcal{T}^h$. Here, np is the number of processors in the parallel computer. To obtain overlapping subdomains, we expand each subdomain Ω_i^h to a larger subdomain $\Omega_i^{h,\delta}$ with the boundary $\partial\Omega_i^{h,\delta}$. Here δ is an integer indicating the size of overlap. We assume that neither $\partial\Omega_i^h$ nor $\partial\Omega_i^{h,\delta}$ cut any elements of $\mathcal{T}^h$ and use $\mathcal{T}_i^h$ and

$\mathcal{T}_i^{h,\delta}$ to denote the collection of mesh points in Ω_i^h and $\Omega_i^{h,\delta}$, respectively. Now, we define the overlapping subdomain space as

$$\Psi_i^h = \{\psi^h \in \Psi^h \cap \left(H^1(\Omega_i^{h,\delta})\right)^3 : \psi^h = 0 \text{ on } \Omega_i^{h,\delta}\},$$

which is a subspace of Ψ^h, if all subdomain functions are extended to the whole domain by zero.

Let R_i be a restriction operator, which transfers data from $\Psi^h \to \Psi_i^h$. In the matrix representation, R_i is an $n_i \times n$ matrix with values of either 0 or 1, where n and n_i are the total numbers of *interior* mesh points in $\mathcal{T}^h$ and $\mathcal{T}_i^{h,\delta}$, respectively, and $\sum_{i=1}^N n_i \geq n$. Then, the prolongation operator R_i^T can be defined as the transpose of R_i. It should be noted that the multiplication R_i (or R_i^T) with a vector does not involve any arithmetic operation, but does involve communication in a distributed parallel implementation. The restriction operator R_i collects the data from neighboring subdomains, while the prolongation operator R_i^T sends a partial solution to neighboring subdomains. Using restriction and prolongation operators, we can define the one-level overlapping additive Schwarz-type preconditioner,

$$M_k^{-1} = \sum_{i=1}^{np} R_i^T J_i^{-1} R_i, \tag{6}$$

where the sub-Jacobian matrix is defined as $J_i = R_i J_k R_i^T$. For simplicity of implementation, we assign each subdomain problem to a single processor. In practice, one does not need to form the preconditioner M_k^{-1} explicitly and only the operation of the preconditioner with a vector, $u = M_k^{-1} w$, is needed, if a Krylov subspace method is employed to solve the linear system. For example, in the distributed memory parallel implementation, each processor performs the following steps:

- Collect the data from the subdomain and its neighboring subdomains, $w_i = R_i w$
- Solve $J_i x_i = w_i$, using an exact or an inexact sparse direct solver
- Send/receive the partial solution to/from its neighboring subdomain, $\hat{x}_i = R_i^T x_i$
- Calculate the sum via $u = \sum_{i=1}^N \hat{x}_i$

Since J_k is symmetric positive definite, the theoretical convergence results of the additive Schwarz methods due to Dryja and Widlund [35, 36] can be applied.

Assume that each subdomain problem is solved exactly. Then the condition number of $M_k^{-1}J_k$ satisfies

$$\kappa(M_k^{-1}J_k) \leq \frac{C}{H^2}(1 + \frac{H}{\delta}),$$

where the constant C is independent of the parameters H, h, and δ. Here, H is the diameter of a nonoverlapping subdomain. From the above estimate, we summarize the expected convergence behavior of an additive Schwarz preconditioned Krylov subspace method for solving Eq. (4) as follows:

1. Assuming δ is order of H, the number of iterations is independent of h, but not H. The number of iterations increases proportional to $1/H$. Furthermore, if quasi-uniform partitioning is used, the number of iterations is expected to be $O(np^{1/3})$.
2. The convergence rates of Schwarz can be improved by increasing the level of overlap.

4. Numerical results and discussions

Our parallel PBE solver is implemented by the Portable, Extensible Toolkits for Scientific Computation (PETSc) [37] from the Argonne National Laboratory. The parallel PBE solver has been tested and validated on different parallel computer systems. In addition, the CUBIT [38] from the Sandia National Laboratories is employed to generate unstructured meshes for complex geometry, a mesh partition tool, Parmetis [39] from the University of Minnesota, is served as the purpose of parallel processing, and the scientific visualization system, Paraview [40] from the Kitware and the Sandia National Laboratories is used for data analysis. All computations were carried out in double-precision. The timing results reported in seconds were obtained using the IBM cluster 1350 at the National Center for High-performance Computing in Taiwan. The IBM cluster 1350 consists of 512 compute nodes; each node has two Intel Woodcrest 3.0GHz Dual-Core processors with 16 GB of memory. The compute nodes are interconnected by InfiniBand switches of 2 GB/s bandwidth.

