Parallel Computing the Minimized Molecular Surface of Macromolecules and Large Complexes

Ryan Moser, Chuan Li

Department of Mathematics, West Chester University of Pennsylvania

Abstract

Constructing the molecular surface is the first crucial stage to produce accurate calculations of electrostatic potentials and energies for the molecules immersed in the water phase. When the macromolecules and complexes consist of hundreds of thousands of charged atoms, three-dimensional surface construction becomes unbearably slow due to the high computational cost. In this project, we propose a novel grid-based computing scheme to generate the Minimized Molecular Surface (MMS) by effectively utilizing the computational power of multiple CPUs on a computing cluster via the Message Passing Interface (MPI) library. The resulting parallel code will be tested against its sequential version to demonstrate its efficiency and accuracy, and the code will be released to the computational biophysics society free-of-charge for academic usage.

Minimized Molecular Surfaces and Electrostatic Potential

The **Minimized Molecular Surface (MMS)** [2] is constructed by minimizing a total free energy functional

$$G_{total} = \int_{\Omega} \left(\gamma \|\nabla S\| + pS + \rho_0 (1-S) U^{VdW} + S\left[\rho_m \phi - \frac{\epsilon_m}{2}\|\nabla\phi\|^2\right] + (1-S)\left[-\frac{\epsilon_s}{2}\|\nabla\phi\|^2 - k_B T \sum_{j=1}^{N_c} c_j \left(e^{-\frac{q_j\phi}{k_B T}} - 1\right)\right]\right) d\boldsymbol{r} \quad (1)$$

with respect to the electrostatic function $\phi(r)$ and the hypersurface function $S(r)$, $\frac{\partial G}{\partial \phi} = 0$, and $\frac{\partial G}{\partial S} = 0$, and yielding a **generalized Poisson–Boltzmann equation** for the electrostatic potential $\phi(r)$

$$\nabla \cdot (\epsilon(S)\nabla\phi) + (1-S)\sum_{j=1}^{N_c} c_j q_j e^{-\frac{q_j\phi}{k_B T}} = -S\rho_m, \quad (2)$$

and a **generalized Laplace–Beltrami equation** for the hypersurface function $S(r)$

$$-\nabla \cdot \left(\gamma \frac{\nabla S}{\|\nabla S\|}\right) + p - \rho_0 U^{VdW} + \left[\rho_m \phi - \frac{\epsilon_m}{2}\|\nabla\phi\|^2\right] + \left[\frac{\epsilon_s}{2}\|\nabla\phi\|^2 + k_B T \sum_{j=1}^{N_c} c_j \left(e^{-\frac{q_j\phi}{k_B T}} - 1\right)\right] = 0, \quad (3)$$

Where γ is the surface tension, p is the hydrodynamic pressure, ρ_0 is the solvent bulk density, $U^{VdW}(\mathbf{r})$ is the VdW potential, $\rho_m(\mathbf{r})$ is the canonical density of molecular free charges, $\phi(\mathbf{r})$ is the electrostatic potential, ϵ_m is the electric permittivity of the macromolecule, ϵ_s is the electric permittivity of the solvent, K_B is the Boltzmann constant, T is the temperature, c_j is the bulk concentration of jth ionic species, q_j is the charge of the jth ionic species, and N_c is the number of ionic species. Solution to eqn. (3) is equivalent to the solution to the partial differential equation

$$\frac{\partial S}{\partial t}(r,t) = \frac{\partial}{\partial x}\left(\beta(S)\frac{\partial S}{\partial x}\right) + \frac{\partial}{\partial y}\left(\beta(S)\frac{\partial S}{\partial y}\right) + \frac{\partial}{\partial z}\left(\beta(S)\frac{\partial S}{\partial z}\right) + V(\phi; r) \quad (4)$$

in the equilibrium state, where the conductivity coefficient $\beta(S)$ is given by $\beta = \frac{1}{\sqrt{S^2{}_x + S^2{}_y + S^2{}_z + \eta}}$, $\eta = 10^{-7}$, and the generalized potential is given by

$$V(\phi; r) = \frac{1}{\gamma}\left[-p + \rho_0 U^{VdW} - \rho_m \phi + \frac{\epsilon_m}{2}\|\nabla\phi\|^2 - \frac{\epsilon_s}{2}\|\nabla\phi\|^2 - k_B T \sum_{j=1}^{N_c} c_j \left(e^{-\frac{q_j\phi}{k_B T}} - 1\right)\right].$$

