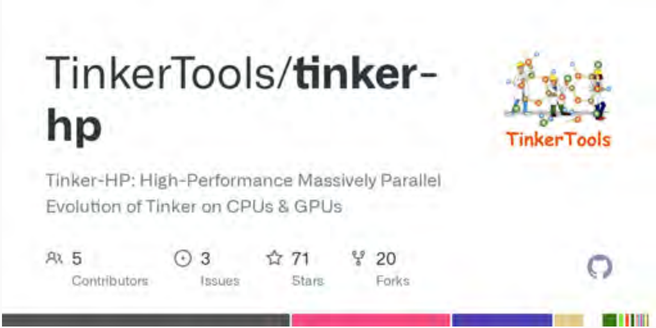
ANKH: A scalable divide and conquer strategy for energy computation on modern HPC architectures

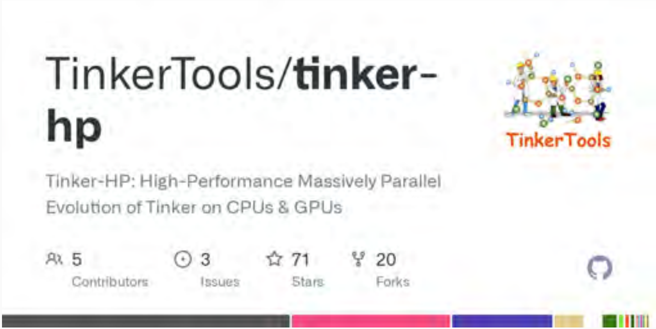
Olivier Adjoua, Igor Chollet, Louis Lagardère, Jean-Philip Piquemal

Numpex 07/02/24



Tinker-hp

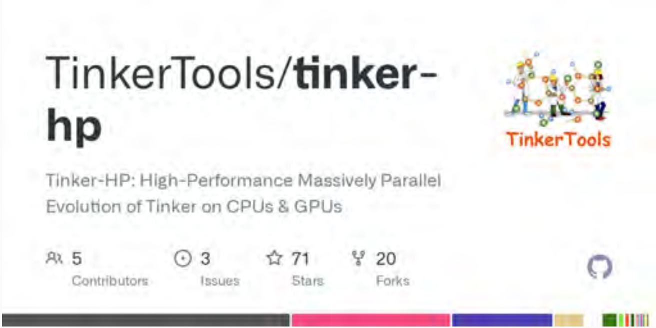
Tinker-hp is a pre-exascale, multi-CPUs, multi-GPUs, multi-precision package dedicated to long molecular dynamics simulations with classical and polarizable force fields.



Tinker-hp

Tinker-hp is a pre-exascale, multi-CPUs, multi-GPUs, multi-precision package dedicated to long molecular dynamics simulations with classical and polarizable force fields.

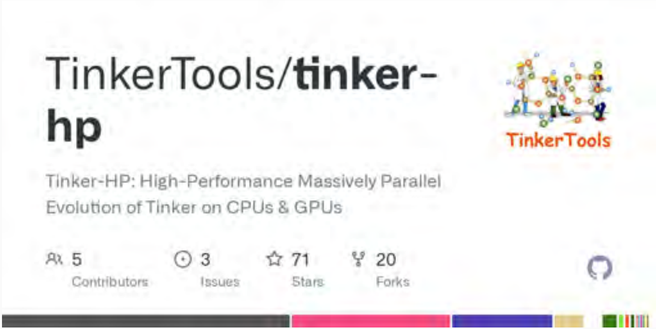
Rely on Ewald Summation and large-scale FFTs for fast computations.



Tinker-hp

Tinker-hp is a pre-exascale, multi-CPUs, multi-GPUs, multi-precision package dedicated to long molecular dynamics simulations with classical and polarizable force fields.

Rely on Ewald Summation and large-scale FFTs for fast computations.

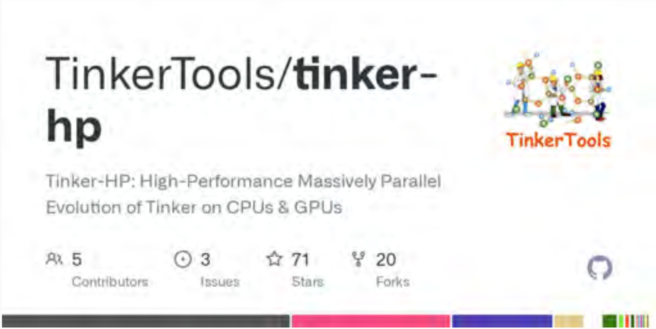


Tinker-hp

Tinker-hp is a pre-exascale, multi-CPUs, multi-GPUs, multi-precision package dedicated to long molecular dynamics simulations with classical and polarizable force fields.

Rely on Ewald Summation and large-scale FFTs for fast computations.

Fast Multipole Methods as alternative to Ewald Summation?



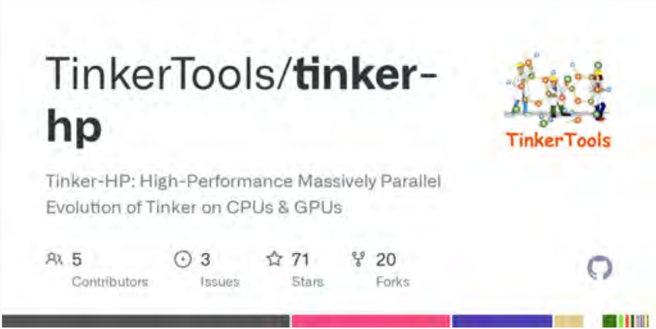
Tinker-hp

Tinker-hp is a pre-exascale, multi-CPUs, multi-GPUs, multi-precision package dedicated to long molecular dynamics simulations with classical and polarizable force fields.

Rely on Ewald Summation and large-scale FFTs for fast computations.

Fast Multipole Methods (FMM) as alternative to Ewald Summation?

$\rightarrow$ Solve scalability in non-polarizable case, may induce errors on forces.

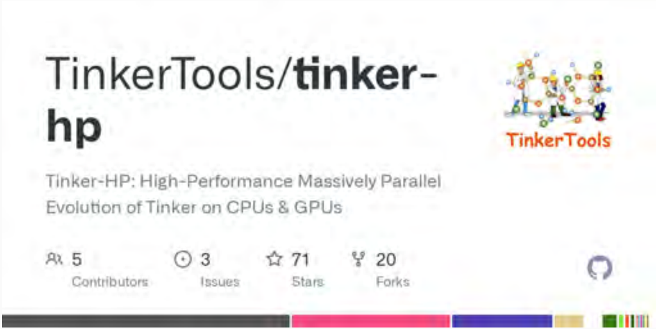


Tinker-hp

Tinker-hp is a pre-exascale, multi-CPUs, multi-GPUs, multi-precision package dedicated to long molecular dynamics simulations with classical and polarizable force fields.

Rely on Ewald Summation and large-scale FFTs for fast computations.

FMM-based Ewald Summation?



Tinker-hp

Tinker-hp is a ~~pre~~-exascale, multi-CPUs, multi-GPUs, multi-precision package dedicated to long molecular dynamics simulations with classical and polarizable force fields.

FMM-based Ewald Summation?

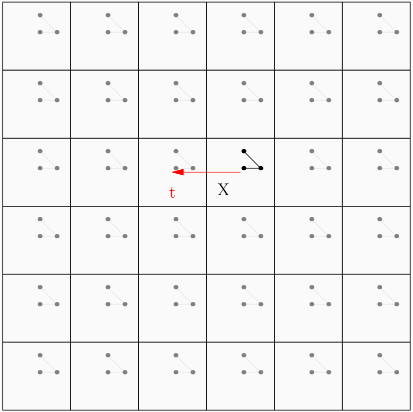
→ ANKH: scalable, fast, adapted to classical and polarizable force field, controlled error bounds, including GPU and CPU+GPU formulations

Task and approach

Energy calculation

$$\mathcal{E} := \sum_{\mathbf{t} \in 2r_B\mathbb{Z}^3} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}} \mathfrak{D}_{\mathbf{y}} |\mathbf{x} - \mathbf{y} + \mathbf{t}|^{-1}$$

with r_B the box radius and $\mathfrak{D}_{\mathbf{x}} := q_{\mathbf{x}} + \mu_{\mathbf{x}} \cdot \nabla_{\mathbf{x}} + \Theta_{\mathbf{x}} : \nabla^2$.



