A New FGT and the DMK Framework

Shidong Jiang

Computational Tools 2024 BIE workshop. Day 2, 6/11/24

Center for Computational Mathematics, Flatiron Institute, Simons Foundation

A new fast Gauss transform

- Consider the evaluation of the discrete and continuous Gauss transforms:

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i - \boldsymbol{y}_j; \delta) q_j, \qquad i = 1, \ldots N,$$

$$u(\boldsymbol{x}) = \int_B G(\boldsymbol{x} - \boldsymbol{y}; \delta) \sigma(\boldsymbol{y}) d\boldsymbol{y}, \qquad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d,$$

  where the Gaussian kernel is given by $G(\boldsymbol{x} - \boldsymbol{y}; \delta) = e^{-|\boldsymbol{x}-\boldsymbol{y}|^2/(4\delta)}$.

- There are many fast Gauss transforms. See, for example, Greengard and Strain (1991), Greengard and Sun (1998), Lee, Gray and Moore (2006), Veerapaneni and Biros (2008), Spivak, Veerapaneni and Greengard (2010), Wang and Greengard (2018), Jiang and Greengard (2022). Most of them use Hermite expansions, local expansions, and plane wave expansions.
- Our discussion follows from a recent paper Greengard, Jiang, Rachh, and Wang, *SIAM Review* **66** (2024), which relies entirely on the Fourier approximation of the Gaussian.

Properties of the Gaussian

- Gaussians have many nice properties. For example, a Gaussian in higher dimensions is the product of one-dimensional Gaussians:

$$G(\boldsymbol{x}-\boldsymbol{y};\delta)=\prod_{i=1}^{d}G(x_i-y_i;\delta).$$

And the Fourier transform of a Gaussian is again a Gaussian:

$$G(x;\delta)=e^{-x^2/4\delta}=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\widehat{G}(k;\delta)e^{ikx}dk,\quad \widehat{G}(k;\delta)=\sqrt{2\delta}e^{-k^2\delta}.$$

- Numerically, Gaussians are compactly supported in both physical and Fourier spaces to any desired precision. In physical space,

$$G(\boldsymbol{x};\delta)\leq\epsilon,\quad \text{for}\quad |\boldsymbol{x}|\leq D_0=2\sqrt{\delta\log(1/\epsilon)},$$

where D_0 is referred to as the "cutoff" length.

FLATIRON INSTITUTE

Non-uniform fast Fourier transforms

- Given data points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \in R^d$, the *type-1 NUFFT* computes sums of the form

$$F(\boldsymbol{k}) = \sum_{j=1}^{N} f_j e^{\pm i \boldsymbol{k} \cdot \boldsymbol{x}_j},$$

  where $\boldsymbol{k} \in \mathcal{I} = \{-n_f/2, \ldots, n_f/2 - 1\}^d$. That is, it computes the discrete Fourier transform at equispaced modes from function values at arbitrary data points.
- Its adjoint, called the *type-2 NUFFT*, computes the value of a truncated Fourier series at arbitrary target points:

$$f(\boldsymbol{x}_j) = \sum_{\mathbf{k} \in \mathcal{I}} F(\boldsymbol{k}) e^{\pm i \boldsymbol{k} \cdot \boldsymbol{x}_j}.$$

- The cost of both type-1 and type-2 NUFFTs is $O(N_F \log N_F) + O(N)$ ($N_F = n_f^d$). The constant implicit in the notation $O(N)$ is roughly $(D+1)^d$, where D is the number of desired digits of accuracy and d is the dimension of the problem. See, for example, Dutt and Rokhlin (1993, 1995), Greengard and Lee (2004), Barnett, Magland, and af Klinteberg (2019).

Summary of the new FGT

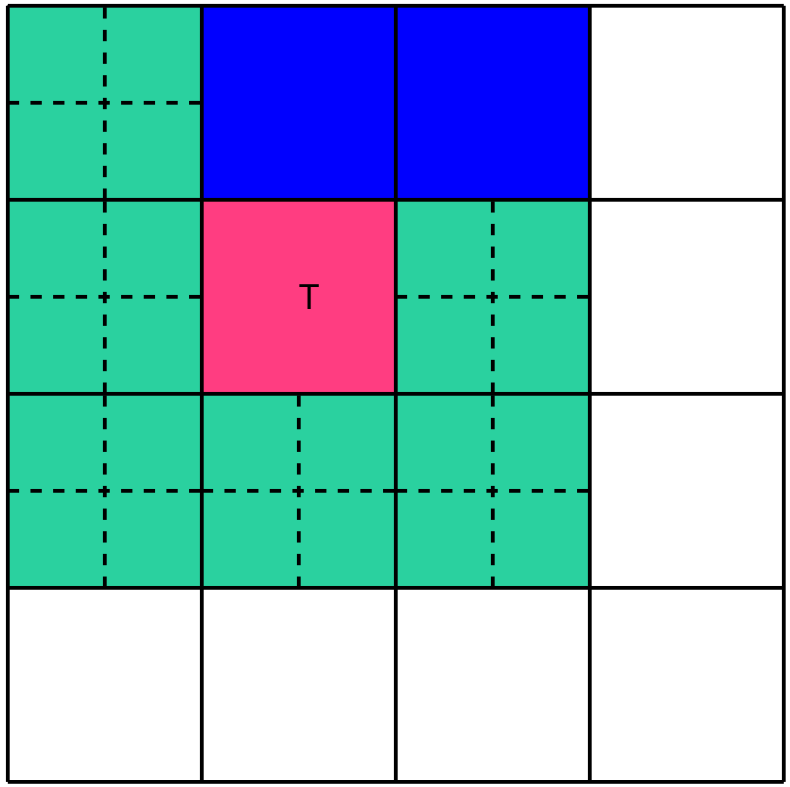


Figure: The new FGT. At the cutoff level, source boxes containing more than n_s particles use plane waves to interact with the target box (in pink), while source boxes containing less than or equal to n_s particles (in blue) interact with the target box directly. NUFFTs are used to reduce the constant in $O(N)$.

