Conference on Fast Direct Solvers, Purdue University, October 23-24, 2021

Fast Summation Methods based on Barycentric Lagrange Interpolation

Robert Krasny, University of Michigan

collaborators

Lei Wang, University of Wisconsin-Milwaukee

Svetlana Tlupova, Farmingdale State College, SUNY

Nathan Vaughn, LANL

Leighton Wilson, Northrop Grumman

Weihua Geng, Southern Methodist University

Extreme Science and Engineering Discovery Environment

MICHIGAN INSTITUTE FOR COMPUTATIONAL DISCOVERY & ENGINEERING UNIVERSITY OF MICHIGAN

motivation : Poisson equation

$$-\nabla^2\phi = 4\pi\rho \;\Rightarrow\; \phi(\mathbf{x}) = \int \frac{\rho(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\,d\mathbf{y}$$

$$\phi(\mathbf{x}_i) \approx \sum_{j=1}^{N} \frac{q_j}{|\mathbf{x}_i - \mathbf{y}_j|}\;,\; i = 1 : N$$

- direct sum is well suited for parallelization, but cost is still $O(N^2)$

tree-based fast summation methods

DTT (Appel 1985)

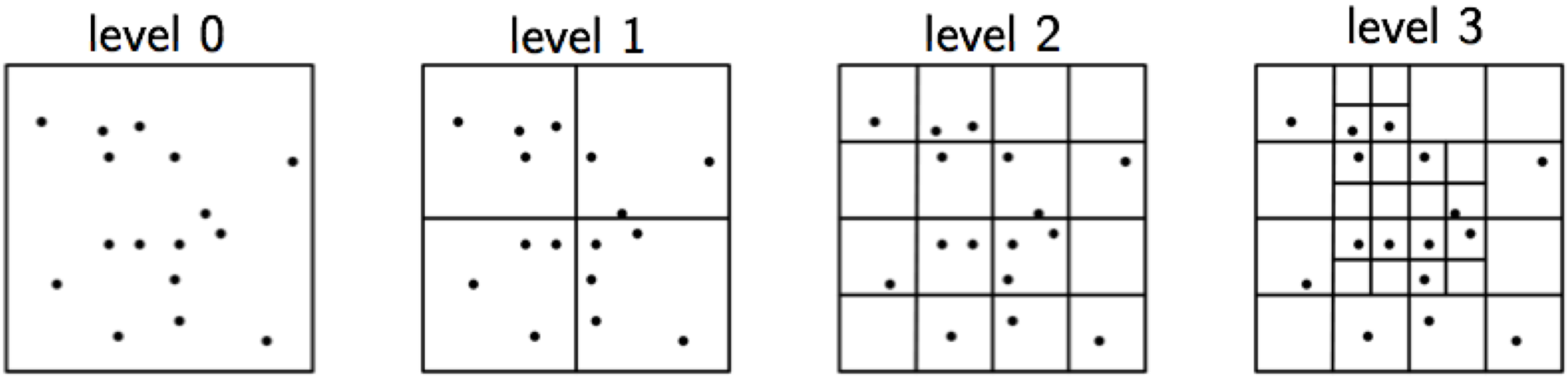
TC (Barnes-Hut 1986)

FMM (Greengard-Rokhlin 1987)

... many variations

tree-based fast summation (1/2)

$$\phi_i = \sum_{j=1}^{N} \frac{q_j}{|\mathbf{x}_i - \mathbf{y}_j|} \ , \ i = 1 : N$$



particle-particle : $O(N^2)$

$\downarrow$

particle-cluster : $O(N \log N)$

$\downarrow$

cluster-cluster : $O(N)$

tree-based fast summation (2/2)

separated kernel approximation

$$\frac{1}{|\mathbf{x}-\mathbf{y}|} \approx \sum_{k=1}^{n} a_k(\mathbf{x}) \cdot b_k(\mathbf{y})$$