Task and approach

Energy calculation

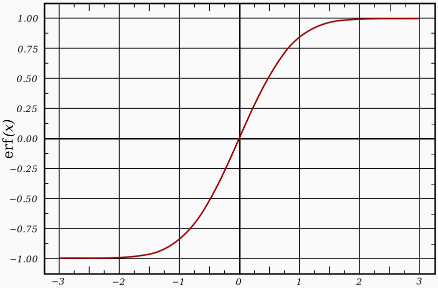
$$\mathcal{E} := \sum_{\mathbf{t}\in 2r_B\mathbb{Z}^3} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}}\mathfrak{D}_{\mathbf{y}} |\mathbf{x}-\mathbf{y}+\mathbf{t}|^{-1}$$

with r_B the box radius and $\mathfrak{D}_{\mathbf{x}} := q_{\mathbf{x}} + \mu_{\mathbf{x}} \cdot \nabla_{\mathbf{x}} + \Theta_{\mathbf{x}} : \nabla^2$.

Problem: the sum over **t**'s only conditionally converges and the convergence is very slow.

Idea: $\frac{1}{R} = \frac{\lambda(R)+1-\lambda(R)}{R}$

Good choice: $\lambda(R) = erf(\xi R)$



Task and approach

Energy calculation

$$\mathcal{E} := \sum_{\mathbf{t} \in 2r_B\mathbb{Z}^3} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}} \mathfrak{D}_{\mathbf{y}} |\mathbf{x} - \mathbf{y} + \mathbf{t}|^{-1}$$

with r_B the box radius and $\mathfrak{D}_{\mathbf{x}} := q_{\mathbf{x}} + \mu_{\mathbf{x}} \cdot \nabla_{\mathbf{x}} + \Theta_{\mathbf{x}} : \nabla^2$.

Ewald summation

$$\mathcal{E} = \mathcal{E}_{real} + \mathcal{E}_{rec} + \mathcal{E}_{self}$$

$$\mathcal{E}_{real} := \sum_{\mathbf{t}} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}} \mathfrak{D}_{\mathbf{y}} \left(\frac{erfc\left(\xi|\mathbf{x} - \mathbf{y} + \mathbf{t}|\right)}{|\mathbf{x} - \mathbf{y} + \mathbf{t}|} \right),$$

$$\mathcal{E}_{rec} := \frac{1}{2\pi(2r_B)^2} \sum_{\mathbf{m} \neq \mathbf{0}} \frac{e^{-2\left(\frac{\pi r_B}{\xi}\right)^2 \mathbf{m}\cdot\mathbf{m}}}{\mathbf{m} \cdot \mathbf{m}} S(\mathbf{m}) S(-\mathbf{m}),$$

$$S(\mathbf{m}) := \sum_{\mathbf{x}} e^{i\frac{\pi}{r_B}\mathbf{m}\cdot\mathbf{x}} (q_{\mathbf{x}} + 2i\pi r_B \mu_{\mathbf{x}} \cdot \mathbf{m} - (2\pi r_B)^2 \Theta_{\mathbf{x}} : (\mathbf{m}\mathbf{m}^T)),$$

$$\mathcal{E}_{self} := -\frac{\xi}{\sqrt{\pi}} \sum_{\mathbf{x}} \left(q_{\mathbf{x}}^2 + \frac{2\xi^2}{3} \mu_{\mathbf{x}} \cdot \mu_{\mathbf{x}} + \frac{8\xi^4}{5} \Theta_{\mathbf{x}} : \Theta_{\mathbf{x}} \right).$$

Task and approach

Particle Mesh Ewald (PME)

Find optimal ξ, apply cutoff for $\mathcal{E}_{real}$, compute $S(\mathbf{m})$'s using interpolation and Fast Fourier Transforms.

Ewald summation

$$\mathcal{E} = \mathcal{E}_{real} + \mathcal{E}_{rec} + \mathcal{E}_{self}$$

$$\mathcal{E}_{real} := \sum_{\mathbf{t}} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}} \mathfrak{D}_{\mathbf{y}} \left(\frac{\mathit{erfc}\,(\xi|\mathbf{x}-\mathbf{y}+\mathbf{t}|)}{|\mathbf{x}-\mathbf{y}+\mathbf{t}|} \right),$$

$$\mathcal{E}_{rec} := \frac{1}{2\pi(2r_B)^2} \sum_{\mathbf{m}\neq\mathbf{0}} \frac{e^{-2\left(\frac{\pi r_B}{\xi}\right)^2 \mathbf{m}\cdot\mathbf{m}}}{\mathbf{m}\cdot\mathbf{m}} S(\mathbf{m}) S(-\mathbf{m}),$$

$$S(\mathbf{m}) := \sum_{\mathbf{x}} e^{i\frac{\pi}{r_B}\mathbf{m}\cdot\mathbf{x}} (q_{\mathbf{x}} + 2i\pi r_B \mu_{\mathbf{x}} \cdot \mathbf{m} - (2\pi r_B)^2 \Theta_{\mathbf{x}} : (\mathbf{m}\mathbf{m}^T)),$$

$$\mathcal{E}_{self} := -\frac{\xi}{\sqrt{\pi}} \sum_{\mathbf{x}} \left(q_{\mathbf{x}}^2 + \frac{2\xi^2}{3} \mu_{\mathbf{x}} \cdot \mu_{\mathbf{x}} + \frac{8\xi^4}{5} \Theta_{\mathbf{x}} : \Theta_{\mathbf{x}} \right).$$

Task and approach

Particle Mesh Ewald (PME)

Find optimal ξ, apply cutoff for $\mathcal{E}_{real}$, compute $S(\mathbf{m})$'s using interpolation and Fast Fourier Transforms $\rightarrow$ possibly "poor" scaling on distributed memory

Ewald summation

$$\mathcal{E} = \mathcal{E}_{real} + \mathcal{E}_{rec} + \mathcal{E}_{self}$$

$$\mathcal{E}_{real} := \sum_{\mathbf{t}} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}} \mathfrak{D}_{\mathbf{y}} \left(\frac{erfc\,(\xi|\mathbf{x}-\mathbf{y}+\mathbf{t}|)}{|\mathbf{x}-\mathbf{y}+\mathbf{t}|} \right),$$

$$\mathcal{E}_{rec} := \frac{1}{2\pi(2r_B)^2} \sum_{\mathbf{m}\neq\mathbf{0}} \frac{e^{-2\left(\frac{\pi r_B}{\xi}\right)^2 \mathbf{m}\cdot\mathbf{m}}}{\mathbf{m}\cdot\mathbf{m}} S(\mathbf{m}) S(-\mathbf{m}),$$

$$S(\mathbf{m}) := \sum_{\mathbf{x}} e^{i\frac{\pi}{r_B}\mathbf{m}\cdot\mathbf{x}} (q_{\mathbf{x}} + 2i\pi r_B \mu_{\mathbf{x}} \cdot \mathbf{m} - (2\pi r_B)^2 \Theta_{\mathbf{x}} : (\mathbf{m}\mathbf{m}^T)),$$

$$\mathcal{E}_{self} := -\frac{\xi}{\sqrt{\pi}} \sum_{\mathbf{x}} \left(q_{\mathbf{x}}^2 + \frac{2\xi^2}{3} \mu_{\mathbf{x}} \cdot \mu_{\mathbf{x}} + \frac{8\xi^4}{5} \Theta_{\mathbf{x}} : \Theta_{\mathbf{x}} \right).$$

Task and approach

New simple idea

Choose very small ξ (in practice, $\xi \approx 0.01$) and exploit fast hierarchical techniques for N-body problems to compute $\mathcal{E}_{real}$.

Ewald summation

$$\mathcal{E} = \mathcal{E}_{real} + \mathcal{E}_{rec} + \mathcal{E}_{self}$$

$$\mathcal{E}_{real} := \sum_{\mathbf{t}} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}} \mathfrak{D}_{\mathbf{y}} \left(\frac{erfc\,(\xi|\mathbf{x}-\mathbf{y}+\mathbf{t}|)}{|\mathbf{x}-\mathbf{y}+\mathbf{t}|} \right),$$

$$\mathcal{E}_{rec} := \frac{1}{2\pi(2r_B)^2} \sum_{\mathbf{m}\neq\mathbf{0}} \frac{e^{-2\left(\frac{\pi r_B}{\xi}\right)^2 \mathbf{m}\cdot\mathbf{m}}}{\mathbf{m}\cdot\mathbf{m}} S(\mathbf{m}) S(-\mathbf{m}),$$

$$S(\mathbf{m}) := \sum_{\mathbf{x}} e^{i\frac{\pi}{r_B}\mathbf{m}\cdot\mathbf{x}} (q_{\mathbf{x}} + 2i\pi r_B \mu_{\mathbf{x}} \cdot \mathbf{m} - (2\pi r_B)^2 \Theta_{\mathbf{x}} : (\mathbf{m}\mathbf{m}^T)),$$

$$\mathcal{E}_{self} := -\frac{\xi}{\sqrt{\pi}} \sum_{\mathbf{x}} \left(q_{\mathbf{x}}^2 + \frac{2\xi^2}{3} \mu_{\mathbf{x}} \cdot \mu_{\mathbf{x}} + \frac{8\xi^4}{5} \Theta_{\mathbf{x}} : \Theta_{\mathbf{x}} \right).$$

Task and approach

New simple idea

Choose very small ξ (in practice, $\xi \approx 0.01$) and exploit fast hierarchical techniques for N-body problems to compute $\mathcal{E}_{real}$.