Numerical Methods/Derivations

- **Temporal Discretization.**

An Alternating Direction Implicit (ADI) method is utilized to discretize the temporal domain

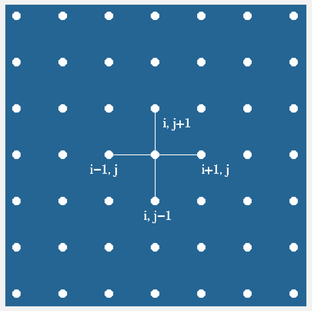
$$\begin{aligned}
\left(1 - \frac{\Delta t}{2}\delta_{xx}\right) S^*_{i,j,k} &= \left[1 + \frac{\Delta t}{2}(\delta_{xx} + 2\delta_{yy} + 2\delta_{zz})\right] S^n_{i,j,k} + \Delta t V^{n+\frac{1}{2}}_{i,j,k}, \\
\left(1 - \frac{\Delta t}{2}\delta_{yy}\right) S^{**}_{i,j,k} &= S^*_{i,j,k} - \frac{\Delta t}{2}\delta_{yy} S^n_{i,j,k}, \\
\left(1 - \frac{\Delta t}{2}\delta_{zz}\right) S^{n+1}_{i,j,k} &= S^{**}_{i,j,k} - \frac{\Delta t}{2}\delta_{zz} S^n_{i,j,k},
\end{aligned} \quad (5)$$

where δ_{xx}, δ_{yy}, and δ_{zz} are the second order finite difference operators in *x*-, *y*-, and *z*-direction, respectively.

- **Spatial Discretization**

Above spatial operators are approximated by

$$\begin{aligned}
\left[\frac{\partial}{\partial x}\left(\beta\frac{\partial S}{\partial x}\right)\right]_{i,j,k} &\approx \frac{1}{h^2}\left(\beta_{i+\frac{1}{2},j,k}\left(S_{i+1,j,k} - S_{i,j,k}\right) - \beta_{i-\frac{1}{2},j,k}\left(S_{i,j,k} - S_{i-1,j,k}\right)\right), \\
\left[\frac{\partial}{\partial y}\left(\beta\frac{\partial S}{\partial y}\right)\right]_{i,j,k} &\approx \frac{1}{h^2}\left(\beta_{i,j+\frac{1}{2},k}\left(S_{i,j+1,k} - S_{i,j,k}\right) - \beta_{i,j-\frac{1}{2},k}\left(S_{i,j,k} - S_{i,j-1,k}\right)\right), \\
\left[\frac{\partial}{\partial z}\left(\beta\frac{\partial S}{\partial z}\right)\right]_{i,j,k} &\approx \frac{1}{h^2}\left(\beta_{i,j,k+\frac{1}{2}}\left(S_{i,j,k+1} - S_{i,j,k}\right) - \beta_{i,j,k-\frac{1}{2}}\left(S_{i,j,k} - S_{i,j,k-1}\right)\right),
\end{aligned} \quad (6)$$



