A Multiscale Butterfly Algorithm for Fourier Integral Operators

Yingzhou (Ryan) Li, Haizhao Yang, Lexing Ying

ICME & Math, Stanford

November 21, 2014

Outline

Fourier Integral Operators

$$(\mathcal{L}f)(x) = \int_{\mathbb{R}^d} a(x,\xi) e^{2\pi\imath\Phi(x,\xi)} \widehat{f}(\xi) d\xi,$$

where

- $\Phi(x,\xi)$ is a phase function assumed to be smooth in (x,ξ) for $\xi \neq 0$ and obeys a homogeneity condition of degree 1 in ξ, namely, $\Phi(x,\lambda\xi) = \lambda\Phi(x,\xi)$ for each $\lambda > 0$;
- $a(x,\xi)$ is a smooth amplitude function;
- $\widehat{f}$ is the Fourier transform of f defined by

$$\widehat{f}(\xi) = \int_{\mathbb{R}^d} f(x) e^{-2\pi\imath x\cdot\xi} dx.$$

Fourier Integral Operators

The space domain and frequency domain are discretized as

$$X = \left\{ x = (\frac{n_1}{N}, \frac{n_2}{N}), 0 \le n_1, n_2 < N \text{ with } n_1, n_2 \in \mathbb{Z} \right\},$$

$$\Omega = \left\{ \xi = (n_1, n_2), -\frac{N}{2} \le n_1, n_2 < \frac{N}{2} \text{ with } n_1, n_2 \in \mathbb{Z} \right\}.$$

The discrete FIO is defined as follows:

$$(Lf)(x) = \sum_{\xi \in \Omega} e^{2\pi\imath\Phi(x,\xi)} \widehat{f}(\xi)$$

for every $x \in X$.

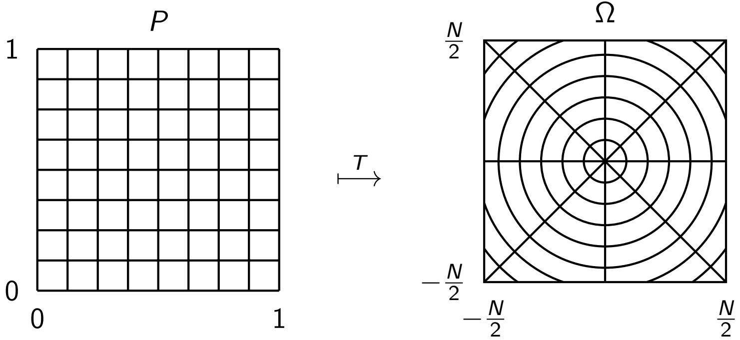
Related Works

- $\mathcal{O}(N^4)$ algorithm
  - Dense matrix-vector multiplication
- $\mathcal{O}(N^3 \log N)$ algorithm
  - Precomputation of Butterfly Factorization [Li et al., 2014]
- $\mathcal{O}(N^{2.5} \log N)$ algorithm
  - Wedge separation algorithm [Candès et al., 2007]
- $\mathcal{O}(N^2 \log N)$ algorithm
  - Polar butterfly algorithm [Candès et al., 2009, Poulson et al., 2014]
  - Multiscale butterfly algorithm [Yang et al., 2014]
  - Application of Butterfly Factorization [Li et al., 2014]

Polar-Cartesian coordinate transformation

Let $P = [0,1)^2$, a polar-Cartesian coordinate transformation $T : P \mapsto \Omega$ is defined as

$$\xi = (\xi_1, \xi_2) = \frac{\sqrt{2}}{2} N p_1 e^{2\pi\imath p_2}.$$



Low-rank approximation in polar grid

Theorem ([Candès et al., 2009])

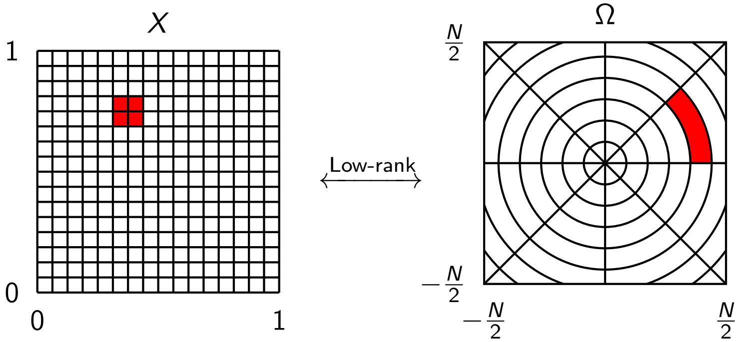
Suppose $\Phi(x,\xi)$ *is a standard phase function in an FIO. Let* A *and* B' *be two boxes in* X *and* P*, respectively, obeying* $w(A)w(B') \lesssim 1/N$*. For any positive* $\epsilon \leq \epsilon_0$ *and* $N \geq N_0$*, where* ϵ_0 *and* N_0 *are some positive constants, there exists an approximation satisfying*