Ewald summation

$$\mathcal{E} = \mathcal{E}_{real} + \mathcal{E}_{rec} + \mathcal{E}_{self}$$

$$\mathcal{E}_{real} := \sum_{\mathbf{t} < T_{conv}} \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathfrak{D}_{\mathbf{x}} \mathfrak{D}_{\mathbf{y}} \left(\frac{erfc\,(\xi|\mathbf{x}-\mathbf{y}+\mathbf{t}|)}{|\mathbf{x}-\mathbf{y}+\mathbf{t}|} \right),$$

$$\mathcal{E}_{rec} \approx 0,$$

$$\mathcal{E}_{self} = \text{easy to compute quantity.}$$

N-body problems

N-body problems

Notations: $\mathbb{C}[Z]$ the set of application from Z to $\mathbb{C}$. The cardinal of Z is denoted $\#Z$. $\mathbb{C}[Z]$ isomorphic to $\mathbb{C}^{\#Z} = \mathbb{C} \times ... \times \mathbb{C}$.

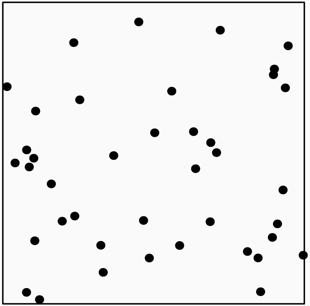
Goal: given $\mathbb{X}$ and $\mathbb{Y}$ two point clouds, quickly evaluate

$$\mathbb{G} \ : \ \mathbb{C}[\mathbb{Y}] \to \mathbb{C}[\mathbb{X}] \text{ such that } \forall\, q \in \mathbb{C}[\mathbb{Y}]$$

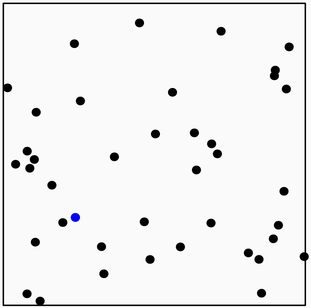
$$\left(\mathbb{G} \cdot q\right)(\mathbf{x}) := \sum_{\mathbf{y} \in \mathbb{Y}} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}), \qquad \forall\, \mathbf{x} \in \mathbb{X},$$

$G \ : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$ **asymptotically smooth** and **singular at $\mathbf{x} = \mathbf{y}$**, such as **Coulomb kernel** $G(\mathbf{x}, \mathbf{y}) := \frac{1}{|\mathbf{x} - \mathbf{y}|}$.