We declare that the numerical solution converges when the stopping condition for Newton

$$\|F(x^{(k)})\| \leq \max\{10^{-10}\|F(x^{(0)})\|, 10^{-50}\}.$$

is satisfied. A zero initial vector is employed for all test runs. It should be noted that from our numerical experiences, as can be found in Tables 4 to 7, INB is quite robust and the line search technique is never needed for all test cases, i.e. INB always took a full Newton step and exhibited the local quadratic convergence behavior. A right additive Schwarz preconditioned GMRES [31] with a zero initial guess is employed to solve the resulting Jacobian system with the constant forcing term $\eta_k = 10^{-4}$.

4.1. Parallel code validation

We first validate the correctness of our parallel PBE solver by considering the two following test cases as benchmark problems. One is two isolated charged particles (Problem 1) and the other one is two colloidal particles in a cylindrical pore (Problem 2). In both test cases, due to symmetry of the solution with respect to the yz-plane and the x-axis, only 1/32 of a cylindrical computational domain (See Figure 1 for the detailed geometry configuration) is considered. For the case of two isolated charged particles, we set the radius and length of the cylinder, $r_c = 12.0$ and $\overline{EF} = 17.25$, respectively. One sphere of radius $r_s = 5.0$ is located at $(5.25, 0.0, 0.0)^T$. The separation distance between two spheres $2\overline{AB}$ is set to be 1.0. Except that the potential on the surface of the sphere is kept to be 2.0 the Neumann boundary condition, $\partial\psi/\partial n = 0$ is imposed on all boundaries. For the case of two colloidal particles in a cylindrical pore, the corresponding parameters are set as follows. $r_s = 1.185$, $r_c = 9.115$, and $\overline{EF} = 17.25$. $2\overline{AB}$ is varied from 1.0 to 8.0. Constant potentials are imposed on the surface of the colloidal particle $\psi = 3.0$ and the surface of the pore $\psi = 5.0$. The homogenous Neumann boundary condition is imposed on the rest of the boundaries. Typical computed electrostatic potential distributions around the charged sphere for both cases are shown in Figure 2.

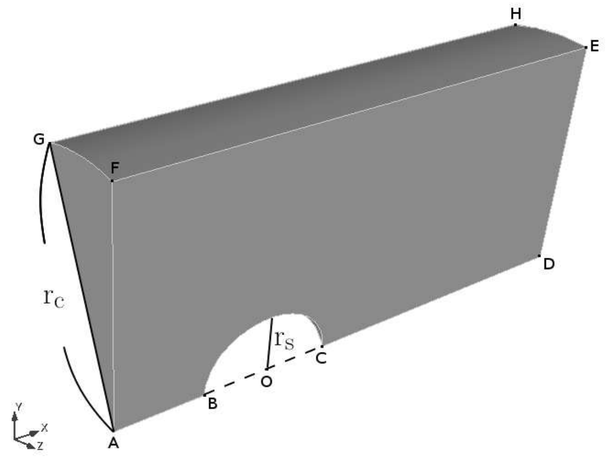

Figure 1: A computational domain for both of Problem 1 and Problem 2.

We calculate numerically the electrostatic forces on the sphere and/or along with the midplane and then compare our results with other published ones. Recall that the total stress tensor can be written as

$$\mathbf{T} = -p\mathbf{I} + \widehat{\mathbf{T}},$$

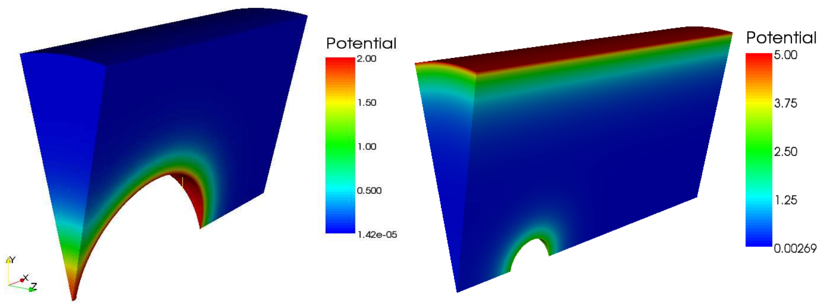

Figure 2: Electrostatic potential distributions: Problem 1 (left) and Problem 2 (Right).