Where, for example, $\beta_{i\pm\frac{1}{2},j,k} = \left[\frac{1}{\sqrt{S^2{}_x + S^2{}_y + S^2{}_z + \eta}}\right]_{i\pm\frac{1}{2},j,k}$. It can be effectively calculated by

$$\left(S^2_x\right)^n_{i\pm\frac{1}{2},j,k} \approx \left(\frac{S^n_{i+1,j,k} - S^n_{i,j,k}}{h}\right)^2$$

$$\left(S^2_y\right)^n_{i\pm\frac{1}{2},j,k} \approx \left(\frac{S^n_{i,j+1,k} - S^n_{i,j-1,k}}{4h} + \frac{S^n_{i+1,j+1,k} - S^n_{i+1,j-1,k}}{4h}\right)^2$$

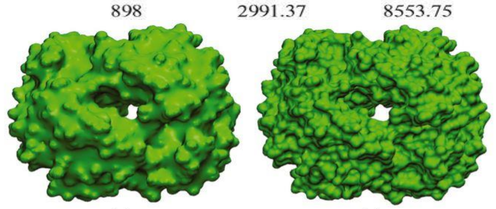
$$\left(S^2_z\right)^n_{i\pm\frac{1}{2},j,k} \approx \left(\frac{S^n_{i,j,k+1} - S^n_{i,j,k-1}}{4h} + \frac{S^n_{i+1,j,k+1} - S^n_{i+1,j,k-1}}{4h}\right)^2$$

Biological Simulations

The chart to the left details CPU-time (s) consumed creating the MMS of a molecule listed below along with the number of atoms, and the area ($Å^2$) and volume($Å^3$) of the resulting molecular surface. The two surfaces pictured are a comparison between an MMS generated surface using the ADI algorithm and the Solvent Exclusive Surface (SES) (the solvent excluded domain of the VdW surface) generated by the MSMS package for two molecules. (B-DNA double helix segment PDB ID: 425D and another molecule, hemoglobin, PDB ID: 1HGA). This was done with mesh spacing $h = 0.5Å, \Delta t = 0.1\ until\ T = 10$. The MMS was extracted at S=.98 and the SES generated by the MSMS package used probe 1.5Å. While the times to generate the surfaces of the given molecules was almost insignificant, one must consider the time it would take to generate the surface of molecular complexes of millions of atoms. The computational time seems to be linearly related to the number of atoms so one could expect a computational time of around 15 minutes for a molecule with $1.2 * 10^6$ atoms which is far from a maximum. This gives rise to the desire to parallelize the code that generates the molecular surfaces potentially decreasing the computational time by orders of 10.

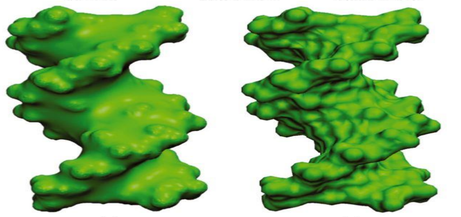
PDB ID	No. of atoms	Area	Volume	CPU
1ajj	519	2038.11	4972.75	0.43
2erl	573	2142.13	5581.13	0.41
1bbl	576	2434.84	5615.00	0.59
1vii	596	2309.51	5401.63	0.40
1cbn	648	2204.76	6088.25	0.51
2pde	667	2534.55	6411.00	0.55
1sh1	702	2493.44	6889.13	0.54
1fca	729	2505.68	7337.00	0.49
1ptq	795	2681.86	7590.00	0.62
1uxc	809	2637.54	7243.13	0.57
1fxd	824	2739.86	8245.13	0.62
1bor	832	2687.56	7607.50	0.66
1hpt	858	3069.32	8146.75	0.63
1bpi	898	2991.37	8553.75	0.72

PDB ID	No. of atoms	Area	Volume	CPU
1mbg	903	2837.37	8368.50	0.57
1r69	997	2835.00	9304.25	0.63
1neq	1187	4372.92	11461.13	1.24
451c	1216	3733.98	11864.13	0.74
1a2s	1272	3938.48	12583.88	0.92
1svr	1435	4226.87	12844.00	1.05
1frd	1478	4047.54	14235.88	0.91
1r63	2065	6430.95	19286.75	1.64
1a7m	2809	6952.82	25650.00	1.99
1beb	4972	11269.56	48149.63	3.65
1vng	8808	16869.10	86000.25	5.33
1tas	12636	22731.36	122470.88	7.43
1maa	33257	57205.56	322510.13	46.27
4bfl	46160	60314.91	461149.75	30.51



(a) (b)

Figure 6. The minimal molecular surface (a) and the solvent-excluded surface (b) of the hemoglobin (PDB ID: 1hga).



