

Parallel Solvers for Flexible Approximation Schemes in Multiparticle Simulation*

Masha Sosonkina¹ and Igor Tsukerman²

¹ Ames Laboratory/DOE, Iowa State University,
Ames, IA 50011, USA
`masha@scl.ameslab.gov`

² Department of Electrical and Computer Engineering,
The University of Akron,
Akron, OH 44325-3904, USA
`igor@uakron.edu`

Abstract. New finite difference schemes with flexible local approximation are applied to screened electrostatic interactions of spherical colloidal particles governed by the Poisson-Boltzmann equation. Local analytical approximations of the solution are incorporated directly into the scheme and yield high approximation accuracy even on simple and relatively coarse Cartesian grids. Several parallel iterative solution techniques have been tested with an emphasis on suitable parallel preconditioning for the nonsymmetric system matrix. In particular, flexible GMRES preconditioned with the distributed Schur Complement exhibits good solution time and scales well when the number of particles, grid nodes or processors increases.

1 Introduction

The study described in this paper has three key ingredients: (1) A new class of finite difference (FD) schemes; (2) Efficient parallel solvers; (3) Applications to the Poisson-Boltzmann equation (PBE) for electrostatics in solvents. These three components are intertwined: the new schemes are particularly suitable for electrostatic fields of dielectric particles (for other applications, see [1, 2]), and their practical use is facilitated greatly by suitable parallel solvers.

Under the mean field theory approximation, the electrostatic potential of multiple charged particles in a solvent is governed by the Poisson-Boltzmann equation (Sect. 2). Several routes are available for the numerical simulation. First, if particle sizes are neglected and the PBE is linearized, the solution is simply the sum of the Yukawa potentials of all particles. If the characteristic length of the exponential field decay (the Debye length) is small, the electrostatic interactions are effectively short-range and therefore relatively inexpensive to compute. For

* The work of M.S. was supported in part by the U.S. Department of Energy under Contract W-7405-ENG-82, NERSC, by NSF under grant NSF/ACI-0305120, and by University of Minnesota Duluth. The work of I.T. was supported in part by the NSF awards 0304453, 9812895 and 9702364.

weak ionic screening (long Debye lengths) Ewald-type methods [3] or the Fast Multipole Method [4] can be used. However, our goal is to develop algorithms that would be applicable to finite-size particles and extendable to nonlinear problems. Ewald-type methods and FMM are not effective in such cases.

An alternative approach is the Finite Element Method (FEM) and the Generalized Finite Element Method (GFEM) [5]. FEM requires very complex meshing and re-meshing even for a modest number of moving particles and quickly becomes impractical when the number of particle grows. In addition, re-meshing is known to introduce a spurious numerical component in force calculation (e.g., [6]). GFEM relaxes the restrictions of geometric conformity in FEM and allows suitable non-polynomial approximating functions to be included in the approximating set. This has been extensively discussed in the literature [7], including our own publications [8, 5, 9]. Unfortunately, GFEM has a substantial overhead due to numerical quadratures and a higher number of degrees of freedom in generalized finite elements around the particles.

A two-grid approach from computational fluid mechanics has been adapted to colloidal simulation: a spherical mesh around each particle and a common Cartesian background grid [10, 11]. The potential has to be interpolated back and forth between the local mesh of each particle and the common Cartesian grid; the numerical loss of accuracy in this process is unavoidable. In contrast, the Flexible Local Approximation MEthod (FLAME) (Sect. 2, [1, 2]) has only one global Cartesian grid but incorporates *local* approximations of the potential near each particle into the difference scheme. The Cartesian grid can remain relatively coarse – on the order of the particle radius or even coarser. This is in stark contrast with classical FD schemes, where the grid size has to be much smaller than the particle radius to avoid the spurious ‘staircase’ effects.