$$\left| e^{2\pi\imath\Phi(x,\xi)} - \sum_{t=1}^{t_\epsilon} \alpha_t^{AB'}(x)\beta_t^{AB'}(\xi) \right| \leq \epsilon$$

for $x \in A$ *and* $\xi \in T(B')$ *with* $t_\epsilon \lesssim \log^4(\frac{1}{\epsilon})$*.*

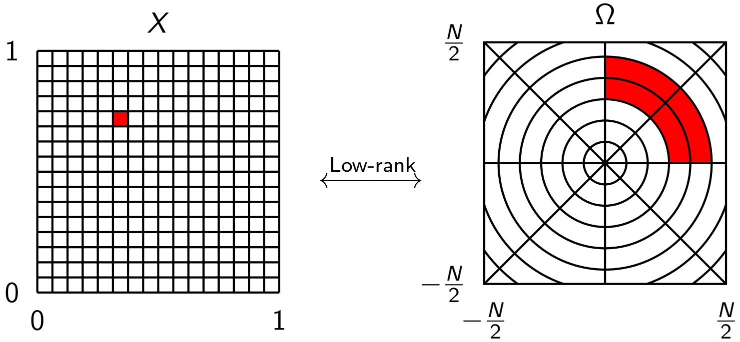
Low-rank approximation in polar grid

Assume $N = 64$.



Low-rank approximation in polar grid

Assume $N = 64$.



Low-rank approximation in Cartesian grid

Theorem

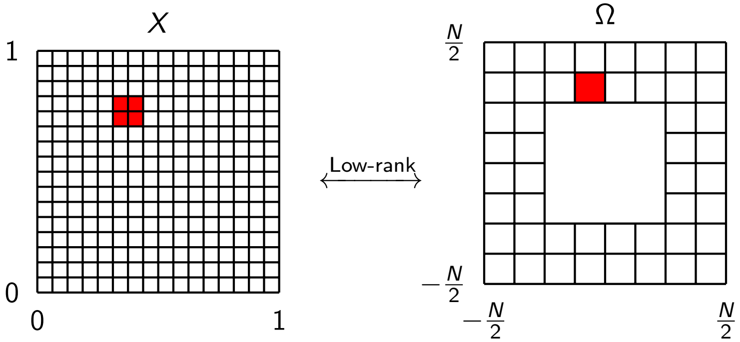
Suppose $\Phi(x,\xi)$ is a standard phase function in an FIO. Let A and B be two boxes in X and Ω, respectively, obeying $w(A)w(B) \leq 1$ and $dist(B,0) \geq \frac{M}{4}$. For any positive $\epsilon \leq \epsilon_0$ and $M \geq M_0$, where ϵ_0 and M_0 are some positive constants, there exists an approximation satisfying

$$\left| e^{2\pi \imath \Phi(x,\xi)} - \sum_{t=1}^{t_\epsilon} \alpha_t^{AB}(x)\beta_t^{AB}(\xi) \right| \leq \epsilon$$

for $x \in A$ and $\xi \in B$ with $t_\epsilon \lesssim \log^4(\frac{1}{\epsilon})$.

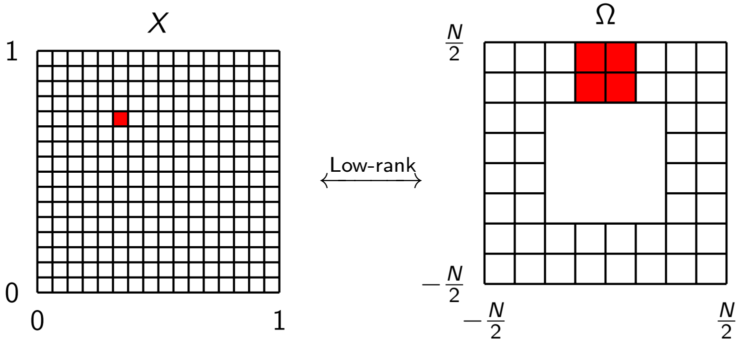
Low-rank approximation in Cartesian grid

Assume $N = 64$.