(a) (b)

Figure 5. The minimal molecular surface (a) and the solvent-excluded surface (b) of a B-DNA double helix segment (PDB ID: 425D).

Parallel Computing Techniques

- **Spatial Domain Decomposition**

The most straightforward parallel computing technique consider decompose a finite spatial domain of $\mathbb{R}^3$ equally among the available CPU's so that each CPU can work on it's own part of the domain. Each piece of the subdomain with which a CPU is working will have some overlap with another CPU in each direction and this overlap needs to be synchronized. [1]

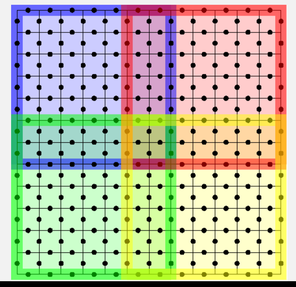
- **Temporal Domain Decomposition**

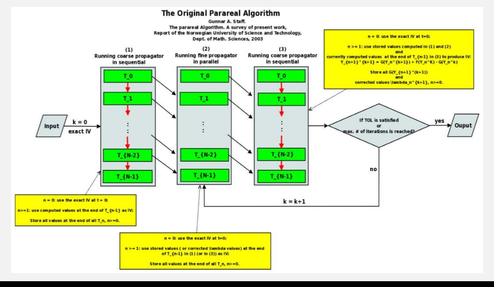
The **Parareal algorithm** is an effective parallel computing scheme which allows the temporal domain to be divided into time slices and the calculations on each slice can be carried out on various CPUs at the same time. Two solvers, one coarse (G) and one fine (F), are required in the scheme and work together in the prediction-and-correction style formula

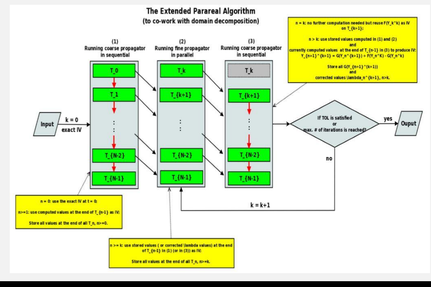
$$U^{k+1}_{n+1} = G\left(t_{n+1}; t_n, U^{k+1}_n\right) + F(t_{n+1}; t_n, U^k_n) - G(t_{n+1}; t_n, U^k_n) \quad (7)$$

- **Time-and-space Domain Decomposition**

The **Extended Parareal Algorithm** is an adaptation of the original Parareal Algorithm that has removed its redundant computations. In the Extended Parareal algorithm, spatially-parallelized solvers are well incorporated into the framework of the Parareal algorithm in order to achieve both time and space parallel calculations. [3]







References

[1] C. Li, L. Li, J. Zhang, E. Alexov "Highly Efficient and Exact Method for Parallelization of Grid-Based Algorithms and its Implementation in DelPhi." *Journal of Computational Chemistry* (2012) 33, 1960–1966. DOI: 10.1002/jcc.23033

[2] W. Tian and Shan Zhao "A Fast Alternating Direction Implicit Algorithm for Geometric Flow Equations in Biomolecular Surface Generation." *International Journal of Numerical Methods in Biomedical Engineering.* (2014) 30, 490–516 DOI: 10.1002/cnm.2613

[3] C. Li and V Alexiades (2016) A Time-and-Space Parallel Scheme on the Cable Equation. PhD Thesis