2 Formulation of the Problem

The electrostatic field of charged colloidal particles (‘macroions’) in the solvent is screened by the surrounding microions of opposite charge. If microion correlations are ignored (a good approximation for monovalent ions under normal conditions), the electrostatic potential is known to be governed by the PBE

$$\epsilon_s \nabla^2 u = - \sum_{\alpha} n_{\alpha} q_{\alpha} \exp\left(-\frac{q_{\alpha} u}{k_B T}\right), \quad (1)$$

where summation is over all species of ions present in the solvent, n_{α} is volume concentration of species α in the bulk, q_{α} is the charge of species α ; ϵ_s is the (absolute) dielectric permittivity of the solvent; k_B is the Boltzmann constant, and T is the absolute temperature. The right hand side of (1) reflects the Boltzmann redistribution of microions in the mean field with potential u . Note that inside the colloidal particles the potential simply satisfies the Laplace equation.

If the electrostatic energy $q_{\alpha} u$ is smaller than thermal energy $k_B T$, PBE can be approximately linearized to yield (after taking electroneutrality into account)

$$\nabla^2 u - \kappa^2 u = 0, \quad \text{with } \kappa = (\epsilon_s k_B T)^{-\frac{1}{2}} \left(\sum_{\alpha} n_{\alpha} q_{\alpha}^2 \right)^{\frac{1}{2}}. \quad (2)$$

(This value of the *Debye-Hückel parameter* κ corresponds to linearization about $u = 0$.)¹ Typically, the sources of the electrostatic field are surface charges on the colloidal particles. Standard boundary conditions apply on the particle-solvent interface: the potential is continuous, while the normal component of the displacement vector $\mathbf{D} = -\epsilon \nabla u$ has a jump equal to the surface charge density. The Dirichlet condition at infinity is zero (in practice, the domain boundary is taken sufficiently far away from the particles). Alternative boundary conditions (e.g. periodic) are possible but will not be considered here.

The remainder of the paper deals only with the linearized equation and with the relevant FD schemes and parallel solvers for it. However, the numerical procedure can be extended to the nonlinear PBE by applying the Newton-Raphson-Kantorovich method to the *continuous* problem [1, 2].

Flexible Local Approximation Methods (FLAME). The accuracy of classical Taylor-based FD schemes deteriorates substantially if the solution is not smooth enough (e.g. at material interfaces where one or more components of the field are discontinuous). In FLAME schemes [1, 2], the Taylor expansions of classical FD are replaced with local basis functions that in many important cases provide very accurate and physically meaningful approximation. In the ‘Trefftz’ version of FLAME, these basis functions are chosen as *local* solutions of the underlying differential equation. Various examples ranging from the Schrödinger equation to PBE to electromagnetic scattering are given in [1, 2].

Let r_{α} ($\alpha = 1, 2, \dots, M$) be the node positions of an M -point grid stencil; usually a regular Cartesian grid is used. Further, let ψ_{β} ($\beta = 1, 2, \dots, m$) be a set of *local* approximating functions which (in Trefftz-FLAME) satisfy the underlying differential equation; the number m of these functions is typically equal to $M - 1$. First, let the underlying linear differential equation be homogeneous (no sources) in the vicinity of the stencil. Then the FLAME scheme is a coefficient vector $s \in R^M$ such that $s \in \text{Null}(N^T)$, where matrix N comprises the nodal values of all basis functions on the stencil: $N_{\alpha\beta} = \psi_{\beta}(r_{\alpha})$, $1 \leq \alpha \leq M$, $1 \leq \beta \leq m$. As shown in [2], consistency of FLAME schemes follows directly from the approximation properties of the basis set.

For the electrostatic problem governed by the linearized PBE (2), a FLAME scheme on a global Cartesian grid is obtained in the following way. First, one chooses a suitable grid stencil (standard seven-point grid stencils are used in the numerical experiments reported in Sect. 4; higher-order schemes on expanded stencils will be considered elsewhere. In the vicinity of a given particle, Trefftz-FLAME basis functions – local solutions of the electrostatic problem – can be generated by matching harmonic expansions inside and outside the particle:

$$\psi_{nm}(r, \theta, \phi) = \begin{cases} r^n Y_{nm}(\theta, \phi), & r \leq r_p \\ (a_{nm} j_n(i\kappa r) + b_{nm} n_n(i\kappa r)) Y_{nm}(\theta, \phi), & r \geq r_p \end{cases}, \quad (3)$$