- multipole expansion : FMM (Greengard-Rokhlin 1987)

$$\frac{1}{|\mathbf{x}-\mathbf{y}|} \approx \sum_{\ell=0}^{n} \sum_{m=-\ell}^{\ell} r_x^{-(\ell+1)} Y_\ell^m(\theta_x, \phi_x) \cdot r_y^\ell Y_\ell^{-m}(\theta_y, \phi_y)$$

kernel-independent methods

- equivalent density : KIFMM (Ying-Biros-Zorin 2003)
- polynomial interpolation

$$G(\mathbf{x}, \mathbf{y}) \approx \sum_{k_1=0}^{n} \sum_{k_2=0}^{n} \sum_{k_3=0}^{n} G(\mathbf{x}, \mathbf{s_k}) L_{k_1}(y_1) L_{k_2}(y_2) L_{k_3}(y_3)$$

bbFMM (Fong-Darve 2009) : Chebyshev polynomial form

BLTC (Wang-Tlupova-K 2020) : barycentric Lagrange form

polynomial interpolation in 1D

given $f(x)$ on $[-1,1]$, $\{s_i, i=0:n\}$: Chebyshev points of 2nd kind

Lagrange form

$$p(x) = \sum_{k=0}^{n} L_k(x) f_k \ , \ L_k(x) = \frac{\prod_{\substack{j=0\\j\neq k}}^{n}(x-s_j)}{\prod_{\substack{j=0\\j\neq k}}^{n}(s_k-s_j)} \ , \ k=0:n$$

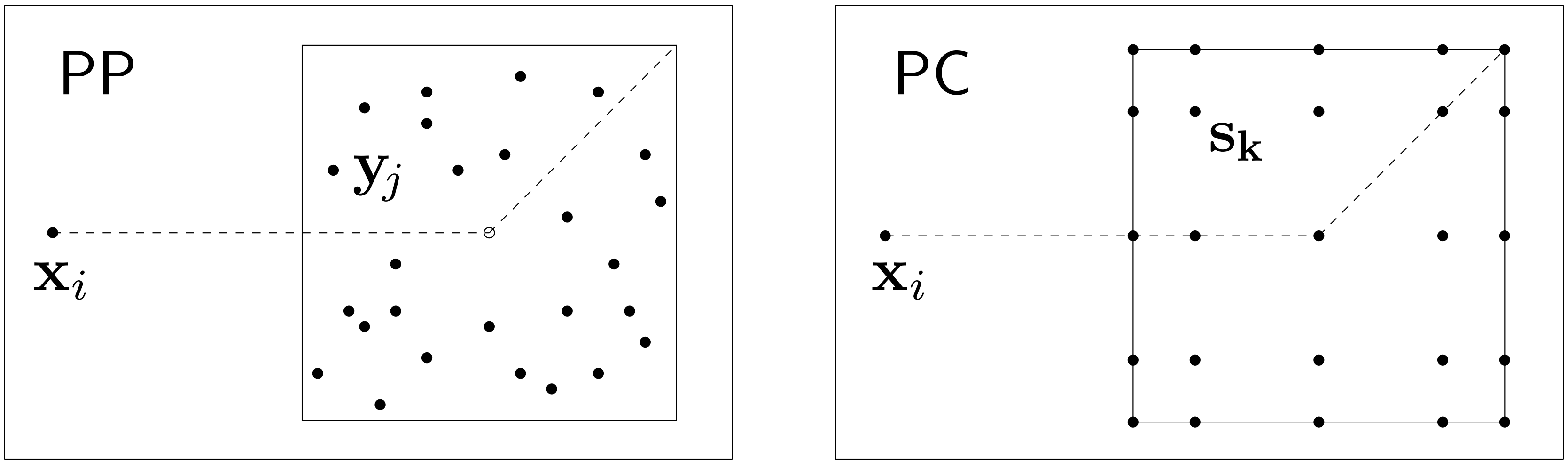
barycentric Lagrange form : Berrut-Trefethen (2004 SIREV)

$$L_k(x) = \frac{\dfrac{w_k}{x-s_k}}{\displaystyle\sum_{j=0}^{n}\frac{w_j}{x-s_j}} \ , \ w_k = (-1)^k \delta_k \, , \ \delta_k = \begin{cases} 1 & , \ k = 1:n-1 \\ 1/2 & , \ k = 0, n \end{cases}$$

