Supporting Information

Finite Difference Formulation

Let $K_0(x_0, y_0, z_0)$ be an arbitrary grid point away from the boundary of the cubical Ω. As shown in Figure 1 of [1], there exists a "unit element" V_0 of volume h^3 which is centered at K_0 such that

$$| x - x_0 | \leq \frac{1}{2} h, \ \ | y - y_0 | \leq \frac{1}{2} h, \ \ | z - z_0 | \leq \frac{1}{2} h \tag{A1}$$

for any point $P(x, y, z)$ inside V_0.

Integrating eqn. (1) over the volume element V_0 yields

$$\int_{V_0} \nabla \bullet (\varepsilon(x) \nabla \phi(x)) dx - \int_{V_0} \kappa(x)^2 \sinh(\phi(x)) dx = -4\pi \int_{V_0} \rho(x) dx. \tag{A2}$$

We define the charge q_0 assigned to K_0 as

$$q_0 = \int_{V_0} \rho(x) dx, \tag{A3}$$

and assume Debye-Huckel parameter $\kappa(x)$ and potential $\phi(x)$ do not change in the volume element V_0, provided the grid spacing h is small. Thus, the second term on the left-hand side of eqn. (A2)

$$\int_{V_0} \kappa(x)^2 \sinh(\phi(x)) dx \approx \kappa(K_0)^2 \sinh(\phi(K_0)) h^3 = \kappa(K_0)^2 \frac{\sinh(\phi(K_0))}{\phi(K_0)} h^3 \phi(K_0). \tag{A4}$$

The first term on the left-hand side of eqn. (A2) can be simplified using Green's Theorem

$$\begin{aligned} \int_{V_0} \nabla \bullet (\varepsilon(x) \nabla \phi(x)) dx &= \int_{\partial V_0} \frac{\partial}{\partial n} (\varepsilon(x) \nabla \phi(x)) ds \\ &= \sum_{\substack{\text{on 6 faces} \\ \text{of } V_0}} \int \frac{\partial}{\partial n} (\varepsilon(x) \nabla \phi(x)) ds. \end{aligned} \tag{A5}$$

The surface integrals in eqn. (A5) can be approximated by differences. For example, on the face of $\partial V_0 |_{x-x_0=\frac{1}{2}h,|y-y_0|\le\frac{1}{2}h,|z-z_0|\le\frac{1}{2}h}$, we have

$$\begin{aligned}
\int \frac{\partial}{\partial n}(\varepsilon(x)\nabla\phi(x))ds \quad &= \int_{|y-y_0|\le\frac{1}{2}h} dy \int_{|z-z_0|\le\frac{1}{2}h} \frac{\partial}{\partial x}(\varepsilon(x_0+\frac{1}{2}h,y,z)\phi(x_0+\frac{1}{2}h,y,z))dz \\
&\approx \varepsilon_{x_0+\frac{1}{2}h,y_0,z_0} \int_{|y-y_0|\le\frac{1}{2}h} dy \int_{|z-z_0|\le\frac{1}{2}h} \frac{\partial}{\partial x}(\phi(x_0+\frac{1}{2}h,y,z))dz \qquad \text{(A6)} \\
&\approx \varepsilon_{x_0+\frac{1}{2}h,y_0,z_0} \frac{\phi_{x_0+h,y_0,z_0}-\phi_{x_0,y_0,z_0}}{h} \int_{|y-y_0|\le\frac{1}{2}h} dy \int_{|z-z_0|\le\frac{1}{2}h} dz \\
&= \varepsilon_{x_0+\frac{1}{2}h,y_0,z_0} \frac{\phi_{x_0+h,y_0,z_0}-\phi_{x_0,y_0,z_0}}{h} h^2 \\
&= h\varepsilon_{x_0+\frac{1}{2}h,y_0,z_0}(\phi_{x_0+h,y_0,z_0}-\phi_{x_0,y_0,z_0})
\end{aligned}$$