¹ For a systematic account of ‘optimal’ linearization procedures, see [12, 13].

where Y_{nm} are the spherical harmonics, r_p is the radius of the particle, and the coefficients a_{nm} , b_{nm} can be determined from the boundary conditions. The spherical Bessel functions $j_n(i\kappa r)$ and $n_n(i\kappa r)$ in (3) are expressible in terms of hyperbolic sines / cosines and hence relatively easy to work with. The actual expressions for the coefficients a_{nm} , b_{nm} are given in [1]. For the seven-point stencil, one gets a valid FLAME scheme by adopting six basis functions: one ‘monopole’ term ($n = 0$), three dipole terms ($n = 1$) and any two quadrupole harmonics ($n = 2$). Away from the particles, the classical seven-point scheme for the Helmholtz equation is used for simplicity. For inhomogeneous equations, the FLAME scheme is constructed by splitting the potential up into a particular solution $u_f^{(i)}$ of the inhomogeneous equation and the remainder $u_0^{(i)}$ satisfying the homogeneous one [1, 2]. For the linearized PBE, the inhomogeneous part can be taken as the Yukawa potential that satisfies the PBE in the solvent, the Laplace equation (in a trivial way) inside the particle, and the boundary conditions:

$$u_{Yukawa} = \begin{cases} q [4\pi\epsilon_s r_p (\kappa r_p + 1)]^{-1}, & r \leq r_p \\ q \exp(-\kappa(r - r_p)) [4\pi\epsilon_s r (\kappa r_p + 1)]^{-1}, & r \geq r_p \end{cases}. \quad (4)$$

To summarize, the FLAME scheme in the vicinity of charged particles is constructed in the following way: (i) compute the FLAME coefficients for the homogeneous equation; for each grid stencil, this gives the nonzero entries of the corresponding row of the global system matrix; (ii) apply the scheme to the Yukawa potential to get the entry in the right hand side.

3 Parallel Solution Methods

Realistic multiparticle simulations require three-dimensional grids with millions of grid points, which renders direct solvers infeasible. Iterative methods provide a suitable alternative. FLAME matrices have a regular sparsity structure (7-diagonal if the standard 7-point stencil is used) but are not symmetric and not in general diagonally dominant. This increases the importance of parallel preconditioning in making the iterative solution scalable with respect to problem size and the number of processors.

In parallel iterative methods, each processor holds a set of equation and the associated unknowns. The resulting distributed sparse linear system may be solved using techniques similar to those of Domain Decomposition [14]. Three types of variables are distinguished: (1) Interior variables coupled only with local variables; (2) Inter-domain interface variables coupled with external as well as the local ones; and (3) External interface variables that belong to neighboring processors. These external interface variables must be first received from neighboring processor(s) before a distributed matrix-vector product can be completed. Each local vector of unknowns x_i ($i = 1, \dots, p$) is also split into two parts: the sub-vector u_i of interior variables followed by the sub-vector y_i of inter-domain interface variables. The right-hand side b_i is conformally split into sub-vectors f_i and g_i . The local matrix A_i residing in processor i is block-partitioned according to this splitting. The equations assigned to processor i can be written as follows:

$$\begin{pmatrix} B_i & F_i \\ E_i & C_i \end{pmatrix} \begin{pmatrix} u_i \\ y_i \end{pmatrix} + \begin{pmatrix} 0 \\ \sum_{j \in N_i} E_{ij} y_j \end{pmatrix} = \begin{pmatrix} f_i \\ g_i \end{pmatrix}. \quad (5)$$

The term $E_{ij}y_j$ is the contribution to the local equations from the neighboring sub-domain number j , and N_i is the set of sub-domains that are neighbors to sub-domain i .

Additive Schwarz and Schur Complement Preconditioning. Additive Schwarz (AS) procedures are the simplest parallel preconditioners available. They are easily parallelized and incur communications only in the exchange of interface variables, which may be the same as communications during a distributed matrix vector product and must precede the update of the local residual vector. This vector is used as the right-hand side for the local system to find the local update. There are several options for solving the local system: a (sparse) direct solver, a standard preconditioned Krylov solver, or a forward-backward solution associated with an accurate ILU preconditioner [14, 15].