where p is the osmotic pressure difference between the electrolyte on the particle surface and the bulk solution, $\mathbf{I}$ is an identity tensor, and Maxwell's stress tensor $\widehat{\mathbf{T}}$ is defined as $EE - \frac{1}{2}(E \cdot E)\mathbf{I}$ with the electric field $E = -\nabla\psi = (E_1, E_2, E_3)^T$. Furthermore, the flow is assumed to be stationary and the effect of gravity is neglected, so p can be expressed as $2[\cosh(\psi) - 1]$ [12] by solving analytically the incompressible Navier-Stokes equations. Then the electrostatic force between between two spheres can be calculated via

$$F_s = \int_S \mathbf{T} \cdot n \, dS,$$

where n is a unit outward normal vector to the surface of the sphere, S. The electrostatic force along the midplane, F_m, can be defined in a similar manner by replacing S with the surface of a cylindrical shaped control volume. To accurately compute the electrostatic force, PBE needs to be solved with high resolution meshes within the EDL. For Problem 1, we employe the ability of CUBIT to generate nonuniform unstructured meshes, which are locally refined both near the surface of the sphere and along the midplane. At each level, we refine one layer of the elements closest to the boundaries, where each element is divided into four new subelements as shown in Figure 3.

Table 1 summarizes the evolution of the number of elements, the number of nodes, and the computed electrostatic forces F_s and F_m interacting between two spheres in the x-axis direction at different refinement levels. We should note that even though quadratic finite elements are used for calculations only first order accuracy for the computed electrostatic forces is expected. Hence, to further improve the accuracy of computed electrostatic force, we apply Richardson's extrapolation technique [41], which is given as follows.

$$R_j(h^*) = R_{j-1}\left(\frac{h^*}{2}\right) + \frac{1}{2^{j-1} - 1}\left[R_{j-1}\left(\frac{h^*}{2}\right) - R_{j-1}(h^*)\right],$$

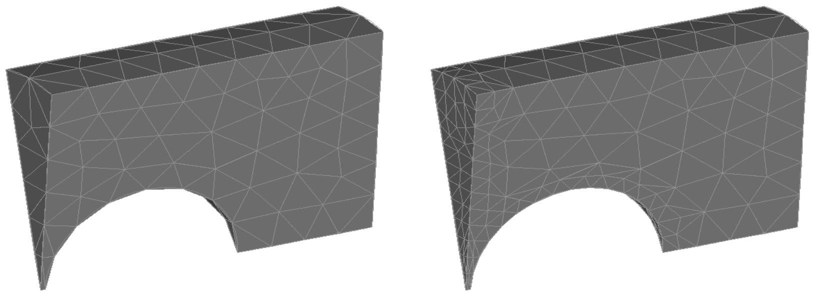
Figure 3: The surface mesh distributions for Problem 1: the original uniform mesh (left) and the mesh with local mesh refinements around the surface of the sphere and along midplane (Right).

where $j = 1, 2, ...$ and h^* is the mesh size of an element on the sphere or along the midplane. As shown in Table 2, the extrapolated electrostatic forces using quadratic finite elements listed in the last column of the table agree with other published numerical results, e.g., $F_s = 48.840$ and $F_m = 48.835$ in [7] also in [4] $F_m = 48.835$, which is the best known result for this problem.

Refinement	# of elements	Linear element			Quadratic element		
Level		DOFs	F_s	F_m	DOFs	F_s	F_m
0	101,992	19,792	40.225392	46.983952	147,117	42.036832	48.441344
1	120,803	24,138	44.192416	48.319296	176,736	45.221360	48.736912
2	194,739	40,773	46.308784	48.624880	291,503	46.957888	48.810864
3	489,652	106,084	47.461728	48.697264	745,320	47.878272	48.829088
4	1,670,467	365,435	48.095952	48.715328	2,554,053	48.351152	48.833600
5	6,401,288	1,400,100	48.432352	48.719872	9,783,256	48.590864	48.834720

Table 1: Computed electrostatic forces on the sphere (F_s) and along the midplane (F_m) in the x-axis direction for Problem 1. DOFs: Total degrees of freedom.