"Barycentric interpolation is a variant of Lagrange polynomial interpolation that is fast and stable. It deserves to be known as the standard method of polynomial interpolation."

barycentric Lagrange treecode : BLTC

$$\phi_i = \sum_{j=1}^{N} G(\mathbf{x}_i, \mathbf{y}_j) q_j = \sum_{C} \sum_{\mathbf{y}_j \in C} G(\mathbf{x}_i, \mathbf{y}_j) q_j \to \begin{cases} C_{\text{near}} : \text{PP} \\ C_{\text{far}} \;\; : \text{PC} \end{cases}$$



$$\text{PP} : \sum_{\mathbf{y}_j \in C} G(\mathbf{x}_i, \mathbf{y}_j) q_j \approx \sum_{\mathbf{k}} G(\mathbf{x}_i, \mathbf{s}_\mathbf{k}) \underbrace{\widehat{q}_\mathbf{k}}_{\uparrow \; \sum_{\mathbf{y}_j \in C} \Pi_{\ell=1}^{3} L_{k_\ell}(y_{j\ell}) q_j} : \text{PC} \leftarrow \text{note direct sum form}$$

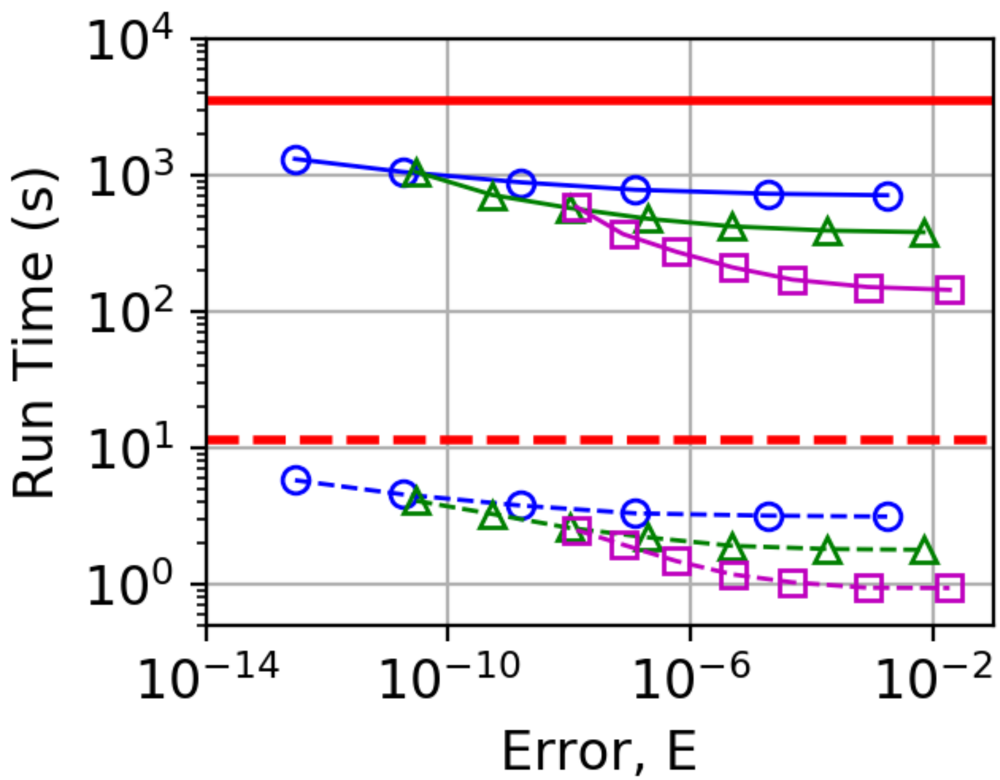
use PC $\Leftrightarrow \mathbf{x}_i \,, C$: well-separated $\Leftrightarrow r/R < \theta$: MAC $\Rightarrow O(N \log N)$