Schur complement (SC) methods iterate on the inter-domain interface unknowns only, implicitly using interior unknowns as intermediate variables. SC systems are derived by eliminating the variables u_i from (5):

$$S_i y_i + \sum_{j \in N_i} E_{ij} y_j = g_i - E_i B_i^{-1} f_i \equiv g'_i, \quad (6)$$

where S_i is the “local” SC, $S_i = C_i - E_i B_i^{-1} F_i$. Equations (6) for all sub-domains i ($i = 1, \dots, p$) constitute a global system of equations $Sy = g'$ involving only the inter-domain interface unknown vectors y_i . Once the global SC system is (approximately) solved, *each* processor computes the u -part of the solution vector by solving the system $B_i u_i = f_i - E_i y_i$ obtained by substitution from (5) [16].

4 Numerical Results

We have used parallel iterative methods as implemented in the pARMS package [17, 18]. The pARMS package contains several state-of-the-art algebraic parallel preconditioning techniques, such as several distributed SC algorithms. For the subdomain solution, pARMS has a wide range of options including the Algebraic Recursive Multilevel Solver (ARMS) [19]. The following is a subset of pARMS preconditioners used here for the numerical experiments: `add_ilu(k)` denotes an AS procedure described in Sect. 3. The local system is solved approximately by applying incomplete LU factorization with level k fill-in (called ILU(k)) as solver. `add_arms` is similar to `add_ilu(k)` but ARMS is used as an approximate solver for local systems. `sch_ilu(k)` consists of solving the (global) SC system, associated with the inter-domain interface variables, with a few iterations of GMRES preconditioned by Block-Jacobi, in which ILU(k) is used locally on inter-domain interface variables. `sch_arms` is the same as `sch_ilu(k)` but ARMS acts as local solver for the Block-Jacobi preconditioner. In the pARMS package, typical input parameters, such as the maximum number of iterations, convergence tolerance,

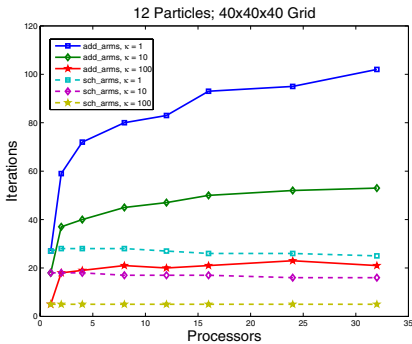


Fig. 1. Iterations for `add_arms` and `sch_arms` and various κ values

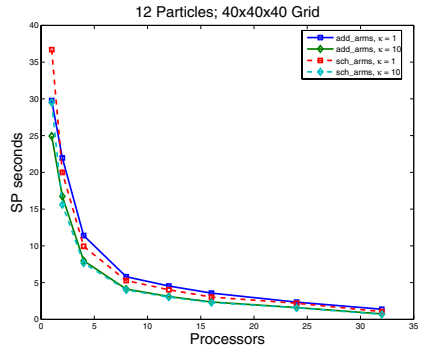


Fig. 2. Timings for `add_arms` and `sch_arms`

and the restart value (for GMRES), are augmented with the parameters that fine-tune the preconditioners. To estimate these parameters, consider that the FLAME matrices for PBE are non-symmetric and the weight of the main diagonal, defined as the *weak diagonal dominance*, depends on the Debye-Hückel parameter κ in (2). For example, for the $60 \times 60 \times 60$ grid the relative weak diagonal dominance is 99% when $\kappa = 10$ but decreases by 10% when $\kappa = 1$. We have investigated the preconditioners described earlier with a different amount of fill governed by the input parameter `lfil`. It means either the level of fill, for the (local) ILU(k) preconditioner or the number of entries to be retained in the L and U factors of ARMS. or the latter, the drop tolerance has been defined as 10^{-3} . The problems have been solved using flexible GMRES (FGMRES) [15] with the restart value 20 and tolerance 10^{-9} . Flexible GMRES allows preconditioners containing inner iterations and thus changing for each outer iteration. In particular, the `sch_arms` preconditioner is set to perform five inner iterations, which is a rather small number taken to accelerate the convergence without large increases of the execution time. The experiments have been performed on the IBM SP RS/6000 supercomputer at NERSC, which has 6,080 processors on 16-way SMP nodes interconnected by the IBM proprietary switching network. Each processor has the peak performance of 1.5 GFlops and shares between 16 and 64 GBytes of main memory.