Next we consider Problem 2. Figure 4 shows the comparison of the computed electrostatic forces F_s in the x-axis direction at different levels of mesh refinement and the reference solution [7] with respect to the separation distance. We perform local mesh refinement near the surface of the sphere and found that our computed electrostatic force gradually converges toward the reference curves as more levels of local mesh refinements are performed. Also note that no matter how long the separation distance is we always obtained repulsive electrostatic forces, even though the electrostatic forces are quite small (but always are positive), e.g., $F_m = 0.031584$ and 0.026000 at the separation distances, 6.0 and 8.0, respectively. Note that the reference solutions [7] for both problems we compared with were

F_s							
Level	h^*	R_1	R_2	R_3	R_4	R_5	R_6
0	0.515187	42.036832					
1	0.257779	45.221360	48.405888				
2	0.128913	46.957888	48.694416	48.790592			
3	0.064460	47.878272	48.798656	48.833403	48.839519		
4	0.032231	48.351152	48.824032	48.832491	48.832361	48.831884	
5	0.016116	48.590864	48.830576	48.832757	48.832795	48.832824	48.832854

F_m							
Level	h^*	R_1	R_2	R_3	R_4	R_5	R_6
0	0.515187	48.441344					
1	0.257779	48.736912	49.032480				
2	0.128913	48.810864	48.884816	48.835595			
3	0.064460	48.829088	48.847312	48.834811	48.834699		
4	0.032231	48.833600	48.838112	48.835045	48.835078	48.835103	
5	0.016116	48.834720	48.835840	48.835083	48.835088	48.835089	48.835089

Table 2: F_s and F_m calculations using Richardson's extrapolation in the x-axis direction for Problem 1 (quadratic element)

obtained by solving axisymmetric PBE using a sequential adaptive quadratic finite element code.

4.2. 3D realistic applications

Next, we consider the test case of two equal potential spherical particles in a rough charged micro-capillary (see the top of Figure 5, labeled as Problem 3), which is taken from Ref. [5] with some small modifications. This example demonstrates the applicability of our parallel PBE solver in predicting the electrostatic potential distribution for a realistic colloidal particle interaction problem. No thorough physical interpretations or physical parametric studies are explored in the current study, although these are of course worthy for further investigation.

We place the centers of these two charged particles at $O_1 = (1.2, 0.1, 0.0)^T$ and $O_2 = (-1.2, -0.1, 0.0)^T$ such that the segment connecting the two points is not parallel to the axis of the capillary. Hence, this test case truly requires a three-dimensional PBE numerical simulation. For simplicity, instead of more complicated Bézier curves we use a sine curve to generate a rough surface. As shown at the bottom of Figure 5, the radius of the particles is set to $a = 1.0$. The mean radius and the length of the capillary are $b = 1.2$ and 12.0, respectively. The wave surface with wave length $\lambda = 1.0$ and amplitude $\alpha = 0.1$ is considered. The constant potentials on the surface of the spheres as well as the wall are maintained to be -3.0.

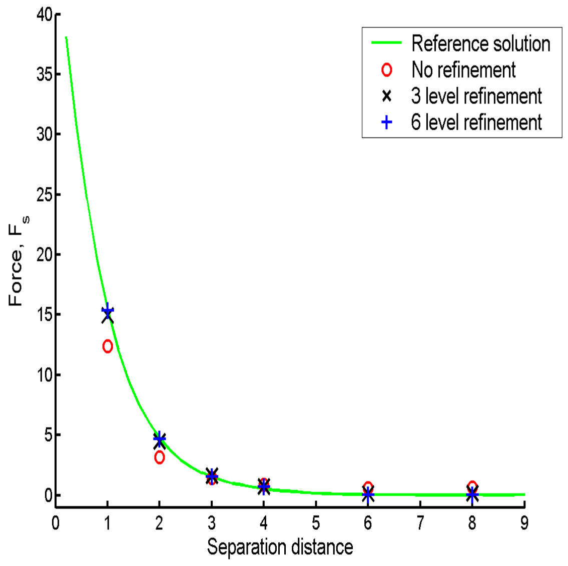

Figure 4: A comparison of the numerical electrostatic forces on the sphere F_s in the x-axis direction with the reference solution at the different separation distances for Problem 2.