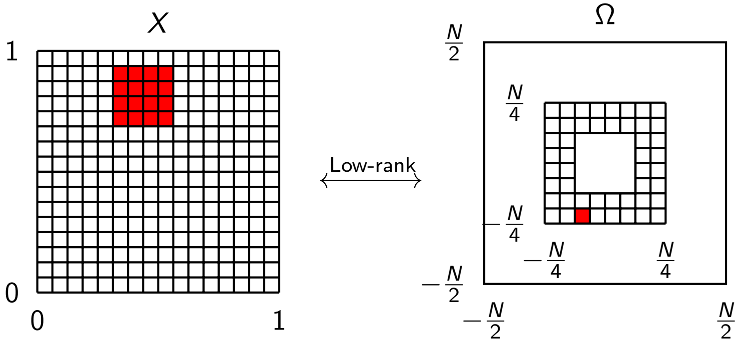
Low-rank approximation in Cartesian grid

Assume $N = 64$.



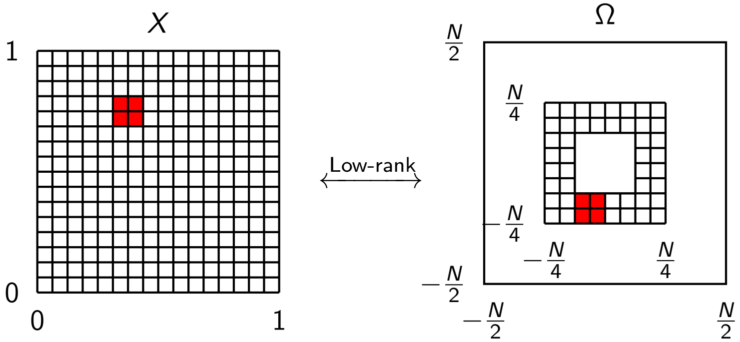
Low-rank approximation in Cartesian grid

Assume $N = 64$.



Low-rank approximation in Cartesian grid

Assume $N = 64$.



Cartesian Butterfly Algorithm: Preliminaries

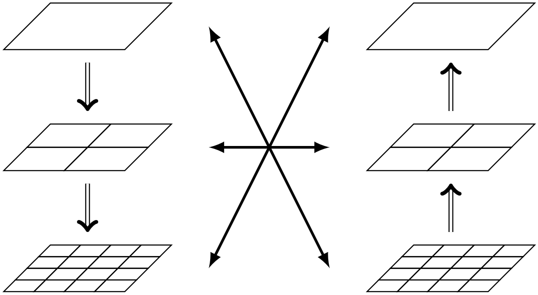


Figure : Hierarchical domain trees of the 2D butterfly algorithm. Left: T_X for the spacial domain X. Right: T_Ω for the frequency domain Ω. The interaction between sub-domain $A \subset X$ and $B \subset \Omega$ starts from the root of T_X and the leaf of T_Ω.

Cartesian Butterfly Algorithm: Preliminaries

Using Chebyshev interpolation, we have two group of low-rank expansion functions: when $w(B) \leq w(A)$, we use

$$\alpha_s^{AB}(x) = e^{2\pi\imath\Phi(x,\xi_s^B)},$$
$$\beta_s^{AB}(\xi) = e^{-2\pi\imath\Phi(c(A),\xi_s^B)} M_s^B(\xi) e^{2\pi\imath\Phi(c(A),\xi)};$$

and when $w(B) > w(A)$, we use

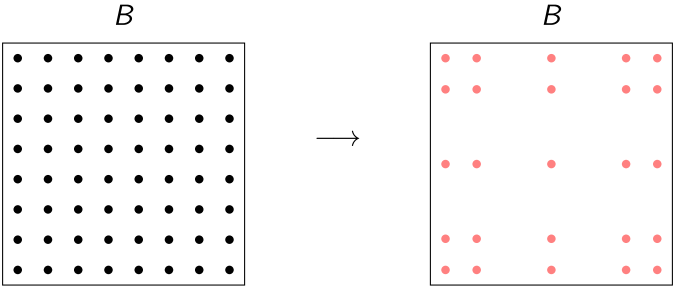
$$\alpha_t^{AB}(x) = e^{2\pi\imath\Phi(x,c(B))} M_t^A(x) e^{-2\pi\imath\Phi(x_t^A,c(B))},$$
$$\beta_t^{AB}(\xi) = e^{2\pi\imath\Phi(x_t^A,\xi)}.$$

$M_s^B(\xi)$ and $M_t^A(x)$ are Lagrange interpolation polynomials on the Chebyshev grid adapted to the box B and box A respectively.