For a 3D 12-particle problem with 40 grid nodes in each direction, Fig. 1 shows outer iterations to convergence for `add_arms` and `sch_arms` when different κ values are used and `lfil`=25. Although the iteration numbers grow as κ decreases for both `add_arms` and `sch_arms`, the latter exhibits a more scalable behavior as the number of processors increases. Since `sch_arms` performs inner iterations and solves the global system at each step, it is somewhat more costly than `add_arms` (Fig. 2) for a small number of processors. To study the effect of the amount of fill, we have varied the `lfil` parameter for the same twelve-particle problem. Figures 3 and 4 indicate that, even for small κ , decreasing fill-in yields acceptable convergence and may reduce the execution time. Thus for large

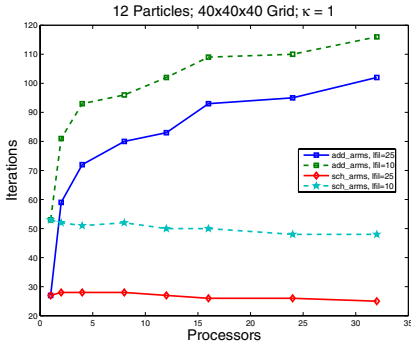


Fig. 3. Iteration numbers of `add_arms` and `sch_arms` for two fill-in amounts

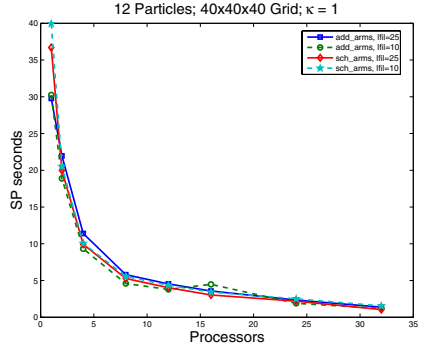


Fig. 4. Timings of `add_arms` and `sch_arms` for two fill-in amounts

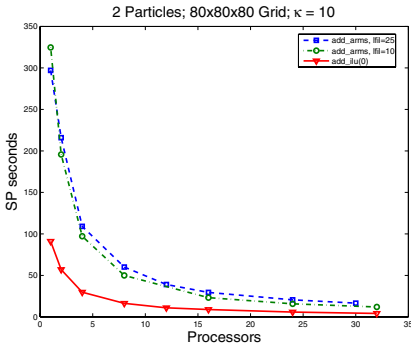


Fig. 5. Timings for a large problem with different fill-in values in `add_arms`

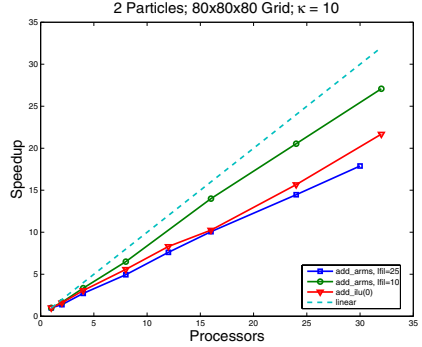


Fig. 6. Speedup of `add_arms` for different fill-in amounts on a large problem

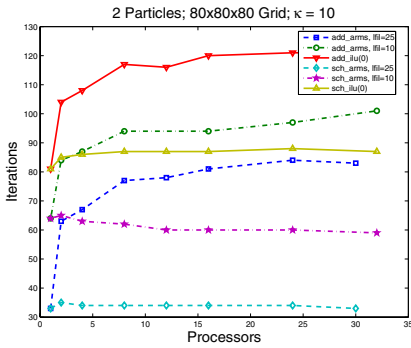


Fig. 7. Iteration numbers for a two-particle large problem with different fill

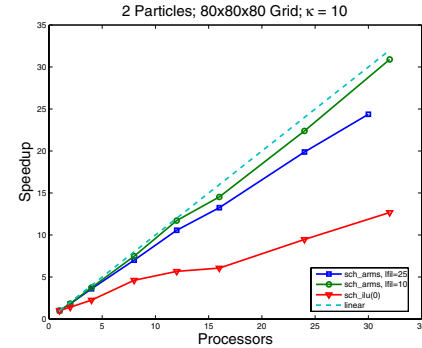


Fig. 8. Speedup of `sch_arms` for different fill-in amounts on a large problem

problem sizes, it may be beneficial to very small fill-in, as in `arms_ilu(0)`, on smaller numbers of processors. Figure 5 presents a comparison of the execution times for a two-particle problem on an 80^3 grid. For AS with different amounts of fill, the speedups are shown in (Fig. 6). Note that the speedup is defined as the ratio of the execution time of the corresponding sequential algorithm to the parallel execution time. Due to a rather drastic increase, by 38% on average, in the number of iterations from sequential to parallel execution for AS (Fig. 7), the best speedup is only 84% of the ideal (linear) case. On the other hand, the speedups for `sch_arms` are almost linear for a sufficient amount of fill (Fig. 8).

5 Conclusion