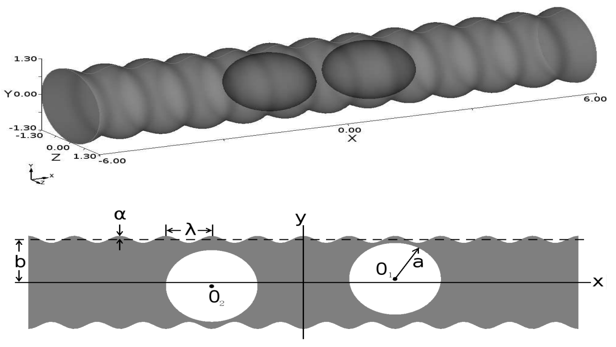

Figure 5: A computational domain (top) and a geometry configuration (bottom) for Problem 3.

Similar to the previous two test cases, we solve the PBE using a sequence of locally refined meshes near charged particles with both linear and quadratic finite elements. The corresponding electrostatic forces in all x-, y-, and z-axis directions at each refinement level are summarized in Table 3. For purposes of comparison, note that if the two charged particles are located at $(1.2, 0, 0)^T$ and $(-1.2, 0, 0)^T$ then the electrostatic force between them in the x-axis direction at the rough wall would be about 4.701, which is close to the value as shown in Figure 3 (b) of Ref. [5] when $\lambda = 1.0$, $\alpha = 0.1$ and $\kappa h = 0.4$ and is slightly larger than that in our Problem 3. Also due to a loss of axisymmetry with respect to the $x-$axis, the electrostatic forces on the surface of the particles in the y-axis direction do not vanish anymore. They are roughly identical with opposite signs. Figure 6 displays the distribution of electrostatic potential around the charged particles.

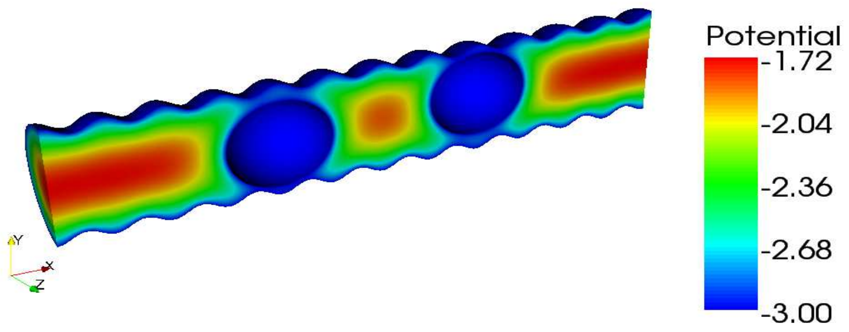

Figure 6: The distribution of electrostatic potential on the xy-plane for Problem 3.

4.3. NKS algorithmic parameter and parallel performance studies

To study the parallel performance of our algorithms, we consider fixed-storage-per-processor (weak) scaling and the fixed-problem-size (strong) scaling. For the fixed-storage-per-processor scaling, we use Problem 1 as a test case and refine the mesh uniformly as the number of processors (np) is increased so that a total number of roughly 250,000 elements is kept on each processor. A parallel algorithm is considered to be scalable in this sense when it preserves a nearly constant or at worst logarithmically growing computing time. Table 4 presents the results for the number of Newton iterations (NNI), the average number of GMRES iterations (ANLI) and the corresponding total execution time with respect to the number of processors. Here, we consider different sizes of overlap, $\delta = 1$ or 2, for the additive Schwarz preconditioner and a LU decomposition is employed as a subdomain solver. On the other hand, in practice, to save the computational cost