Fourier spectral approximation of the Gaussian

- One may apply truncated trapezoidal rule to discretize the Fourier representation of the Gaussian. This is spectrally accurate due to the fast decay of the Gaussian.
- In one dimension, for any $R \geq D_0$,

$$\left| G(x;\delta) - \frac{h\sqrt{\delta}}{2\sqrt{\pi}} \sum_{m=-n_f/2}^{n_f/2-1} e^{-m^2h^2\delta} e^{imhx} \right| \leq \epsilon, \quad |x| \leq R.$$

- Here, the stepsize $h = 2\pi\sqrt{\delta}/(R + D_0)$, and the number of Fourier modes $n_f = O(\log(1/\epsilon))$. For example, n_f is about 30 with $\epsilon = 10^{-6}$ and 56 with $\epsilon = 10^{-12}$ for $R = 2D_0$. In higher dimensions, the total number of Fourier modes $N_F = n_f^d = O(\log^d(1/\epsilon))$.

FLATIRON INSTITUTE

Discrete FGT

- When the number of Fourier modes N_F is independent of number of particles N, as is the case for Gaussians within $2D_0$ distance in each dimension, both NUFFTs are in fact linear algorithms w.r.t. N.
- We construct an adaptive tree so that D_0 is the exact dimension of a leaf box at some level of refinement. That level is referred to as the "cutoff" level. It is clear that one only needs to compute the interactions between nearest neighbors for boxes at or above the "cutoff" level.
- In order to avoid too many boxes in the tree, boxes are refined until each leaf node either has fewer than n_s sources and targets or the tree has reached the cutoff level. Here n_s is a user-supplied constant.
- For any leaf source box at or above the "cutoff" level, we compute the interaction between the source box and its nearest neighbors directly.

Discrete FGT

- For any nonleaf source box at the "cutoff" level that contains many particles, we use Fourier spectral approximation of the Gaussian. Let J denote the set of particle indices in the nonleaf source box with $|J| = N_s$, I denote the set of particle indices in its neighboring target box with $|I| = N_t$. Then the interaction between these two boxes can be computed as follows:

$$
\begin{aligned}
u_I &= \sum_J G(\boldsymbol{x}_I - \boldsymbol{y}_J; \delta) q_J \approx \sum_J \sum_{n=1}^{N_F} D(\boldsymbol{k}_n) e^{i\boldsymbol{k}_n \cdot (\boldsymbol{x}_I - \boldsymbol{y}_J)} q_J \\
&= \sum_{n=1}^{N_F} e^{i\boldsymbol{k}_n \cdot \boldsymbol{x}_I} D(\boldsymbol{k}_n) \sum_J e^{-i\boldsymbol{k}_n \cdot \boldsymbol{y}_J} q_J \\
&:= \mathcal{F}_2 D \mathcal{F}_1 q_J,
\end{aligned}
$$

where $\mathcal{F}_2$ is an $N_t \times N_F$ Fourier transform matrix whose nth column is $e^{i\boldsymbol{k}_n \cdot \boldsymbol{x}_I}$, D is an $N_F \times N_F$ diagonal matrix whose diagonal entries are $\frac{h\sqrt{\delta}}{2\sqrt{\pi}} e^{-|\boldsymbol{k}_n|^2 h^2 \delta}$, and $\mathcal{F}_1$ is an $N_F \times N_s$ matrix whose nth row is $e^{-i\boldsymbol{k}_n \cdot \boldsymbol{y}_J}$.

FLATIRON INSTITUTE

Discrete FGT

- The action of $\mathcal{F}_1$ can be carried out via a type-1 NUFFT in $O(N_F \log N_F + N_s)$ work. The action of D can be carried out in $O(N_F)$ work. And the action of $\mathcal{F}_2$ can be carried out via a type-2 NUFFT in $O(N_F \log N_F + N_t)$ work. Thus, the total work is $O(N_t + N_s)$, instead of $O(N_t \cdot N_s)$.
- One may also merge all "outgoing" plane wave expansions from neighboring source boxes to a single "incoming" plane wave expansion on the target box via *diagonal translation*. It is easy to see that the total cose is $O(N)$ instead of $O(N^2)$.
- Note that NUFFTs are used *locally* on each nonleaf source box at the "cutoff" level, and that the purpose of NUFFTs is to reduce the constant in the $O(N)$ complexity. To be more precise, one could just calculate the Fourier transform directly to obtain an algorithm with linear complexity for the Gauss transform.