Wang-K-Tlupova, Commun. Comput. Phys. 28 (2020)

BLTC is well suited for GPU acceleration

example : $N = 10^6$, $G(\mathbf{x},\mathbf{y}) = \dfrac{1}{|\mathbf{x}-\mathbf{y}|}$

degree : $n = 1:2:13$, MAC : $\theta = 0.5\,, 0.7\,, 0.9$



6 CPU cores + OpenMP
2.67 GHz Intel Xeon X5650

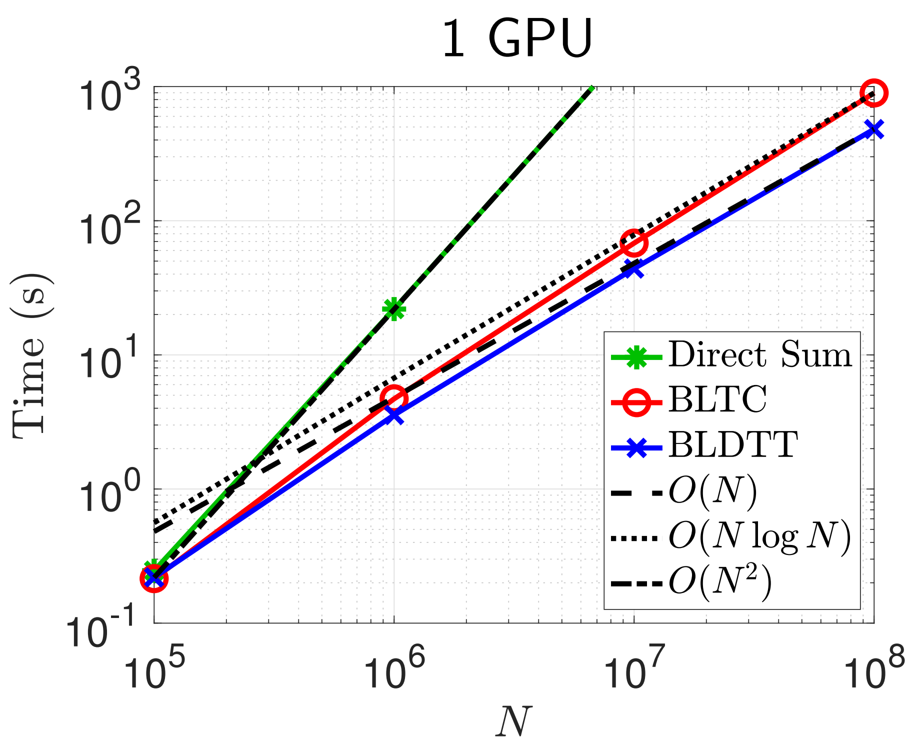
1 GPU + OpenACC
NVIDIA Titan V