Refinement Level	# of elements	DOFs	$O_1 = (1.2, 0.1, 0.0)$ $O_2 = (-1.2, -0.1, 0.0)$		
			x-direction	y-direction	z-direction
Linear element					
0	288,350	55,418	3.609719 -3.595272	-2.265363 2.284293	-0.001925 0.004145
1	842,790	155,481	4.080840 -4.075498	-2.833951 2.836868	-0.000956 -0.003026
2	1,813,929	337,585	4.358576 -4.354893	-3.159859 3.159259	-0.001297 -0.000898
3	3,418,762	685,918	4.466005 -4.471687	-3.314075 3.309478	0.001058 0.000216
4	9,792,855	2,072,727	4.552583 -4.556071	-3.408725 3.405736	0.000704 0.000567
Quadratic element					
0	288,350	414,302	3.891844 -3.886524	-2.699316 2.702384	-0.002994 -0.008252
1	842,790	1,183,445	4.236093 -4.233961	-3.076378 3.077844	-0.001592 -0.003745
2	1,813,929	2,554,858	4.421612 -4.420854	-3.280781 3.280950	-0.000874 -0.001465
3	3,418,762	4,998,917	4.508147 -4.510742	-3.382989 3.381122	0.000131 -0.000426
Richardson's Extrapolation			4.586885 -4.595647	-3.483692 3.478383	0.001811 0.000505

Table 3: The numerical calculation of F_s in all x-, y-, and z-axis directions for two charged particles with mesh refinements at each level for Problem 3.

and the memory use, the J_i^{-1} in M_k^{-1} often are replaced by an inexact solver, such as ILU with some levels of fill-in. We tested ILU as a subdomain solver with different levels of fill-in, k from 0 to 4 and different size of overlap, δ from 1 to 3 and the results with the best combination of k and δ in terms of computing time with respect to np are summarized in Table 5.

		NNI		ANLI		Total Time (sec.)	
np	# of elements	$\delta = 1$	$\delta = 2$	$\delta = 1$	$\delta = 2$	$\delta = 1$	$\delta = 2$
4	1,020,790	4	4	15.25	10.50	1108.2	1378.6
8	2,023,744	4	4	19.25	15.00	1371.3	1657.7
16	4,050,077	4	4	28.00	21.25	1488.3	1862.0
32	8,084,949	4	4	36.50	26.25	1319.7	2080.1
64	15,921,533	4	4	63.50	36.00	1053.9	1845.9
128	32,064,416	4	4	83.00	53.25	1086.1	2184.7

Table 4: The number of the nonlinear iterations the average number of linear iterations and corresponding timing results for Problem 1. Exact subdomain solve: LU with $\delta = 1$ or 2

				Iteration counts		Time (sec.)	
np	# of elements	ILU(k)	δ	NNI	ANLI	NNI	ANLI
4	1,020,790	0	1	4	55.0	12.1	9.4
8	2,023,744	0	1	4	74.3	17.1	14.3
16	4,050,077	1	1	4	67.5	21.6	18.4
32	8,084,949	0	1	4	132.0	31.4	28.0
64	15,921,533	0	1	4	183.8	47.3	42.8
128	32,064,416	1	1	4	165.0	56.1	51.2

Table 5: The number of the nonlinear iterations the average number of linear iterations and corresponding timing results for Problem 1. Inexact subdomain solve: ILU(k), k is varied from 1 to 4 and the range of δ is from 1 to 3.

Some observations are made from these two tables as follows.

1. As the theory suggests, for the case using LU as a subdomain solver, the convergence of GMRES iterations (see ANLI columns in Table 4) improves as the overlap is increased. However, the smaller number of GMRES iterations does not imply faster convergence in terms of the running time of the programs. Since a large portion of the running time is spent on communication for a larger size of overlap, the NKS algorithm with a minimum size of overlap is always a winner. A similar situation occurs for the case using ILU as the subdomain solver.

2. If δ/H is kept constant, for the case using LU as the subdomain solver the number of GMRES iterations (roughly $O(log_2(np))$ grows slightly faster than the theoretical prediction ($O(np^{1/3})$), where a quasi-uniform partitioning is assumed. In order to further improve the convergence rate of GMRES, adding a coarse grid space is a known solution to remove the dependency of H (or np).
3. An additive Schwarz preconditioned GMRES with LU as a subdomain solver is more scalable than that with ILU; the computing time for the former case remains almost constant, while in the latter case it grows roughly proportional to $log_2(np)$. However the latter case is much faster than the former case.

