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
motivation: Poisson equation

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

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

- 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)

\ldots many variations
tree-based fast summation (1/2)

\[
\phi_i = \sum_{j=1}^{N} \frac{q_j}{|x_i - y_j|}, \quad i = 1 : N
\]

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

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

- multipole expansion : FMM (Greengard-Rokhlin 1987)

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

kernel-independent methods

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

- polynomial interpolation

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

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

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

barycentric Lagrange form : Berrut-Trefethen (2004 SIREV)

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

“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

$$\phi_i = \sum_{j=1}^{N} G(x_i, y_j)q_j = \sum_{C} \sum_{y_j \in C} G(x_i, y_j)q_j \rightarrow \begin{cases} C_{\text{near}} : \text{PP} \\ C_{\text{far}} : \text{PC} \end{cases}$$

PP: $$\sum_{y_j \in C} G(x_i, y_j)q_j \approx \sum_{k} G(x_i, s_k)\hat{q}_k : \text{PC} \leftarrow \text{note direct sum form}$$

$$\sum_{y_j \in C} \prod_{\ell=1}^{3} L_{k_\ell}(y_{j\ell})q_j$$

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

BLTC is well suited for GPU acceleration

eexample : $N = 10^6$, $G(x, y) = \frac{1}{|x - y|}$

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
new: $\text{BLDTT} = O(N)$
- FMM upward/downward pass adapted to BLI
- interaction lists by dual tree traversal (Appel 1985)
- PP, PC, CP, CC

element: $G(x, y) = \frac{1}{|x - y|}$, 7-8 digit accuracy

application : electrostatics of biomolecules

biomolecule $\Omega_1 : -\varepsilon_1 \nabla^2 \phi(x) = \sum_{k=1}^{N_a} \delta(x - y_k)q_k$ : Poisson

solvent $\Omega_2 : -\varepsilon_2 \nabla^2 \phi(x) + \bar{\kappa}^2 \phi(x) = 0$ : Poisson-Boltzmann

molecular surface $\Gamma : \phi_1 = \phi_2 , \ \varepsilon_1 \partial_n \phi_1 = \varepsilon_2 \partial_n \phi_2$

far-field boundary condition : $\phi(x) \rightarrow 0$ as $|x| \rightarrow \infty$

convert to integral form and solve by BEM
integral form (Juffer et al. 1991)

\[ G_0(x, y) = \frac{1}{4\pi|x - y|}, \quad G_\kappa(x, y) = \frac{e^{-\kappa|x - y|}}{4\pi|x - y|}, \quad \kappa^2 = \frac{\kappa_2}{\epsilon_2}, \quad \epsilon = \frac{\epsilon_2}{\epsilon_1} \]

\[
\frac{1 + \epsilon}{2} \phi(x) = \int_{\Gamma} \left[ K_1(x, y) \partial_n \phi(y) + K_2(x, y) \phi(y) \right] dS_y + S_1(x)
\]

\[
\frac{1 + \epsilon^{-1}}{2} \partial_n \phi(x) = \int_{\Gamma} \left[ K_3(x, y) \partial_n \phi(y) + K_4(x, y) \phi(y) \right] dS_y + S_2(x)
\]

\[ K_1 = G_0 - G_\kappa, \quad K_4 = \partial_{n_x n_y}^2 (G_\kappa - G_0) \]

\[ K_2 = \epsilon \partial_{n_y} G_\kappa - \partial_{n_y} G_0, \quad K_3 = \partial_{n_x} G_0 - \epsilon^{-1} \partial_{n_x} G_\kappa \]

\[ S_1(x) = \frac{1}{\epsilon_1} \sum_{k=1}^{N_a} G_0(x, y_k) q_k, \quad S_2(x) = \frac{1}{\epsilon_1} \sum_{k=1}^{N_a} \partial_{n_x} G_0(x, y_k) q_k \]
BEM

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

eexample : RNA binding domain of E. coli $\rho$-factor, $N_a = 2069$

scale = 3, $N \approx 10^5$, run time = 110 s, error $\approx 1\%$
BEM
- triangulate $\Gamma$ by NanoShaper (Decherchi, Rocchia)
- discretize integrals by node-patch (Lu)
- solve for $\phi, \partial_n \phi$ on $\Gamma$ by GMRES
- compute matrix-vector product by BLDTT on 1 GPU

example: Zika virus, $N_\alpha = 13248$

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