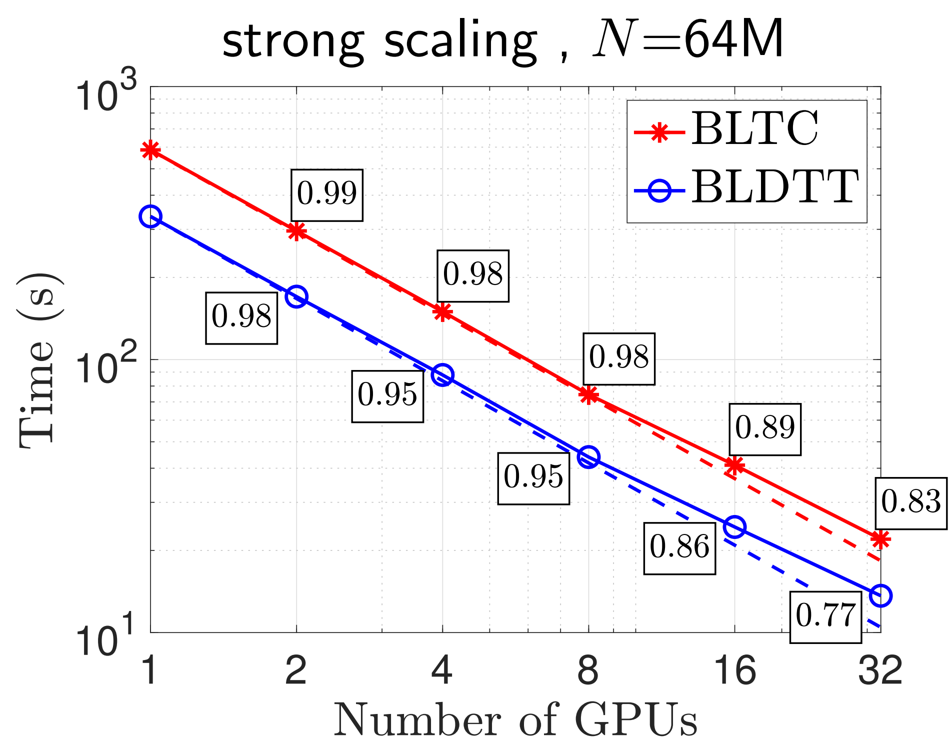
Vaughn-Wilson-K, IEEE IPDPSW (2020)

new : BLDTT $= O(N)$

- FMM upward/downward pass adapted to BLI
- interaction lists by dual tree traversal (Appel 1985)
- PP, PC, CP, CC

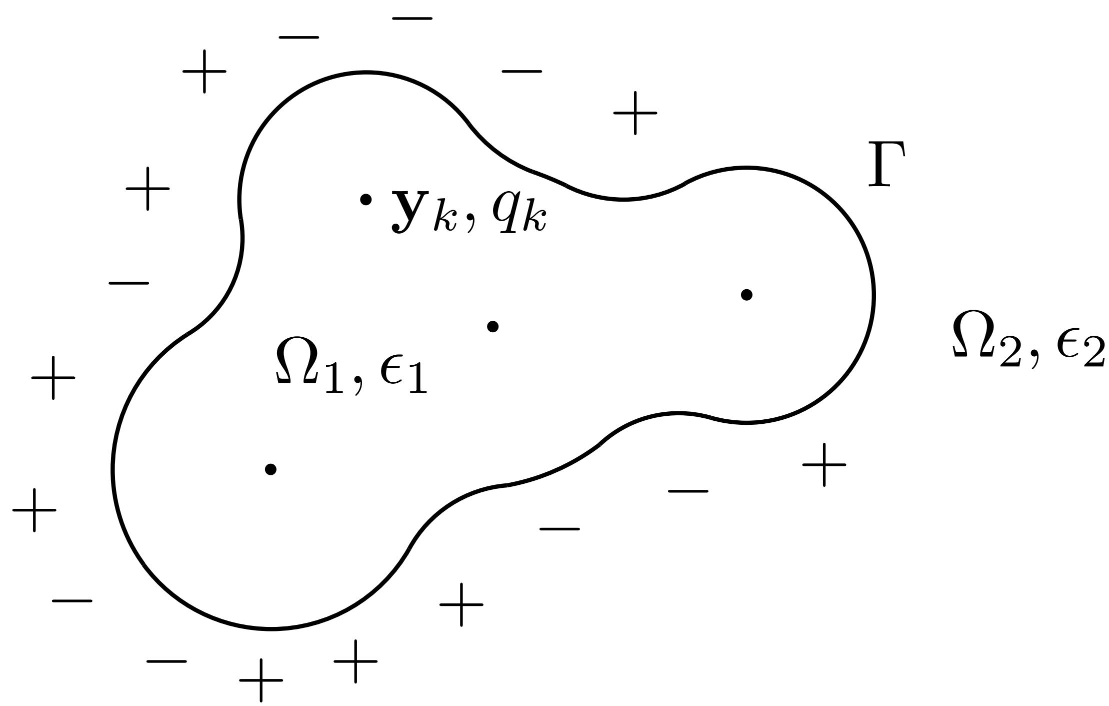
example : $G(\mathbf{x},\mathbf{y}) = \dfrac{1}{|\mathbf{x}-\mathbf{y}|}$, 7-8 digit accuracy