On the other hand, for the fixed-problem-size scaling case, we consider Problem 3 as a test case and increase the number of processors (np), while the problem sizes are kept constant (the finest meshes used for both linear and quadratic element case). Parallel efficiency for the np_2 processor case is defined as $E = T_{np_1}/T_{np_2}$, where T_{np_1} and T_{np_2} are the execution times obtained by using np_1 and np_2 processors, respectively. Ideally, $E_{np_1} = 100\%$. In this set of numerical experiments, the size of overlapping δ is set to be 1 and each subdomain problem is solved by an incomplete LU decomposition with zero level of fill-in (ILU(0)). From Tables 6 and 7, we found that the NKS algorithm for this problem is nonlinearly scalable; i.e. the number of nonlinear iterations (NNI) is independent of np. On the other hand, although the average of linear iterations (ANLI) slightly grows as np increases, the overall efficiency of our algorithms still achieves about 57% for the linear element case and 71% for the quadratic element case on up to 128 processors. To further analyze the performance of our parallel PBE solver, these two tables also present the time breakdown information for three key components of the NKS algorithm, including the timings for solving the Jacobian systems (JSolve), evaluating the vector functions (FEval) and the Jacobian matrices (JEval), respectively. As expected, the most computationally expensive phase is to solve the Jacobian systems; it takes at least 57% of the total execution time for all test runs. Also notice that for linear element cases, JSolve and FEval do not scale well as np increases from 64 to 128. As a result, total efficiency drops dramatically from 73% to 57%.

5. Concluding remarks

In this paper, we developed a parallel three-dimensional finite element PBE solver based on the NKS algorithm using PETSc and investigated its performance on a PC cluster of parallel machines. We found that the coupling of the parallel PBE solver with local mesh refinement near charged particles increased the

np	Iteration counts		Time (sec.)				E_4 (%)
	NNI	ANLI	JSolve	FEval	JEval	Total time	
4	5	21.2	57.7	4.2	18.5	80.6	100
8	5	22.4	29.7	2.1	9.3	41.1	98
16	5	26.6	17.2	1.1	4.8	23.0	88
32	5	28.6	9.9	0.6	2.9	13.5	75
64	5	27.8	4.5	0.4	1.8	6.9	73
128	5	28.4	2.5	0.4	1.5	4.4	57

Table 6: Parallel efficiency. Linear elements are used with mesh refinement level=4. The total number of elements: 9,792,855; the total number of the degrees of freedom: 2,072,727. $\delta = 1$ is used with ILU(0) as a subdomain solver.

np	Iteration counts		Time (sec.)				E_{16} (%)
	NNI	ANLI	JSolve	FEval	JEval	Total time	
16	5	47.6	255.6	3.1	26.7	286.0	100
32	5	46.8	144.3	1.7	14.9	161.1	89
64	5	47.8	73.4	0.9	9.2	83.6	86
128	5	54.2	44.5	0.6	5.0	50.2	71

Table 7: Parallel efficiency. Quadratic elements are used with mesh refinement level=3. The total number of elements: 3,418,762; the total number of the degrees of freedom: 4,998,917.

solution accuracy systematically. In addition, with the help of Richardson's extrapolation, one can more accurately compute some sensitive quantities like the electrostatic force. Furthermore, the parallel PBE solver was applied to simulate the force of interaction between two identical charged spheres in a rough micro-capillary to demonstrate its capability in handling a realistic three-dimensional problem. For this problem with size pertinent to practical 3D applications, our study showed that the parallel efficiency of the PBE solver reached 71% or better using 128 processors for a problem with 5 million unknowns and the performance of the parallel PBE on locally refined unstructured meshes is quite satisfactory. The parallel PBE solver that uses parallel adaptive mesh refinement techniques [24] is under development and it is expected to serve as a more powerful and efficient simulation tool for studying three-dimensional colloidal and interfacial problems in the future.

Acknowledgements

The authors appreciate the anonymous referees for their constructed suggestions used to improve the presentation of this manuscript. The authors would like to acknowledge the National Center for High-Performance Computing, Taiwan in providing resources under the national project, Taiwan Knowledge Innovation National Grid. The first two authors were supported in part by the National Science Council of Taiwan, 95-2115-M-008-007.