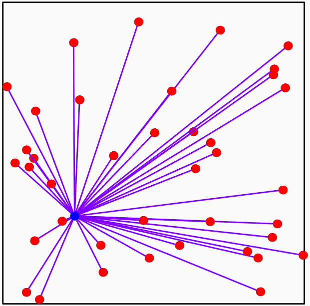
Overall idea



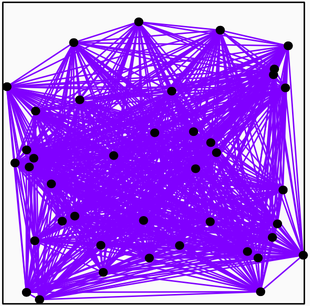
Overall idea



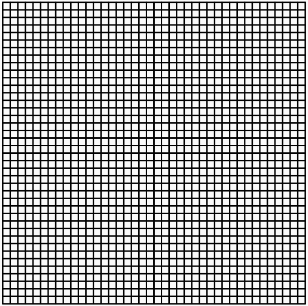
Overall idea

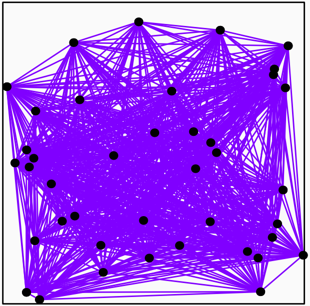


Overall idea

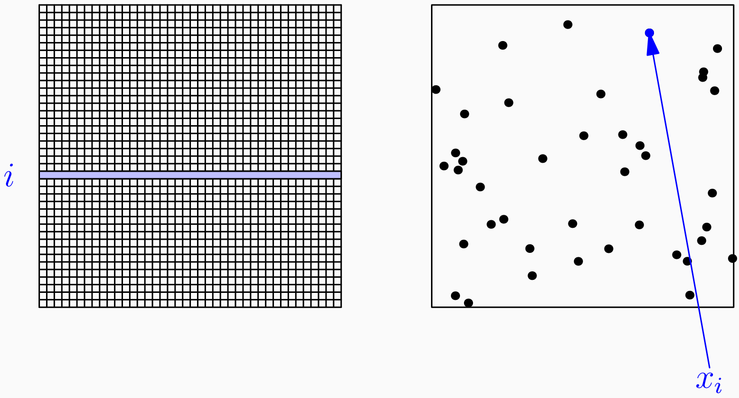


Overall idea

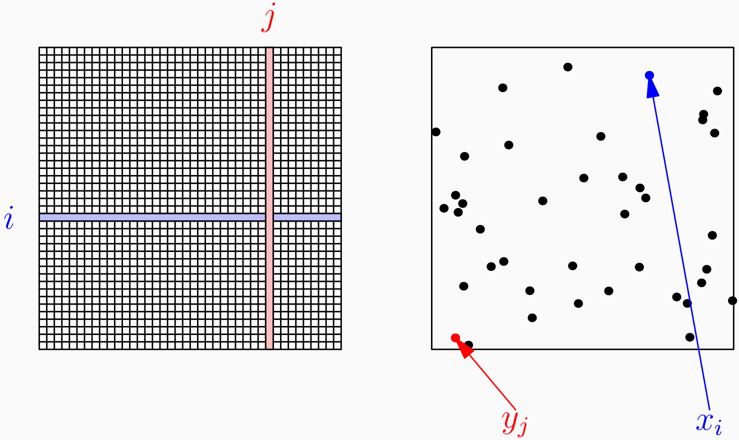




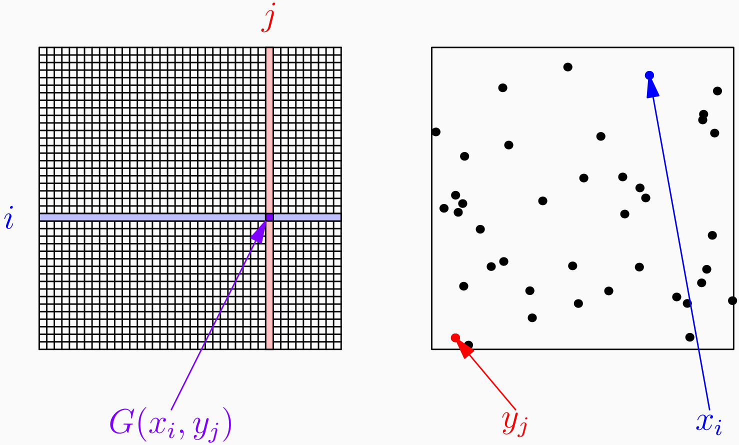
Overall idea



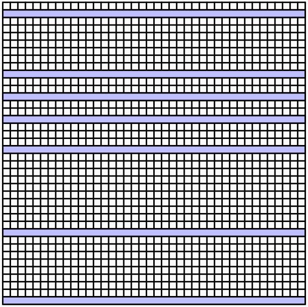
Overall idea

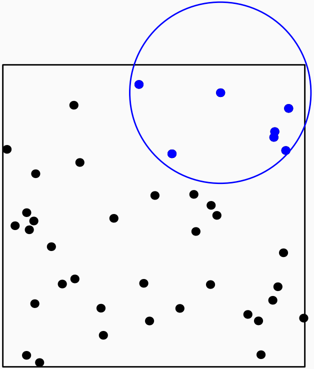


Overall idea

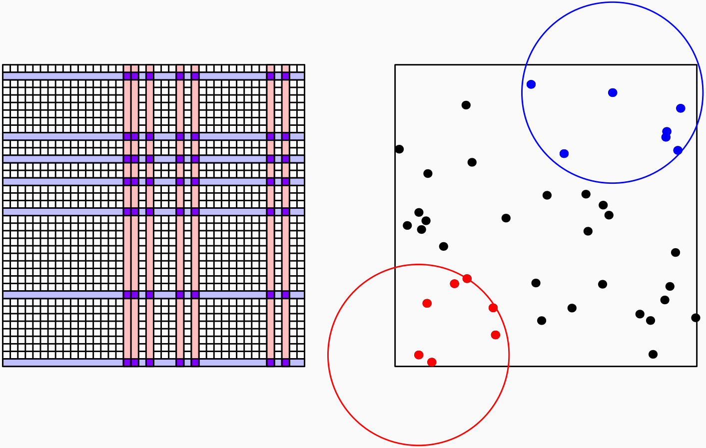


Overall idea

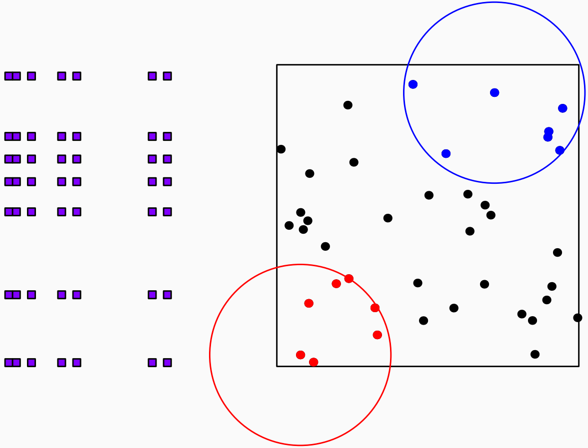




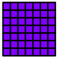
Overall idea

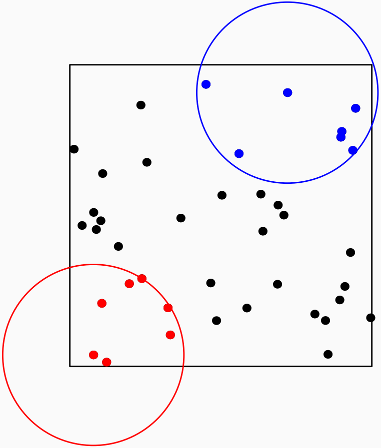


Overall idea

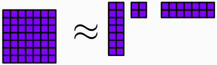


Overall idea

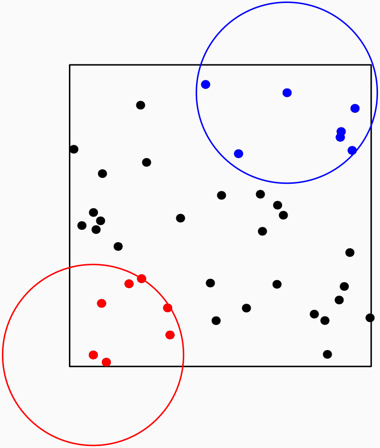




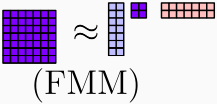
Overall idea

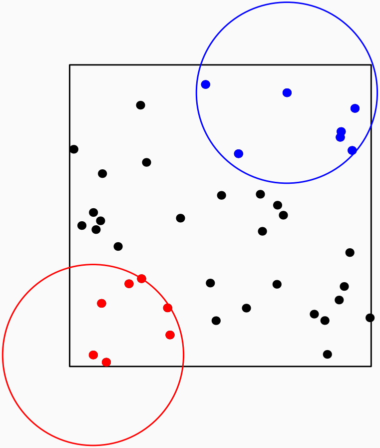


(SVD)

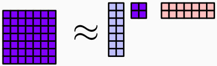


Overall idea

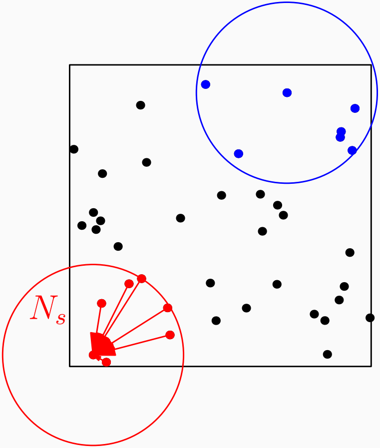




Overall idea

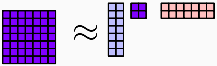


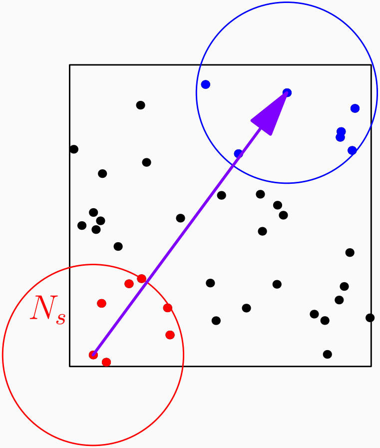
(FMM)



$O(N_s)$

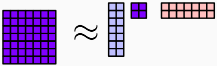
Overall idea

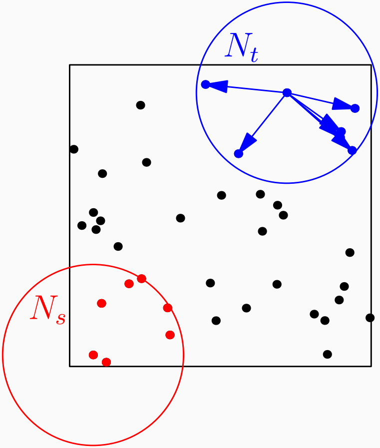




$O(N_s)$ $+O(1)$

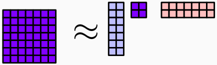
Overall idea

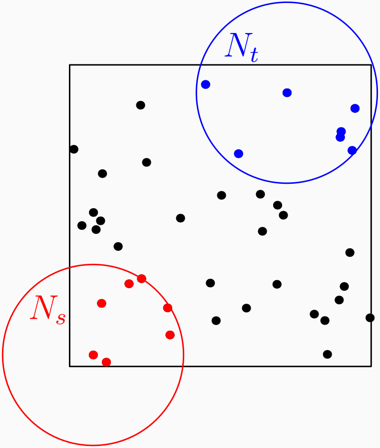




$$O(N_s) + O(1) + O(N_t)$$

Overall idea





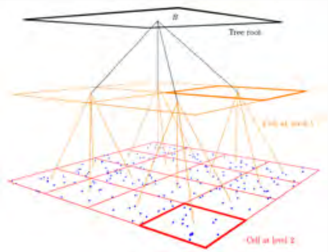
$$\Rightarrow O(N_t + N_s) < O(N_t N_s)$$

Fast Multipole Methods

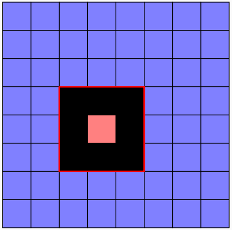
Ingredients of a FMM

Strategy:

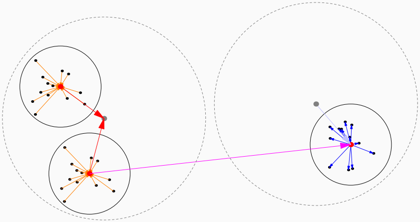
- Introduce a hierarchical space representation (2^d-tree)
- Define a Multipole Acceptance Criterion
- Find way of aggregate and scatter expansions