Wilson-Vaughn-K, Comput. Phys. Commun. 265 (2021)

application : electrostatics of biomolecules



biomolecule Ω_1 : $-\epsilon_1 \nabla^2 \phi(\mathbf{x}) = \sum_{k=1}^{N_a} \delta(\mathbf{x} - \mathbf{y}_k) q_k$: Poisson

solvent Ω_2 : $-\epsilon_2 \nabla^2 \phi(\mathbf{x}) + \bar{\kappa}^2 \phi(\mathbf{x}) = 0$: Poisson-Boltzmann

molecular surface Γ : $\phi_1 = \phi_2$, $\epsilon_1 \partial_n \phi_1 = \epsilon_2 \partial_n \phi_2$

far-field boundary condition : $\phi(\mathbf{x}) \to 0$ as $|\mathbf{x}| \to \infty$

convert to integral form and solve by BEM

integral form (Juffer et al. 1991)

$$G_0(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi|\mathbf{x}-\mathbf{y}|}\ ,\ G_\kappa(\mathbf{x},\mathbf{y}) = \frac{e^{-\kappa|\mathbf{x}-\mathbf{y}|}}{4\pi|\mathbf{x}-\mathbf{y}|}\ ,\ \kappa^2 = \frac{\bar{\kappa}^2}{\epsilon_2}\ ,\ \epsilon = \frac{\epsilon_2}{\epsilon_1}$$

$$\frac{1+\epsilon}{2}\phi(\mathbf{x}) = \int_\Gamma \Big[K_1(\mathbf{x},\mathbf{y})\partial_n\phi(\mathbf{y}) + K_2(\mathbf{x},\mathbf{y})\phi(\mathbf{y})\Big]dS_\mathbf{y} + S_1(\mathbf{x})$$

$$\frac{1+\epsilon^{-1}}{2}\partial_n\phi(\mathbf{x}) = \int_\Gamma \Big[K_3(\mathbf{x},\mathbf{y})\partial_n\phi(\mathbf{y}) + K_4(\mathbf{x},\mathbf{y})\phi(\mathbf{y})\Big]dS_\mathbf{y} + S_2(\mathbf{x})$$

$$K_1 = G_0 - G_\kappa\ ,\ K_4 = \partial^2_{n_\mathbf{x} n_\mathbf{y}}(G_\kappa - G_0)$$