Continuous FGT on a box

- For the box FGT, the so-called direct interactions can be further accelerated via tensor product transform, using the fact that Gaussians in high dimensions are the product of Gaussians in one dimension.
- In three dimensions, both source index i and target index j are shorthand for multi-indices, i.e., $i = (i_1, i_2, i_3)$ and $j = (j_1, j_2, j_3)$ with $i_1, i_2, i_3, j_1, j_2, j_3 = 1, \ldots, p$ for each box.
- We have

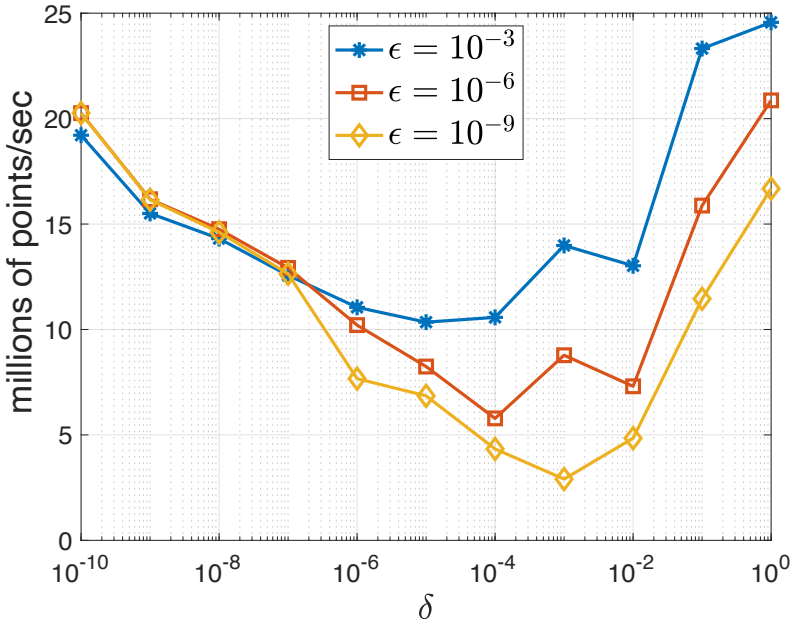
$$\sum_j G(\boldsymbol{x}_i, \boldsymbol{y}_j)\rho_j = \sum_{j_1=1}^{p}\sum_{j_2=1}^{p}\sum_{j_3=1}^{p} G(\boldsymbol{x}_i, \boldsymbol{y}_j)\rho_j$$

$$= \sum_{j_1=1}^{p} G(x_{i_1}, y_{j_1}) \sum_{j_2=1}^{p} G(x_{i_2}, y_{j_2}) \sum_{j_3=1}^{p} G(x_{i_3}, y_{j_3})\rho_{j_1 j_2 j_3}$$

  with $\rho_j = w_j\sigma_j$ and w_j proper quadrature weights.

- It is easy to see that the cost is $O(p^4)$ instead of $O(p^6)$. In general, the work is $O(dp^{d+1})$ in $\mathbb{R}^d$ instead of $O(p^{2d})$. Once again, the purpose is to reduce the constant in $O(N)$.

Performance of the box FGT



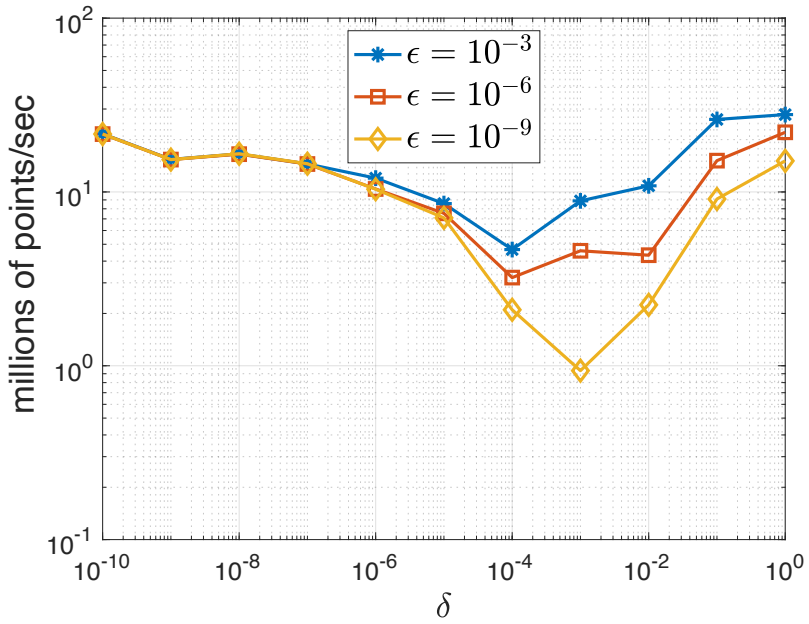


Figure: Throughput of the new box FGT measured in million points per second as a function of Gaussian variance: (left) two dimensions, (right) three dimensions.

FLATIRON INSTITUTE

What about other kernels?

Many kernels admit an integral representation in terms of Gaussians.

- Green's function for the Laplace operator:

$$-\frac{1}{2\pi}\log r = \frac{1}{4\pi}\int_0^\infty \frac{1}{t} e^{-r^2/(4t)}\, dt,$$

$$\frac{1}{4\pi r} = \frac{1}{8\pi^{3/2}}\int_0^\infty \frac{1}{t^{3/2}} e^{-r^2/(4t)}\, dt.$$

This is well-known connection between the Laplace operator and the heat operator. That is, the solution to the Poisson equation is the solution to the initial value problem for the heat equation with the same forcing term as $t \to \infty$.