where the value of ε is assumed to be constant over the whole face and approximated by its value $\varepsilon_{x_0+\frac{1}{2}h,y_0,z_0}$ at the center of the face, and the partial derivative of ϕ with respect to x is approximated by the forward difference. Substituting the approximates of the surface integrals back into eqn. (A5) yields

$$\int_{V_0} \nabla\bullet(\varepsilon(x)\nabla\phi(x))dx \approx h(\sum_{i=1}^{6}\varepsilon_i\phi_i - \phi_0\sum_{i=1}^{6}\varepsilon_i), \qquad \text{(A7)}$$

where $\phi_i, i=1,...,6$ are potentials at six nearest neighboring girds of K_0, and $\varepsilon_i, i=1,...,6$ are dielecstatic constants at midpoints of K_0 and its nearest neighbors.

Finally, plugging eqn. (A3) – (A7) into eqn. (A2) and rearranging terms result in a iteration equation

$$\phi_0 = \frac{\sum_{i=1}^{6}\varepsilon_i\phi_i + 4\pi q_0 / h}{\sum_{i=1}^{6}\varepsilon_i + (\kappa h)^2(\sinh(\phi_0))/\phi_0}. \qquad \text{(A8)}$$

Approximating the hyperbolic sine function by its Taylor series leads to

$$\phi_0^{'} = \frac{\sum_{i=1}^{6} \varepsilon_i \phi_i + 4\pi q_0 / h}{\sum_{i=1}^{6} \varepsilon_i + (\kappa h)^2 (1 + \phi_0^2 / 3! + \phi_0^4 / 5! ... + \phi_0^{2n} / (2n+1)!)}, \qquad \text{(A9)}$$

which has been shown in [2].

Important Quantities and Theoretical Results in Performance Analysis of Parallel Computing

A core concept in parallel computing is speedup, which compares the execution of the parallel program with its serial cousin. Two types of speedup, absolute and relative, will be defined for completeness of this section but only the absolute speedup will be used later.

Let T_A be the wall clock time of the serial implementation and T_P be the wall clock time of the parallel implementation using P processors. The absolute speedup S_{P_A} is defined to be

$$S_{P_A} = \frac{T_A}{T_P}. \qquad \text{(B1)}$$

The absolute speedup has significant theoretical meaning. However, it is difficult, even impossible, to be measured in the cases that the serial code is absent. Thus, an alternative, the relative speedup, is introduced. Let T_1 be the wall clock time of the code running on a single processor, and T_P be the wall clock time of the parallel implementation over P processors. The relative speedup with respect to one processor is defined as

$$S_P^1 = \frac{T_1}{T_P}. \qquad \text{(B2)}$$

In this work we use absolute speedup for all performance analysis in order to compare the parallel algorithm directly to the fastest serial one. To simplify notations, the absolute speedup is denoted as S, S_P, or just speedup from now on.

The speedup is considered to be linear whenever $S_P \approx P$, and is called super linear whenever $S_P > P$. When neither one of these apply, the speedup is said to be nonlinear.

Linear speedup is the best we can expect for most parallel algorithms. Super linear speedup is usually achieved from the improvement of hardware capability. Very few algorithms are capable of achieving linear, much less super linear speedup, due to the fact that communication between processors contributes more in the overhead and significantly slows down the computation. Typically, good parallel algorithms achieve nearly linear speedup for small number of processors, which flattens out for large number of processors.

The efficiency of an algorithm is another primary quantity for performance analysis and is defined as

$$E_P = \frac{S_P}{P}. \tag{B3}$$

Thus, efficiency is speedup per processor. It estimates how well-utilized the processors are in solving the problem, compared to how much effort is expended in communication and synchronization. From its definition, it is clear that efficiency always stays between 0 and 1. Linear speedup corresponds to the highest efficiency $E_P = 1$. Efficiency close to 0 indicates that most effort of processors is wasted in communication and synchronization.