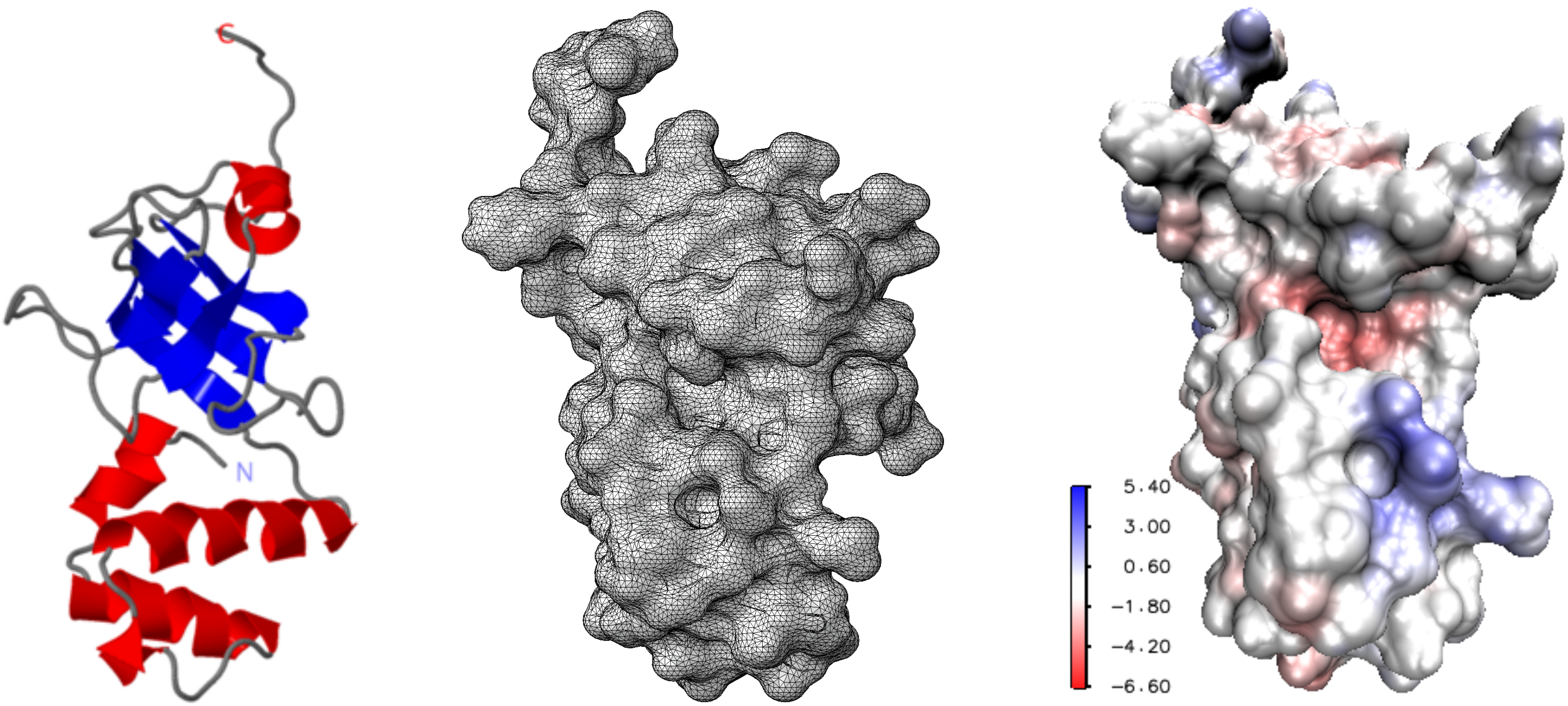
$$K_2 = \epsilon\,\partial_{n_\mathbf{y}}G_\kappa - \partial_{n_\mathbf{y}}G_0\ ,\ K_3 = \partial_{n_\mathbf{x}}G_0 - \epsilon^{-1}\,\partial_{n_\mathbf{x}}G_\kappa$$

$$S_1(\mathbf{x}) = \frac{1}{\epsilon_1}\sum_{k=1}^{N_a} G_0(\mathbf{x},\mathbf{y}_k)q_k\ ,\ S_2(\mathbf{x}) = \frac{1}{\epsilon_1}\sum_{k=1}^{N_a}\partial_{n_x}G_0(\mathbf{x},\mathbf{y}_k)q_k$$

BEM

- triangulate Γ by NanoShaper (Decherchi, Rocchia)
- discretize integrals by node-patch (Lu)
- solve for $\phi\,, \partial_n \phi$ on Γ by GMRES
- compute matrix-vector product by BLDTT on 1 GPU