Several parallel iterative solvers have been applied to linear FLAME systems for PBE, with the goal of finding the best methods and preconditioning techniques in parallel environments for varying problem difficulty and size. It has been observed that Schur Complement preconditioning with a small amount of fill and a few inner iterations scales well and becomes competitive with Additive Schwarz for large number of processors, while attaining almost linear speedup. The number of iterations and the computational time depends only mildly on the Debye parameter. Overall, the parallel distributed Schur Complement solver is promising for the simulation of colloidal suspensions, as well as polymer and protein molecules in solvents.

References

1. Tsukerman, I. *IEEE Trans. Magn.* **41** (2005) 2206–2225
2. Tsukerman, I. *J. Comput. Phys.* **211** (2006) 659–699
3. Salin, G., Caillol, J.M. *J. Chem. Phys.* **113** (2000) 10459–10463
4. Greengard, L.F., Huang, J. *J. Comput. Phys.* **180** (2002) 642–658
5. Plaks, A., Tsukerman, I., Friedman, G., Yellen, B. *IEEE Trans. Magn.* **39** (2003) 1436–1439
6. Tsukerman, I. *IEEE Trans. Magn.* **31** (1995) 1472–1475
7. Strouboulis, T., Babuška, I., Copps, K. *Comput. Meth. Appl. Mech. Eng.* **181** (2000) 43–69
8. Proekt, L., Tsukerman, I. *IEEE Trans. Magn.* **38** (2002) 741–744
9. Basermann, A., Tsukerman, I. Volume LNCS 3402., Springer (2005) 325–339
10. Fushiki, M. *J. Chem. Phys.* **97** (1992) 6700–6713
11. Dobnikar, J., Haložan, D., M.Brumen, von Grünberg, H.H., Rzehak, R. *Comput. Phys. Comm.* **159** (2004) 73–92
12. Deserno, M., von Grünberg, H.H. *Phys. Review E* **66** (2002) 15 pages
13. Bathe, M., Grodzinsky, A.J., Tidor, B., Rutledge, G.C. *J. of Chem. Phys.* **121** (2004) 7557–7561
14. Smith, B., Bjørstad, P., Gropp, W.: *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, New York (1996)
15. Saad, Y.: *Iterative Methods for Sparse Linear Systems*. SIAM (2003)

16. Saad, Y., Sosonkina, M. *SIAM J. Sci. Comput.* **21** (1999) 1337–1356
17. Li, Z., Saad, Y., Sosonkina, M. *Numer. Lin. Alg. Appl.* **10** (2003) 485–509
18. Sosonkina, M., Saad, Y., Cai, X. *Future Gen. Comput. Systems* **20** (2004) 489–500
19. Saad, Y., Suchomel, B. *Numer. Lin. Alg. Appl.* **9** (2002) 359–378