- The general power function in $\mathbb{R}^d$:

$$\frac{1}{r^\alpha} = \frac{1}{\Gamma(\alpha/2)}\int_0^\infty e^{-r^2 t} t^{\alpha/2-1}\, dt.$$

- Green's function for the Yukawa operator:

$$\frac{1}{2\pi} K_0(\lambda r) = \frac{1}{2\pi}\int_0^\infty e^{-r^2 t^2 - \frac{\lambda^2}{4t^2}} \frac{dt}{t}, \quad \boldsymbol{x} \in \mathbb{R}^2,$$

$$\frac{1}{4\pi}\frac{e^{-\lambda r}}{r} = \frac{1}{2\pi^{3/2}}\int_0^\infty e^{-r^2 t^2 - \frac{\lambda^2}{4t^2}}\, dt, \quad \boldsymbol{x} \in \mathbb{R}^3.$$

The DMK framework for general kernels

- Dual-space Multilevel Kernel-splitting Framework for computing sums of the form

$$u_i = \sum_{j=1}^{N} K(\boldsymbol{x}_i, \boldsymbol{y}_j) q_j, \qquad i = 1, \ldots N.$$

- Consider the integral representation of the 3D Laplace kernel. Introduce the change of variable $t = e^s$ to obtain an integral representation from $-\infty$ to $+\infty$. Then the spectrally accurate truncated trapezoidal rule leads to

$$\frac{1}{4\pi r} \approx \sum_{i=1}^{N_G} w_i e^{-r^2/(4\delta_i)}, \qquad r \in (r_0, R),$$

  where the number of Gaussians $N_G = O(\log(1/\epsilon) \cdot \log(R/r_0))$. For example, $N_G \approx 70$ for $\epsilon = 10^{-10}$ and $R/r_0 = 10^5$.

- One may apply the FGT for each Gaussian and asymptotic expansions or direct calculations to handle the $r < r_0$ part to compute the convolution with the 3D Laplace kernel. The total cost is $O(N_G N)$.

The DMK framework

- One may group Gaussians level by level for any given adaptive tree to reduce the cost from $O(N_G N)$ to $O(N_{\mathrm{lev}} N)$, where N_{lev} is the total number of levels in the adaptive tree. Since $N_{\mathrm{lev}} = O(\log N)$, we obtain an $O(N \log N)$ algorithm for general kernels.
- Standard upward (from child to parent) and downward (from parent to child) passes in the FMM are used so that one only needs to access sources and targets once, rendering interactions at coarser levels to box-to-box interactions instead of particle-to-box interactions.
- In the DMK framework, this is basically polynomial interpolation and anterpolation for smooth functions.
- The complexity is reduced from $O(N \log N)$ to $O(N)$ (excluding the time on sorting).

FLATIRON INSTITUTE

Continuous multilevel kernel splitting

- In physical space, we have

$$\begin{aligned}\frac{1}{4\pi r} &= \frac{1}{8\pi^{3/2}} \int_0^\infty \frac{1}{t^{3/2}} e^{-r^2/(4t)} dt \\ &= \frac{1}{8\pi^{3/2}} \left(\int_0^{\delta_L} + \ldots + \int_{\delta_0}^\infty \right) \frac{1}{t^{3/2}} e^{-r^2/(4t)} dt \\ &:= M_0(r) + \sum_{l=0}^{L-1} D_l(r) + R_L(r), \quad L = 0, \ldots, L_{\max}.\end{aligned}$$

- Here the mollified kernel M_0, the difference kernels D_l, and the residual kernels R_L are defined by the formulas

$$M_0(r) = \frac{\operatorname{erf}(2\sqrt{\delta_0} r)}{4\pi r}, \quad R_L(r) = \frac{\operatorname{erfc}(2\sqrt{\delta_L} r)}{4\pi r},$$
$$D_l(r) = \frac{\operatorname{erf}(2\sqrt{\delta_l} r) - \operatorname{erf}(2\sqrt{\delta_{l+1}} r)}{4\pi r},$$

where erf and erfc are the error and complementary error functions

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad \operatorname{erfc}(x) = 1 - \operatorname{erf}(x).$$

Dual-space Multilevel Kernel-splitting

- The mollified kernel M_0 is smooth in physical space, but its Fourier transform is singular at the origin.
- We replace the mollified kernel M_0 by the windowed kernel W_0 whose Fourier transform is given by the formula

$$\widehat{W_0}(\boldsymbol{k}) = 2\left(\frac{\sin(\sqrt{C}|\boldsymbol{k}|)}{|\boldsymbol{k}|}\right)^2 e^{-|\boldsymbol{k}|^2\delta_0}, \qquad C = \sqrt{3} + \log(1/\epsilon)/(2\delta_0),$$

  One can show that $|W_0(r) - M_0(r)| \le \epsilon \frac{1}{4\pi r}$ for $r \le \sqrt{3}$.
- Thus, we have

$$\frac{1}{4\pi r} = W_0(r) + \sum_{l=0}^{L-1} D_l(r) + R_L(r), \quad L = 0, \ldots, L_{\max}$$

$$\frac{1}{|\boldsymbol{k}|^2} = 2\left(\frac{\sin(\sqrt{C}|\boldsymbol{k}|)}{|\boldsymbol{k}|}\right)^2 e^{-|\boldsymbol{k}|^2\delta_0} + \frac{e^{-|\boldsymbol{k}|^2\delta_1} - e^{-|\boldsymbol{k}|^2\delta_0}}{|\boldsymbol{k}|^2} + \cdots + \frac{1 - e^{-|\boldsymbol{k}|^2\delta_L}}{|\boldsymbol{k}|^2}.$$