The Amdahl's law [3] is one important theoretical result about the performance of parallel algorithms. It states the speedup S_P, given P processors, is

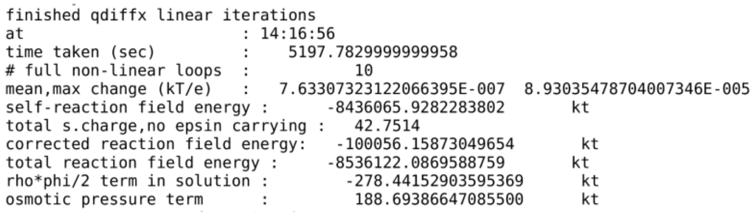
$$S_P = \frac{1}{f + (1 - f)/P}, \tag{B4}$$

where f denote the sequential fraction of the computation. Simple mathematical arguments on eqn. (B4) lead to useful insights of the Amdahl's law. For example, a perfectly-linear parallelizable algorithm is one in which f tends to 0, since $\lim_{f \to 0} S_P = P.$

Similarly, the maximum speedup is limited by f^{-1}, since $\lim_{P\to\infty} S_P = \frac{1}{f}$. A good example representing the Amdahl's law can be found in [4].

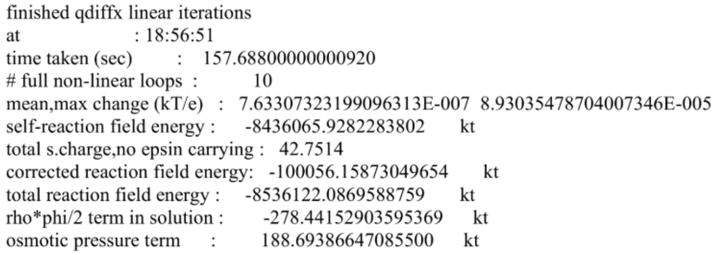
Readers who are interested in parallel computing are directed to [5] for other quantities and theoretical results which are omitted here.

Evidence that the Parallel Implementation is Exact



```
finished qdiffx linear iterations
at                        : 14:16:56
time taken (sec)          :    5197.7829999999958
# full non-linear loops   :           10
mean,max change (kT/e)    :   7.63307323122066395E-007  8.93035478704007346E-005
self-reaction field energy :      -8436065.9282283802       kt
total s.charge,no epsin carrying :   42.7514
corrected reaction field energy:   -100056.15873049654       kt
total reaction field energy :     -8536122.0869588759       kt
rho*phi/2 term in solution :        -278.44152903595369       kt
osmotic pressure term      :         188.69386647085500       kt
```

(a)



```
finished qdiffx linear iterations
at                  : 18:56:51
time taken (sec)        :    157.68800000000920
# full non-linear loops  :           10
mean,max change (kT/e)   :   7.63307323199096313E-007  8.93035478704007346E-005
self-reaction field energy :      -8436065.9282283802       kt
total s.charge,no epsin carrying :   42.7514
corrected reaction field energy:   -100056.15873049654       kt
total reaction field energy :     -8536122.0869588759       kt
rho*phi/2 term in solution :        -278.44152903595369      kt
osmotic pressure term      :        188.69386647085500      kt
```

(b)

Figure S1. Snap-shots of the output screen of serial (a) and parallelized (b) DelPhi. One can see that the output potentials and energies are identical, while the execution time is much shorter for the parallelized DelPhi.

References

1. Klapper, I.; Hagstrom, R.; Fine, R.; Sharp, K.; Honig, B. Proteins: Structure, Function, and Bioinformatics 1986, 1(1), 47-59.
2. Jayaram, B.; Sharp, K. A.; Honig, B. Biopolymers 1989, 28(5), 975-993.
3. Amdahl, G. M. In Proceedings of the April 18-20, 1967, spring joint computer conference; ACM: Atlantic City, New Jersey, 1967, p 483-485.
4. Wikipedia. Amdahl's law, http://en.wikipedia.org/wiki/Amdahl%27s_law#cite_note-0, (accessed April, 2012).

5. Li, C., A TIME-AND-SPACE PARALLELIZED ALGORITHM FOR THE CABLE EQUATION, Ph.D. Thesis. University of Tennessee, Knoxville, TN, August 2011.