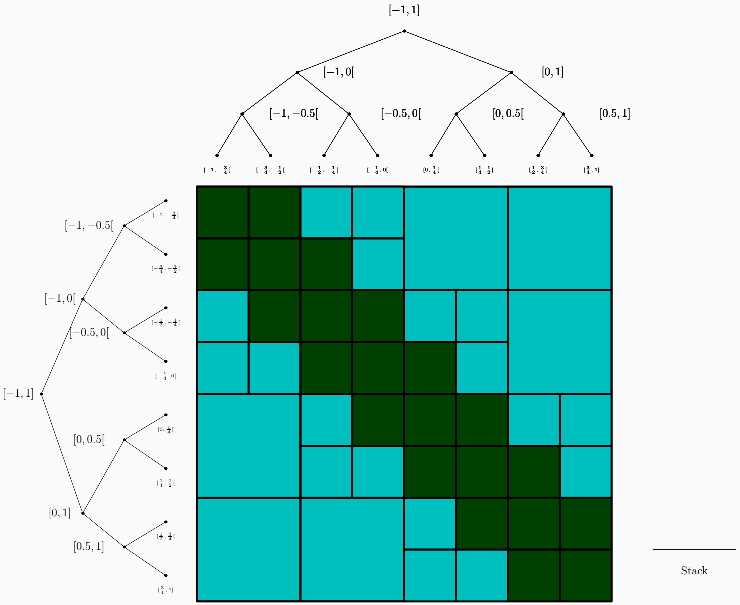
Multipole Acceptance Criterion



Gathering / scattering



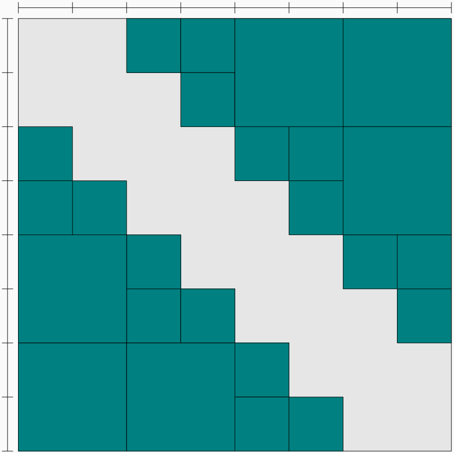
Near and far field separation



Sparsity of the far field matrix

Suppose that t and s are two admissible cells. We have:

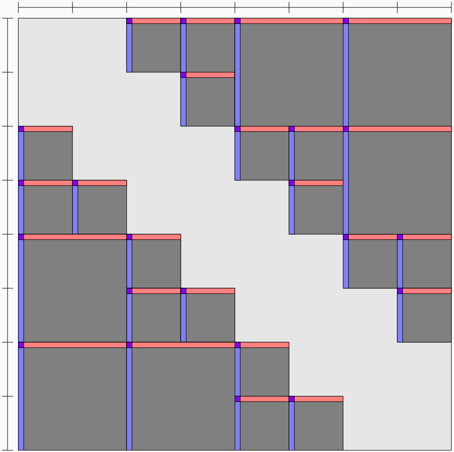
$$\sum_{\mathbf{y}\in Y_{|s}} G(\mathbf{x},\mathbf{y})q(\mathbf{y}) \approx \sum_{k=1}^{r} S_k(\mathbf{x}) \sum_{l=1}^{r} G(\mathbf{x}_k,\mathbf{y}_l) \sum_{\mathbf{y}\in Y_{|s}} S_l(\mathbf{y})q(\mathbf{y})$$



Sparsity of the far field matrix

Suppose that t and s are two admissible cells. We have:

$$\sum_{\mathbf{y}\in Y_{|s}} G(\mathbf{x},\mathbf{y})q(\mathbf{y}) \approx \sum_{k=1}^{r} S_k(\mathbf{x}) \sum_{l=1}^{r} G(\mathbf{x}_k,\mathbf{y}_l) \sum_{\mathbf{y}\in Y_{|s}} S_l(\mathbf{y})q(\mathbf{y})$$



ANKH

Numerical differentiation

Le H be the following *asymptotically smooth* kernel[1]

$$H(\mathbf{x}, \mathbf{y}) := \frac{\mathit{erfc}\,(\xi|\mathbf{x}-\mathbf{y}|)}{|\mathbf{x}-\mathbf{y}|}.$$

Using (Chebyshev-)Lagrange polynomial interpolation[2]:

$$\sum_{\mathbf{x}\in t}\sum_{\mathbf{y}\in s}\mathfrak{D}_{\mathbf{x}}\mathfrak{D}_{\mathbf{y}}H(\mathbf{x},\mathbf{y}) \approx \sum_{k}\underbrace{\left(\sum_{\mathbf{x}\in t}\mathfrak{D}_{\mathbf{x}}S_k[t](\mathbf{x})\right)}_{=:\mathcal{Q}_t(\mathbf{x}_k^t)}\sum_{l}H(\mathbf{x}_k^t,\mathbf{y}_l^s)\underbrace{\left(\sum_{\mathbf{y}\in s}\mathfrak{D}_{\mathbf{y}}S_l[s](\mathbf{y})\right)}_{=:\mathcal{Q}_s(\mathbf{y}_l^s)}$$

$$= \sum_{k}\mathcal{Q}_t(\mathbf{x}_k^t)\sum_{l}H(\mathbf{x}_k^t,\mathbf{y}_l^s)\mathcal{Q}_s(\mathbf{y}_l^s).$$

[1] Badreddine, C., Grigori, Journal of Computational Physics (2023)
[2] C., Lagardère, Piquemal, Journal of Chemical Theory and Computation (2023)

Numerical differentiation

Le H be the following *asymptotically smooth* kernel[1]

$$H(\mathbf{x},\mathbf{y}) := \frac{erfc\left(\xi|\mathbf{x}-\mathbf{y}|\right)}{|\mathbf{x}-\mathbf{y}|}.$$

Using (Chebyshev-)Lagrange polynomial interpolation[2]:

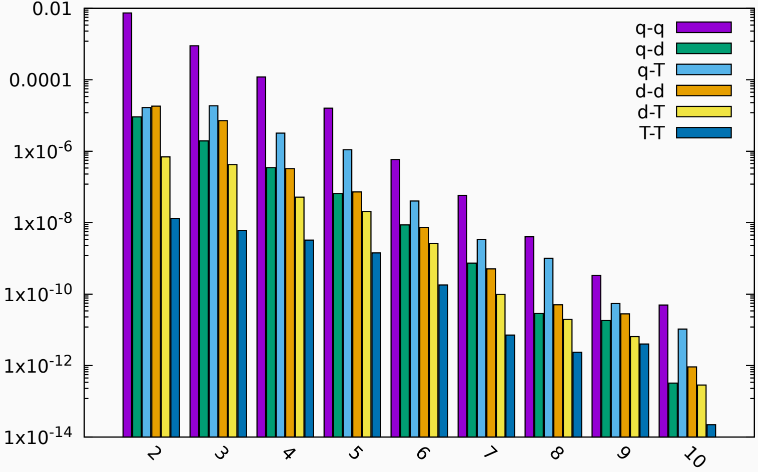
$$\sum_{\mathbf{x}\in t}\sum_{\mathbf{y}\in s}\mathfrak{D}_{\mathbf{x}}\mathfrak{D}_{\mathbf{y}}H(\mathbf{x},\mathbf{y}) \approx \sum_k \underbrace{\left(\sum_{\mathbf{x}\in t}\mathfrak{D}_{\mathbf{x}}S_k[t](\mathbf{x})\right)}_{=:\mathcal{Q}_t(\mathbf{x}_k^t)} \sum_l H(\mathbf{x}_k^t,\mathbf{y}_l^s) \underbrace{\left(\sum_{\mathbf{y}\in s}\mathfrak{D}_{\mathbf{y}}S_l[s](\mathbf{y})\right)}_{=:\mathcal{Q}_s(\mathbf{y}_l^s)}$$

$$= \sum_k \mathcal{Q}_t(\mathbf{x}_k^t)\sum_l H(\mathbf{x}_k^t,\mathbf{y}_l^s)\mathcal{Q}_s(\mathbf{y}_l^s).$$

Problem: $S_k'(x) = \frac{2}{L\sqrt{1-x^2}}\sum_{m=1}^{L-1} mT_m(x_k)sin(m\ acos(x))$, numerically unstable!

[1] Badreddine, C., Grigori, Journal of Computational Physics (2023)
[2] C., Lagardère, Piquemal, Journal of Chemical Theory and Computation (2023)

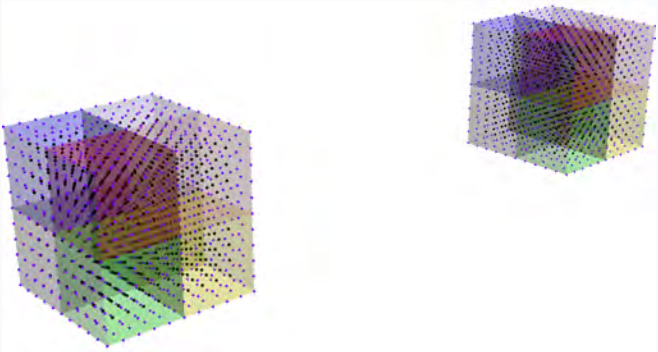
Numerical error



Equispaced interpolation

Exploiting interpolation over equispaced grids...