FLATIRON INSTITUTE

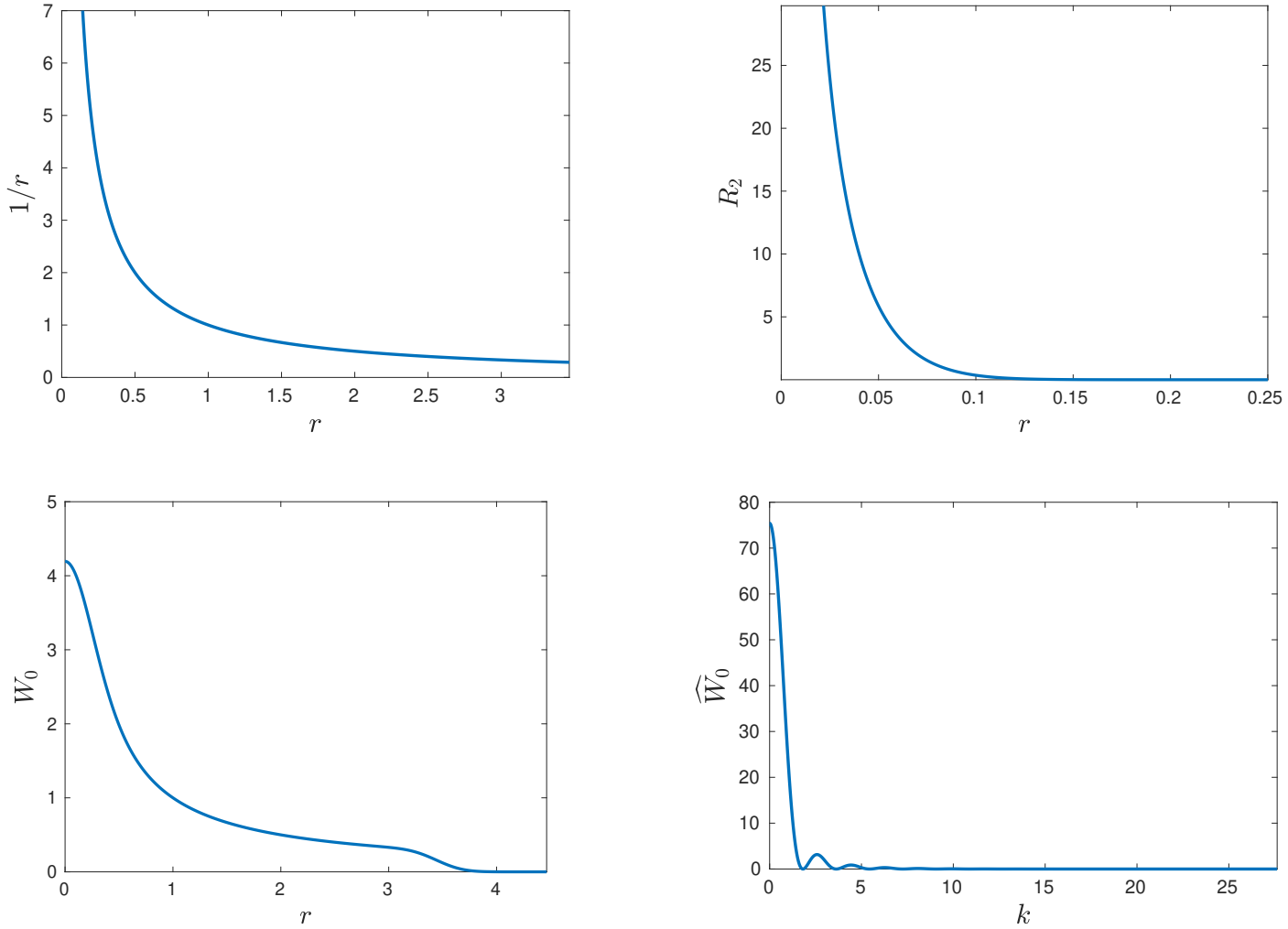


Figure: The dual-space splitting of the $1/r$ kernel using Gaussians on a leaf box at level 2 with six digits of accuracy: $1/r = W_0(r) + D_0(r) + D_1(r) + R_2(r)$ for $r \leq \sqrt{3}$. Top row: left - the original kernel $1/r$; right - the residual kernel $R_2(r)$ that is numerically supported on $[0, 1/4]$. Bottom row: left - the windowed kernel $W_0(r)$; right - its Fourier transform.