References

[1] W.R. Bowen, A.O. Sharif, J. Colloid Interface Sci. 187 (1997) 363
[2] W.R. Bowen, A.O. Sharif, Nature 393 (1998) 663
[3] W.R. Bowen, P.M. Williams, Colloids and Surfaces A: Physicochem. Eng. Aspects 204 (2002) 103
[4] S.L. Carnie, D.Y.C. Chan, J. Stankovish, J. Colloid Interface Sci. 195 (1994) 116
[5] P.K. Das, S. Bhattacharjee, J. Colloids interface Sci. (2004) 278
[6] P.E. Dyshlovenko, J. Comp. Phys. 172 (2001) 198
[7] P.E. Dyshlovenko, Comput. Phys. Comm. 147 (2002) 335
[8] J.C. Neu, Phys. Rev. Lett. 82 (1999) 1072
[9] J.E. Sader, D.Y.C. Chan, J. Colloid Interface Sci. 213 (1999) 268
[10] E. Trizac, Phys. Rev. E 62 (2000) 1465
[11] R. Tuinier, J. Colloid Interface Sci. 258 (2003) 45
[12] J.H. Masliyah, S. Bhattacharjee, Electrokinetic and Colloid Transport Phenomena (John Wiley & Sons, New Jersey, 2006)
[13] R.-Y. Yang, L.-M. Fu, Y.-C. Lin, J. Colloid Interface Sci. 239 (2001) 98
[14] Y. Li, Comput. Phys. Comm. 153 (2003) 359
[15] V. Sirotkin, Computers Math. Applic. 40 (2000) 645
[16] V. Sirotkin, P. Tarvainen, SIAM J. Sci. Comput., 21 (2000) 1587
[17] N.A. Baker, D. Sept, M.J. Holst, J.A. McCammon, IBM J Res Devel 45 (2001) 427
[18] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, J.A. McCammon, Proc. Natl. Acad. Sci. USA 98 (2001) 10037

[19] C.M. Cortis, R.A. Friesner, J. Comput. Chem. 18 (1997) 1570
[20] C.M. Cortis, R.A. Friesner, J. Comput. Chem. 18 (1997) 1608
[21] M.J. Holst, F. Saied, J. Comput. Chem. 16 (1995) 337.
[22] Z. Qiao, Z. Li, T. Tang, J. Comput. Math. 24 (2006) 252
[23] B. Lu, Y. Zhou, M. Holst, J.A. McCammon, Comm. Comput. Phys. 3 (2008) 973
[24] Y.-Y. Lian, K.-H. Hsu, Y.-L. Shao, Y.-M. Lee, Y.-W. Jeng, J.-S. Wu, Comput. Phys. Comm. 175 (2006) 721
[25] A.E. Larsen, D.G. Grier, Nature 385 (1997) 230
[26] X.-C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, D.P. Young, SIAM J. Sci. Comput. 19 (1998) 246
[27] D.A. Knoll, D.E. Keyes, J. Comp. Phys. 193 (2004) 357
[28] J. Dennis, R. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations (SIAM, Philadelphia, 1996)
[29] J. Nocedal, S. J. Wright, Numerical Optimization (Springer-Verlag, New York, 1999)
[30] C. Johnson, Numerical Solution of Partial Differential Equations by the Finite Element Method (Cambridge University Press, Cambriage, 1987)
[31] Y. Saad, M.H. Schultz, SIAM J. Sci. Stat. Comp. 7 (1986), 856
[32] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, (Oxford University Press, Ofxord, 1999)
[33] B. Smith, P. Bjørstad, W. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations (Cambridge University Press, Cambridge, 1996)
[34] A. Toselli, O. Widlund, Domain Decomposition Methods – Algorithms and Theory (Springer-Verlag, Berlin, 2005)
[35] M. Dryja, O. B. Widlund, Tech. Report 339, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, (1987).
[36] M. Dryja, O. B. Widlund, SIAM J. Sci. Comput. 15 (1994) 604
[37] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, H. Zhang Portable, Extensible Toolkit for Scientific Computation (PETSc) home page, http://www.mcs.anl.gov/petsc, 2009
[38] Online CUBIT User's Manual, http://cubit.sandia.gov/documentation.html, Sandia National Laboratories, 2009
[39] G. Karypis, R. Aggarwal, K. Schloegel, V. Kumar, S. Shekhar, METIS home page, http://wwwusers.cs.umn.edu/karypis/metis/
[40] ParaView homepage. http://www.paraview.org, Kitware and Sandia National Laboratories, USA, 2009
[41] R.L. Burden, J.D. Faires, Numerical Analysis (Thomson Brooks/Cole, Belmont, 2005)