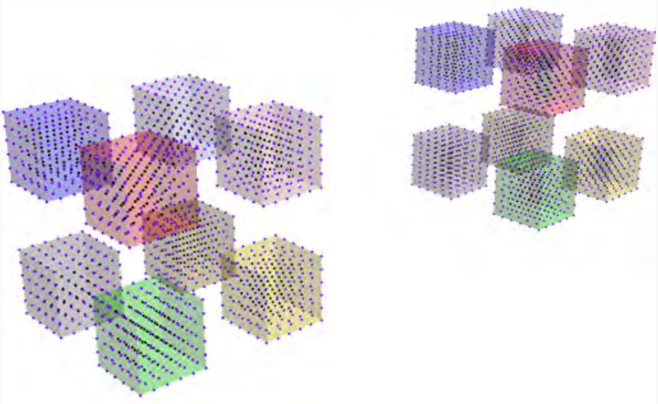
$$G(\mathbf{x}, \mathbf{y})$$



Equispaced interpolation

...we obtain a regular “cartesian” 3D grid of 3D nodes.

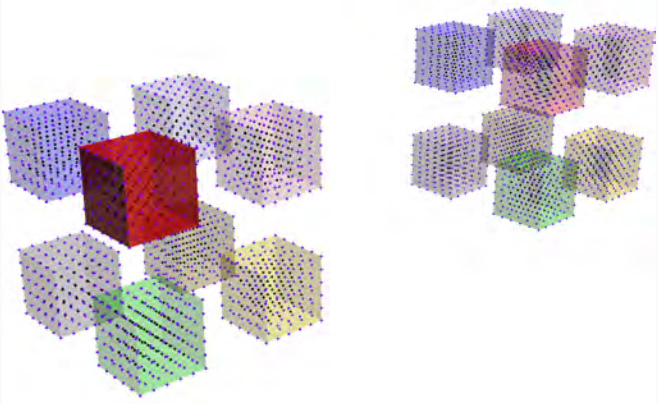
$$G(\mathbf{t}(\mathbf{x}) + \hat{\mathbf{x}}, \mathbf{t}(\mathbf{y}) + \hat{\mathbf{y}})$$



Equispaced interpolation

...we obtain a regular “cartesian” 3D grid of 3D nodes.

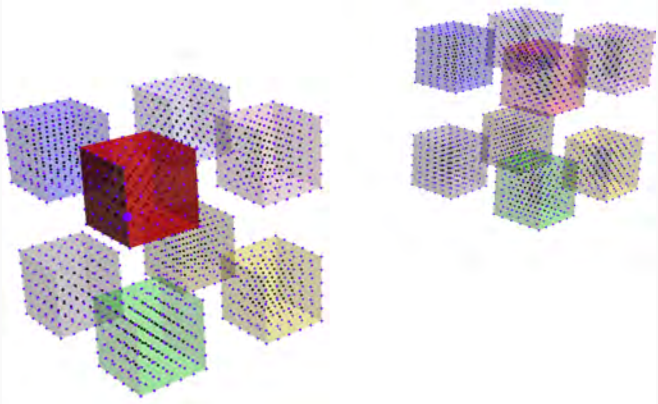
$$G(\underbrace{\mathbf{t_x}}_{cell} + \hat{\mathbf{x}}, \mathbf{t_y} + \hat{\mathbf{y}})$$



Equispaced interpolation

...we obtain a regular "cartesian" 3D grid of 3D nodes.

$$G(\mathbf{t_x} + \underbrace{\hat{\mathbf{x}}}_{node}, \mathbf{t_y} + \hat{\mathbf{y}})$$



Equispaced interpolation

Theorem[a]

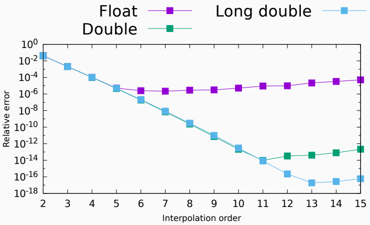
[a] C., Claeys, Fortin, Grigori, SIAM Journal on Scientific Computing (2023)

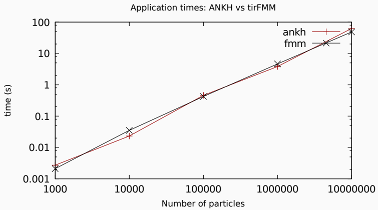
On well-separated cells t, s, $\lim_{L \to +\infty} \left|\left|\mathcal{I}_L^{t \times s}[G] - G\right|\right|_{L^\infty(t \times s)} = 0.$

Theorem[a]

[a] C., Lagardère, Piquemal, JCTC (2023)

$$\mathcal{E}_{real} = \mathcal{N} + \mathbf{v}^T \chi^* \mathbb{F}_6^* \mathbf{D} \mathbb{F}_6 \chi \mathbf{v}.$$

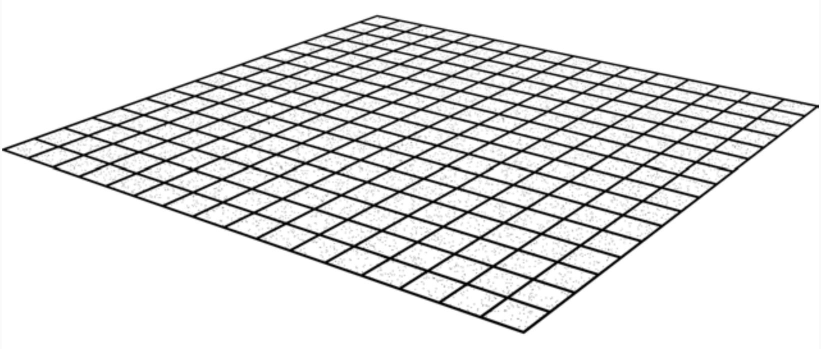




Divide-and-conquer strategy

Algorithmic

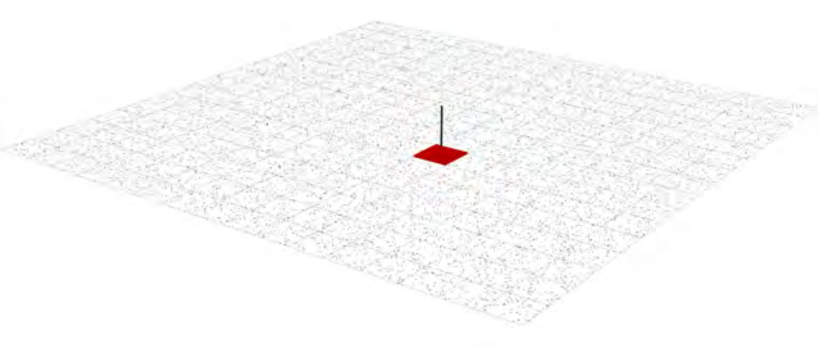
1. Build space decomposition as **perfect** 2^d-tree leaves ($\mathcal{O}(N)$)
2.
3.
4.



Divide-and-conquer strategy

Algorithmic

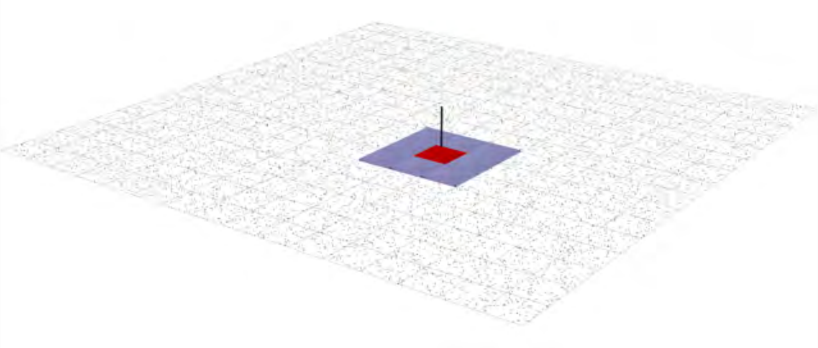
1. Build space decomposition as **perfect** 2^d-tree leaves $\rightarrow \mathcal{O}(N)$
2. Compute local problem compression $\rightarrow \mathcal{O}\left(\frac{N}{log(N)}\right)$
3.
4.