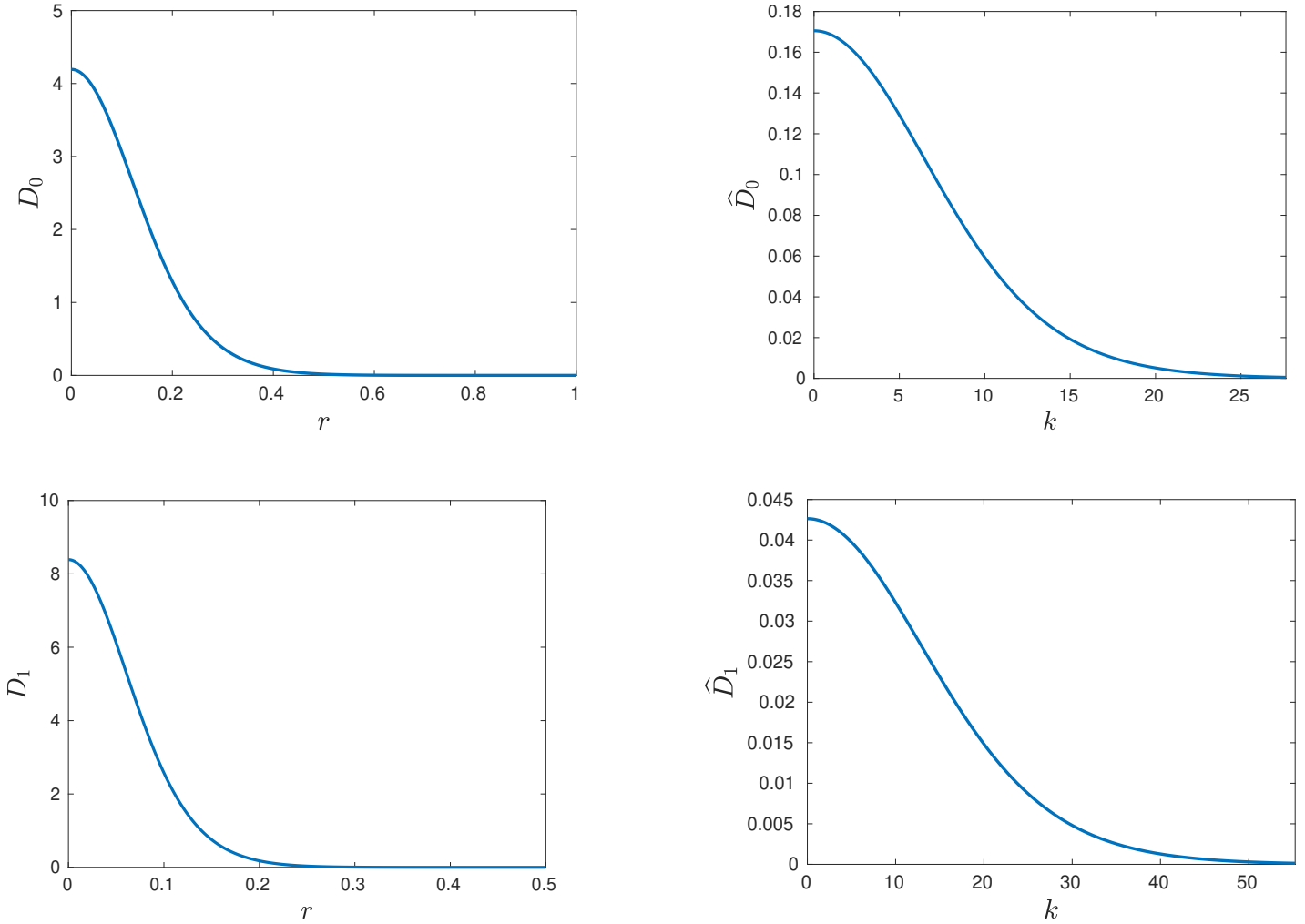


Figure: Top row: left - the difference kernel $D_0(r)$ that is numerically supported on $[0, 1]$; right - its Fourier transform. Bottom row: left - the difference kernel $D_1(r)$ that is numerically supported on $[0, 1/2]$; right - its Fourier transform. Note that the difference kernels D_1 is exactly a rescaled version of D_0, with one half the support and twice the frequency content.

Key algorithmic steps of the DMK framework

- **Upward pass**:
  - Form proxy charges for each leaf box from sources. $O(p^3 N)$
  - Form proxy charges for the parent box by merging proxy charges from its children. $O(p^4 N_B)$
- **Downward pass**:
  - Form the outgoing plane wave expansion from proxy charges. $O(pn_f^3 N_B)$
  - Diagonally translate outgoing plane wave expansions into incoming plane wave expansions. $O(3^3 n_f^3 N_B)$
  - Evaluate proxy potential from incoming plane wave expansions. $O(pn_f^3 N_B)$
  - Split the proxy potential from the parent box to its children. $O(p^4 N_B)$
  - Evaluate the far-field potential at targets in leaf boxes via the proxy potential. $O(p^3 N)$
- **Direct interaction**:
  - Evaluate the local potential directly. $O(3^3 n_s N)$
- **Total cost**: $N_B \leq 2N/n_s \rightarrow O(N)$.