example : RNA binding domain of E. coli ρ-factor, $N_a = 2069$

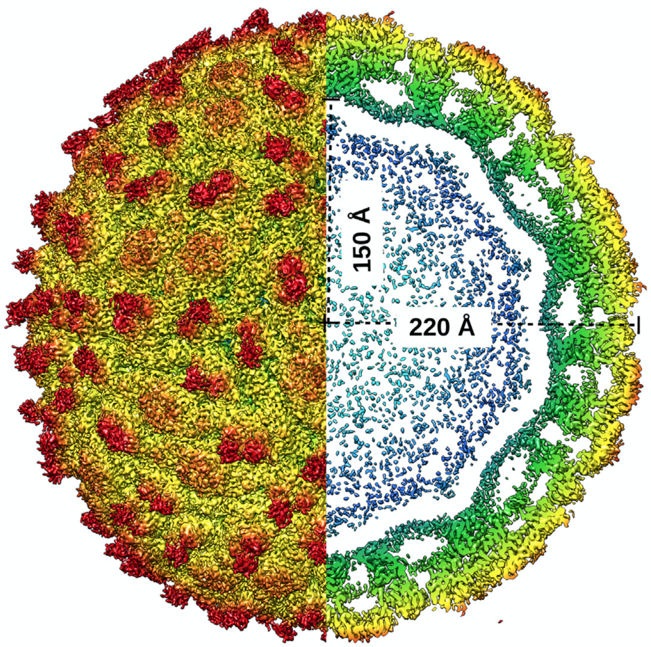


scale = 3, $N \approx 10^5$, run time = 110 s, error ≈ 1%

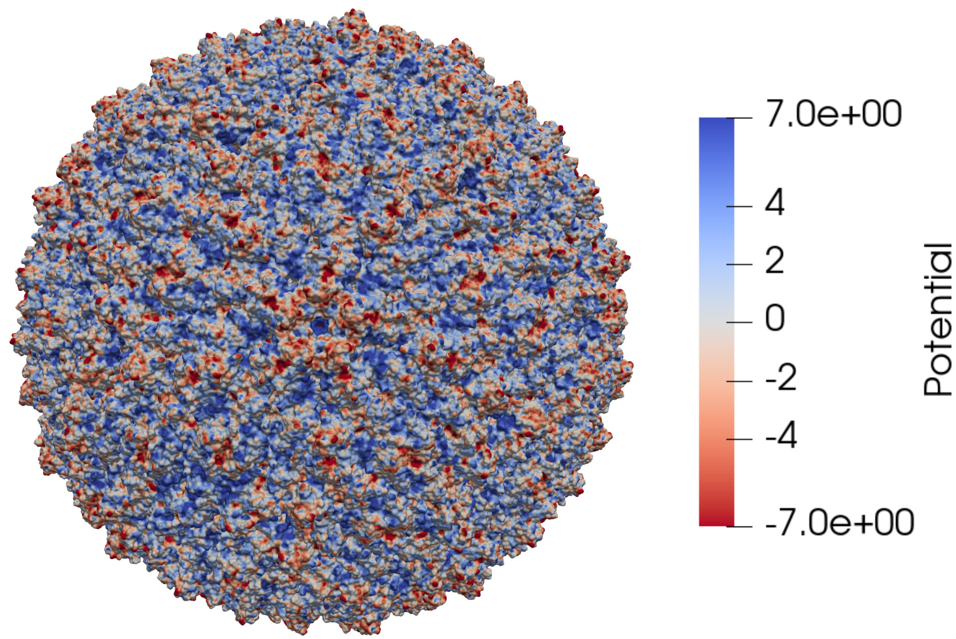
BEM

- triangulate Γ by NanoShaper (Decherchi, Rocchia)
- discretize integrals by node-patch (Lu)
- solve for $\phi\,, \partial_n \phi$ on Γ by GMRES
- compute matrix-vector product by BLDTT on 1 GPU

example : Zika virus, $N_a = 13248$



Sevanna et al. Structure (2018)



scale $= 1.5$, $N \approx 10^7$

run time $= 1689$ s, error $\approx$ tbd

Wilson-Geng-K in preparation

collaborators

Lei Wang

Svetlana Tlupova

Nathan Vaughn

Leighton Wilson

Weihua Geng

summary

- fast summation of particle interactions
- barycentric Lagrange interpolation
- BLTC : $O(N \log N)$, BLDTT : $O(N)$
- kernel-independent, GPU-accelerated
- github.com/Treecodes/BaryTree

applications

- electrostatics
- electronic structure
- fluid dynamics
- plasma dynamics