Divide-and-conquer strategy

Algorithmic

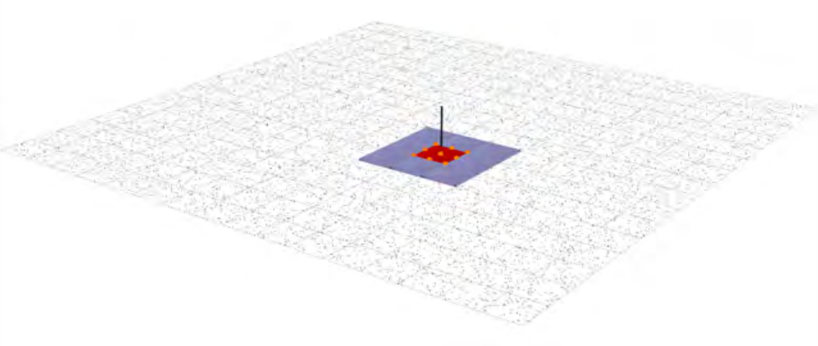
1. Build space decomposition as **perfect** 2^d-tree leaves $\rightarrow \mathcal{O}(N)$
2. Compute local problem compression $\rightarrow \mathcal{O}\left(\frac{N}{log(N)}\right)$
3. Exchange compression with neighbors $\rightarrow \mathcal{O}\left(\alpha + \beta \frac{N}{log(N)}\right)$
4.



Divide-and-conquer strategy

Algorithmic

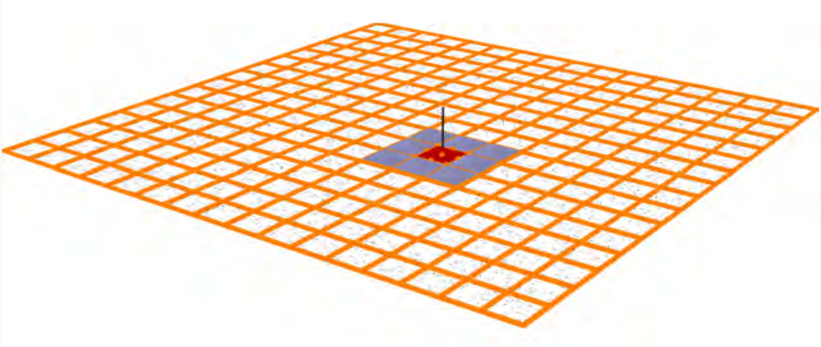
1. Build space decomposition as **perfect** 2^d-tree leaves $\rightarrow \mathcal{O}(N)$
2. Compute local problem compression $\rightarrow \mathcal{O}\left(\frac{N}{log(N)}\right)$
3. Exchange compression with neighbors $\rightarrow \mathcal{O}\left(\alpha + \beta\frac{N}{log(N)}\right)$
4. Compute leaf far field information $\rightarrow \mathcal{O}\left(\frac{N}{log(N)}\right)$



Divide-and-conquer strategy

Algorithmic

1. Build space decomposition as **perfect** 2^d-tree leaves $\rightarrow \mathcal{O}(N)$
2. Compute local problem compression $\rightarrow \mathcal{O}\left(\frac{N}{log(N)}\right)$
3. Exchange compression with neighbors $\rightarrow \mathcal{O}\left(\alpha + \beta \frac{N}{log(N)}\right)$
4. Compute leaf far field information $\rightarrow \mathcal{O}\left(\frac{N}{log(N)}\right)$

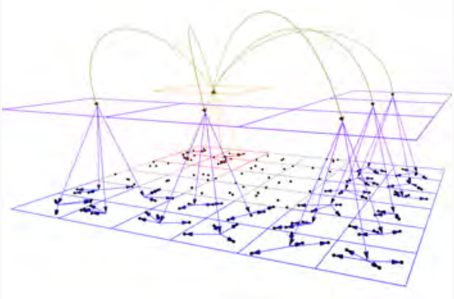


Divide-and-conquer strategy

Algorithmic in parallel

1. Build space decomposition as **perfect** 2^d-tree leaves $\rightarrow \mathcal{O}\left(\frac{N}{P}\right)$
2. Compute local problem compression $\rightarrow \mathcal{O}\left(\frac{N}{P}\right)$
3. Exchange compression with neighbors $\rightarrow \mathcal{O}\left(\alpha + \beta\frac{N}{P}\right)$
4. Compute leaf far field information $\rightarrow \mathcal{O}\left(\frac{N}{P}\right)$

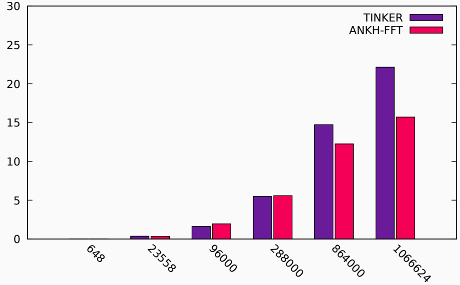
Overall $\mathcal{O}\left(\alpha + \frac{N}{P}(\beta + \gamma)\right)$ (*joint work with P. Fortin for MPI implementation*)

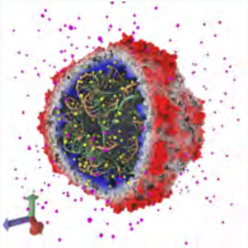


Results

(First MPI version ”soon” publicly available)

- Single CPU execution
- Tinker-hp is a reference implementation of PME
- ANKH-FFT on github: lChollet/ankh-fft



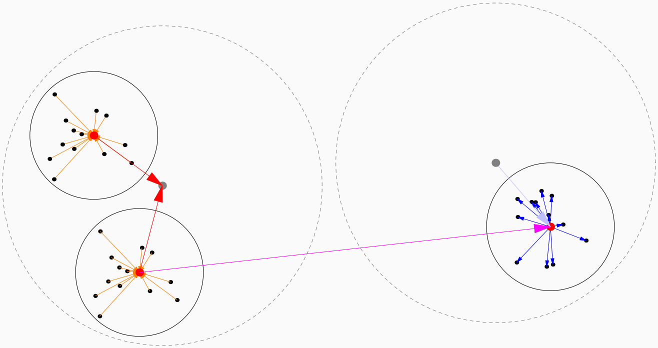


High-level formulation

Freely generated vector spaces and reformulation

If (t,s) is a well-separated pair and $(s', t') \in Sons(s) \times Sons(t)$:

$$P2P[t', s'] \approx L2P[t'] \cdot L2L[t', t] \cdot M2L[t, s] \cdot M2M[s, s'] \cdot P2M[s'] \cdot q_{|s'}.$$



Freely generated vector spaces and reformulation

If (t,s) is a well-separated pair and $(s', t') \in Sons(s) \times Sons(t)$:

$$P2P[t', s'] \approx L2P[t'] \cdot L2L[t', t] \cdot M2L[t, s] \cdot M2M[s, s'] \cdot P2M[s'] \cdot q_{|s'}.$$

Freely generated vector space and abstract FMM operators

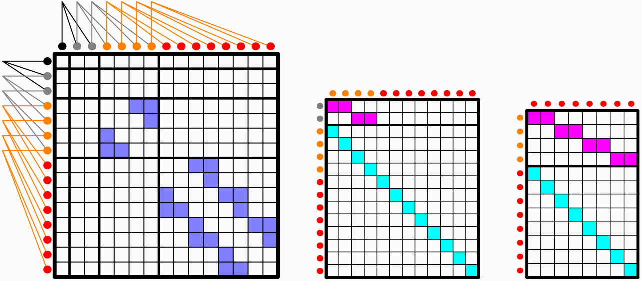
$\mathcal{T}$ the 2^d-tree, $r_{\mathcal{T}} : \mathcal{T} \to \mathbb{N}^*$ is a rank distribution over $\mathcal{T}$.
F.g.v.s. $\mathbb{C}[r_{\mathcal{T}}] := \prod_{c \in \mathcal{T}} \mathbb{C}^{r_{\mathcal{T}}(c)}$. We have

- P2M : $\mathbb{C}\left[Y \cap s'\right] \to \mathbb{C}\left[r_{\mathcal{T}_{|s'}}\right]$,
- M2M : $\mathbb{C}\left[r_{\mathcal{T}_{|s'}}\right] \to \mathbb{C}\left[r_{\mathcal{T}_{|s}}\right]$,
- M2L : $\mathbb{C}\left[r_{\mathcal{T}_{|s}}\right] \to \mathbb{C}\left[r_{\mathcal{T}_{|t}}\right]$,
- L2L : $\mathbb{C}\left[r_{\mathcal{T}_{|t}}\right] \to \mathbb{C}\left[r_{\mathcal{T}_{|t'}}\right]$,
- L2P : $\mathbb{C}\left[r_{\mathcal{T}_{|t'}}\right] \to \mathbb{C}\left[X \cap t'\right]$,
- P2P : $\mathbb{C}\left[X \cap t'\right] \to \mathbb{C}\left[Y \cap s'\right]$.