FLATIRON INSTITUTE

The DMK framework

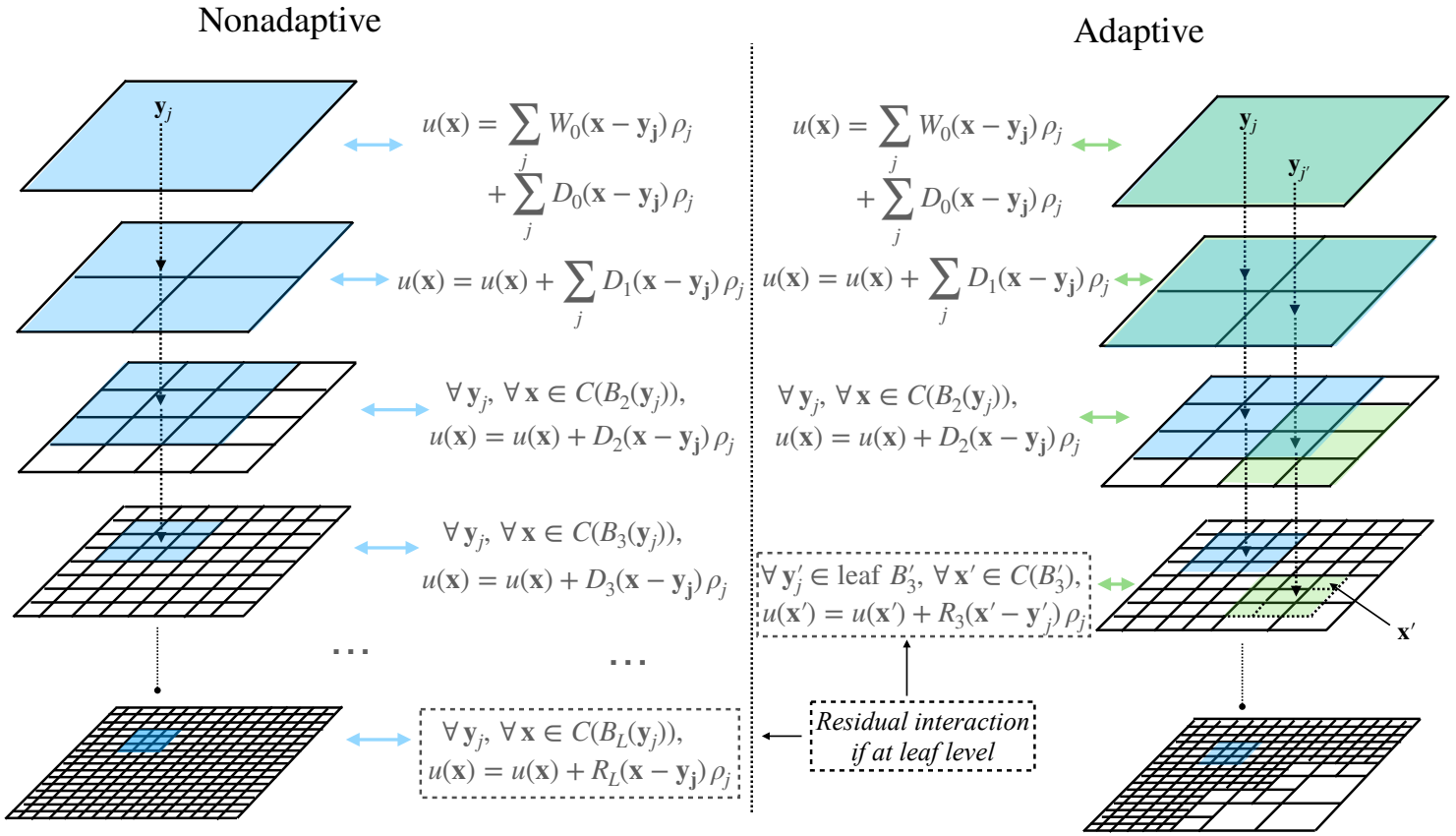


Figure: The DMK framework is essentially the new FGT on every level, where each level has its own difference kernel, i.e., a collection of Gaussians with different variances, interacting via plane wave approximation (with the windowed kernel at the root level as well). At the finest level, direct interactions are computed with the residual kernel R_L for targets within B's colleagues.

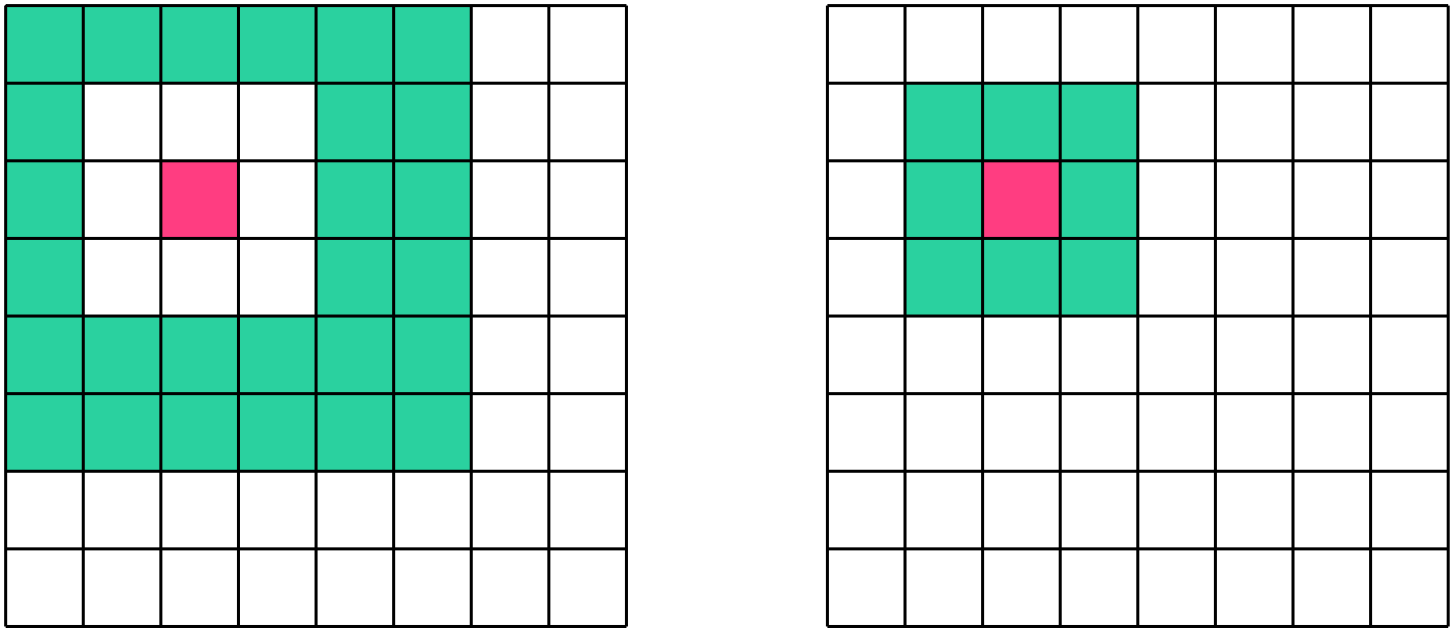
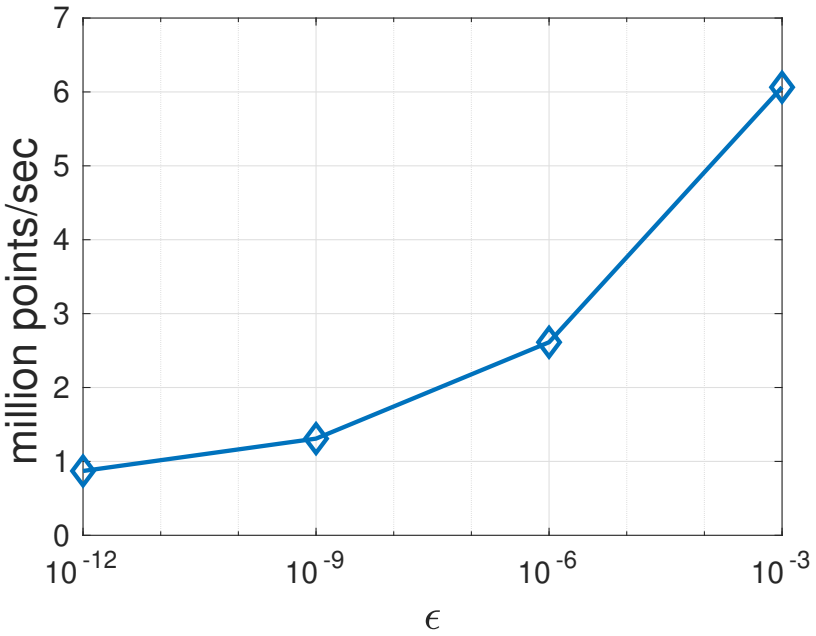


Figure: Interaction lists in green. Left: the interaction list in the FMM that involves up to $6^d - 3^d$ (189 in 3D) boxes. Right: the interaction list in the DMK. each box interacts with up to 3^d (27 in 3D) nearest neighbors at the same level.

The interactions in the DMK only need to be carried out at levels coarser than the leaf boxes. For elliptic PDE kernels, the length of multipole and local expansions in the classic FMM (Greengard and Rokhlin (1987), Cheng, Greengard, and Rokhlin (1999)) is p^{d-1}, while the length of plane wave expansions in the DMK is n_f^d with $n_f \approx p$ for six digits of accuracy.

Performance of the box DMK



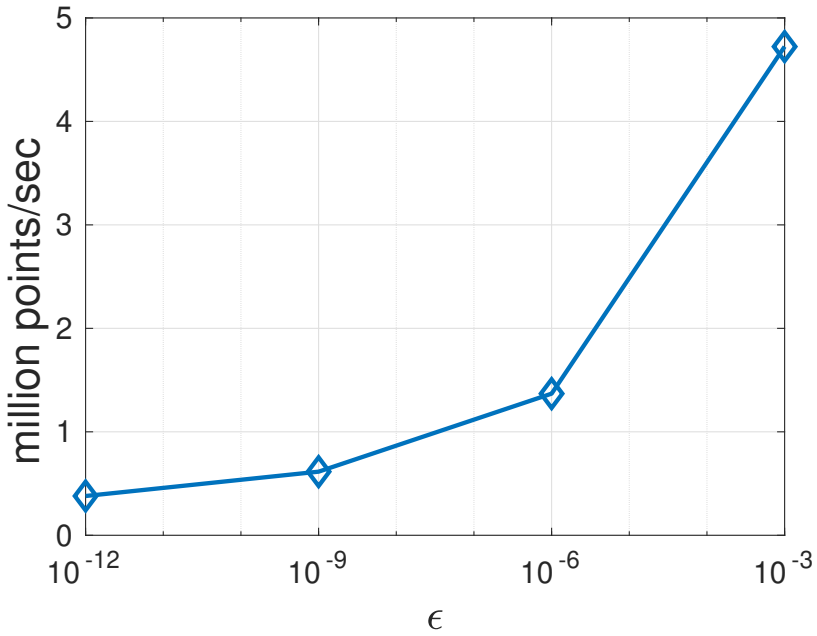


Figure: Average throughput of DMK for the $\sqrt{\Delta}$ kernel in two and three dimensions.

FLATIRON INSTITUTE

Summary

- The DMK (dual-space multilevel kernel-splitting) framework is essentially a simplified FMM. The simplification comes from smooth kernel splitting in the DMK, instead of sharp truncation in the FMM, between the direct interaction part and all other parts.
- Fourier transform serves as the main analytical tool. But one does not need the FFT to achieve linear complexity, since the plane wave expansion length, about $O(\log^d(1/\epsilon))$, depends only on the desired precision for the windowed and difference kernels. Instead, the tensor product transform is used to reduce the constant in $O(N)$.
- There is no far field. We only need to interact with our neighbors in a multiscale world.

FLATIRON INSTITUTE

Resources

- *A Dual-space Multilevel Kernel-splitting Framework for Discrete and Continuous Convolution*, S. Jiang and L. Greengard, *arXiv:2308.00292* (2023).
- *A New Version of the Adaptive Fast Gauss Transform for Discrete and Continuous Sources*, L. Greengard, S. Jiang, M. Rachh, and J. Wang, *SIAM Review* **66**, Iss. 2 (2024).
- *Approximating the Gaussian as a Sum of Exponentials and its Applications to the Fast Gauss Transform*, S. Jiang and L. Greengard, *Communications in Computational Physics* **31** (1), 1-26 (2022). Slightly more efficient FGT in 1D can be constructed using a sum-of-exponentials approximation of the Gaussian in this paper.
- *An accurate and efficient scheme for function extensions on smooth domains*, C. Epstein, F. Fryklund, and S. Jiang, *arXiv:2206.11318* (2023). Solves elliptic BVPs with an inhomogeneous term in complex geometry using a simple function extension scheme, FFT, NUFFT, and fmm3dbie.
- DMK code at `https://github.com/flatironinstitute/dmk`. Robert Blackwell, Dhairya Malhotra, and SJ are developing the HPC version. Coming soon...