Freely generated vector spaces and reformulation

If (t,s) is a well-separated pair and $(s', t') \in \mathit{Sons}(s) \times \mathit{Sons}(t)$:

$$P2P[t', s'] \approx L2P[t'] \cdot L2L[t', t] \cdot M2L[t, s] \cdot M2M[s, s'] \cdot P2M[s'] \cdot q_{|s'}.$$



Shape of FMM operators

M2L symmetries

M2L $A_{t,s}$ between well-separated t and s. $A_{t,s}$ is $\mathcal{G}$-invariant:

$$\forall g, h \in \mathcal{G}, A_{g \cdot t, h \cdot s} = P_g \cdot A_{t,s} \cdot P_h,$$

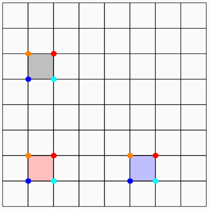
P_g (resp. P_h) representation of g (resp. h) over $\mathbb{C}[r_{\mathcal{T}|t}]$ (resp. $\mathbb{C}[r_{\mathcal{T}|s}]$).

M2L symmetries

M2L $A_{t,s}$ between well-separated t and s. $A_{t,s}$ is $\mathcal{G}$-invariant:

$$\forall g, h \in \mathcal{G}, A_{g \cdot t, h \cdot s} = P_g \cdot A_{t,s} \cdot P_h,$$

P_g (resp. P_h) representation of g (resp. h) over $\mathbb{C}[r_{\mathcal{T}|t}]$ (resp. $\mathbb{C}[r_{\mathcal{T}|s}]$).

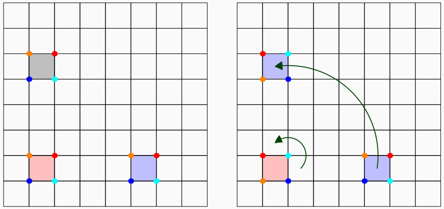


M2L symmetries

M2L $A_{t,s}$ between well-separated t and s. $A_{t,s}$ is $\mathcal{G}$-invariant:

$$\forall g, h \in \mathcal{G}, A_{g\cdot t, h\cdot s} = P_g \cdot A_{t,s} \cdot P_h,$$

P_g (resp. P_h) representation of g (resp. h) over $\mathbb{C}[r_{\mathcal{T}|t}]$ (resp. $\mathbb{C}[r_{\mathcal{T}|s}]$).

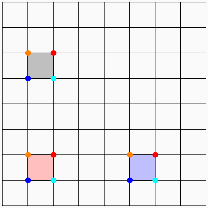


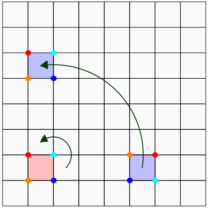
M2L symmetries

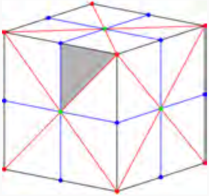
M2L $A_{t,s}$ between well-separated t and s. $A_{t,s}$ is $\mathcal{G}$-invariant:

$$\forall g, h \in \mathcal{G}, A_{g\cdot t, h\cdot s} = P_g \cdot A_{t,s} \cdot P_h,$$

P_g (resp. P_h) representation of g (resp. h) over $\mathbb{C}[r_{\mathcal{T}|t}]$ (resp. $\mathbb{C}[r_{\mathcal{T}|s}]$).





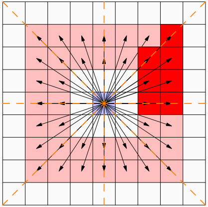


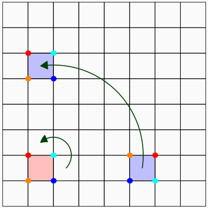
M2L symmetries

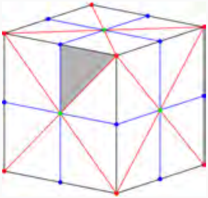
M2L $A_{t,s}$ between well-separated t and s. $A_{t,s}$ is $\mathcal{G}$-invariant:

$$\forall g, h \in \mathcal{G}, A_{g\cdot t, h\cdot s} = P_g \cdot A_{t,s} \cdot P_h,$$

P_g (resp. P_h) representation of g (resp. h) over $\mathbb{C}[r_{\mathcal{T}|t}]$ (resp. $\mathbb{C}[r_{\mathcal{T}|s}]$).





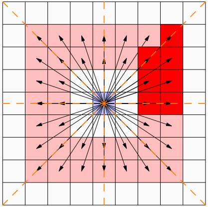


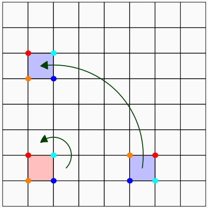
M2L symmetries

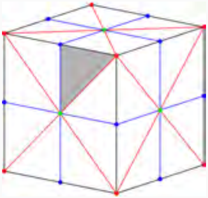
M2L $A_{t,s}$ between well-separated t and s. $A_{t,s}$ is $\mathcal{G}$-invariant:

$$\forall g, h \in \mathcal{G}, A_{g\cdot t, h\cdot s} = P_g \cdot A_{t,s} \cdot P_h,$$

P_g (resp. P_h) representation of g (resp. h) over $\mathbb{C}[r_{\mathcal{T}|t}]$ (resp. $\mathbb{C}[r_{\mathcal{T}|s}]$).





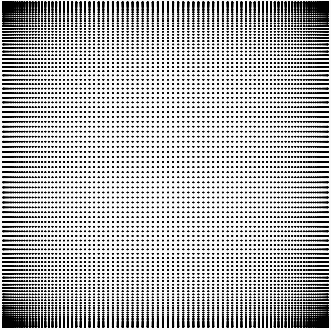


Same hyperoctahedral symmetry than M2L set!

Two-electron integrals

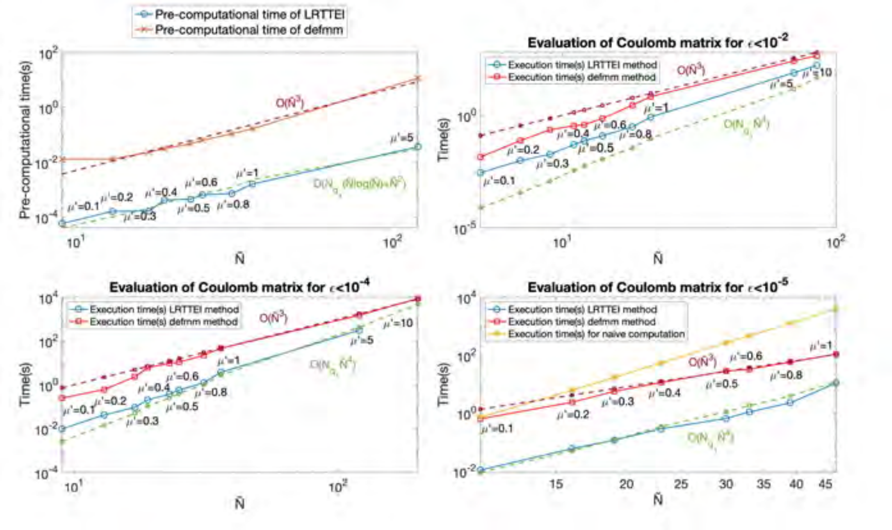
Problem: Evaluate fourth-order tensor $\mathbb{H} \in \mathbb{R}^{N\times N\times N\times N}$ (bottleneck in quantum chemistry):

$$\mathbb{H}_{i,j,k,l} := \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} G(\mathbf{x},\mathbf{y})\mu_i(\mathbf{x})\mu_j(\mathbf{x})\nu_k(\mathbf{y})\nu_l(\mathbf{y})d\mathbf{y}d\mathbf{x}, \quad G(\mathbf{x},\mathbf{y}) = \frac{2\int_0^{\mu|\mathbf{x}-\mathbf{y}|} e^{-t^2}dt}{\sqrt{\pi}|\mathbf{x}-\mathbf{y}|}.$$



Joint work with Siwar Baddredine and Laura Grigori

Two-electron integrals



Thank you for your attention!