Cartesian Butterfly Algorithm: Initialization

Set A to be the root of T_X. For each leaf box $B \in T_\Omega$, we define the expansion weights

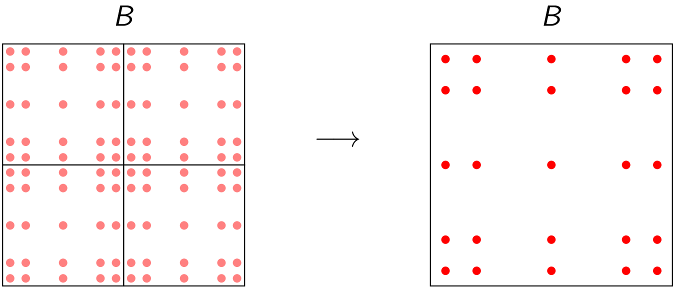
$$\delta_s^{AB} \approx \sum_{\xi \in B} \beta_s^{AB}(\xi) g(\xi) = e^{-2\pi\imath\Phi(c(A),\xi_s^B)} \sum_{\xi \in B} \left(M_s^B(\xi) e^{2\pi\imath\Phi(c(A),\xi)} \widehat{f}(\xi) \right).$$



Cartesian Butterfly Algorithm: Recursion

For each $l \leq \log(N)/2$, we recurse on the domain pair (A, B) and define the new expansion weights as

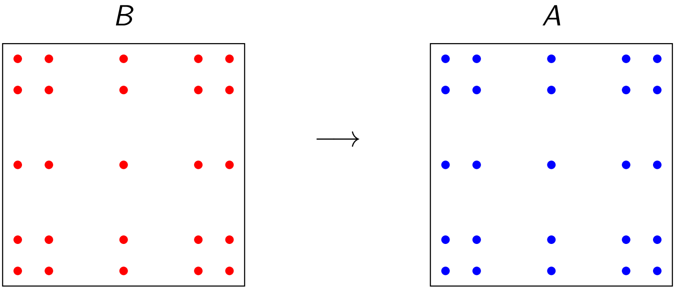
$$\delta_s^{AB} = e^{-2\pi\imath\Phi(c(A),\xi_t^B)} \sum_{c,s'} M_s^B(\xi_{s'}^{B_c}) e^{2\pi\imath\Phi(c(A),\xi_{s'}^{B_c})} \delta_{s'}^{A_p B_c}.$$



Cartesian Butterfly Algorithm: Switch

For $l = \log(N)/2$ and the same domain pairs (A, B) in the last step, we define new expansion weights as

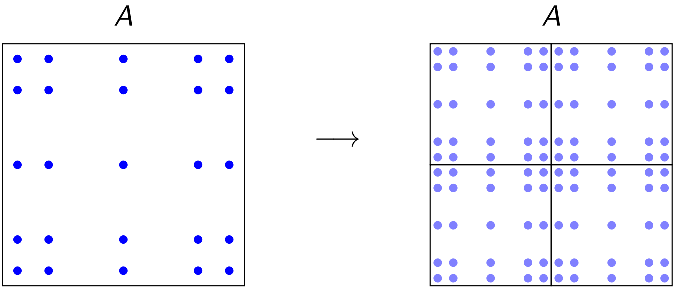
$$\delta_t^{AB} = \sum_s e^{2\pi\imath\Phi(x_t^A,\xi_s^B)}\delta_s^{AB}.$$



Cartesian Butterfly Algorithm: Recursion

For $l \geq \log(N)/2 + 1$, we recurse on the domain pair (A, B) and define the new expansion weights as

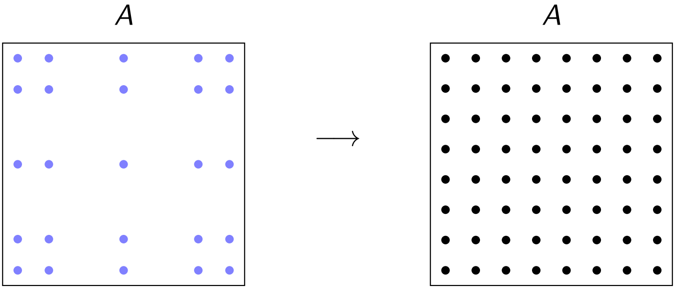
$$\delta_t^{AB} = \sum_c e^{2\pi\imath\Phi(x_t^A, c(B_c))} \sum_{t'} \left(L_{t'}^{A_p}(x_t^A) e^{-2\pi\imath\Phi(x_{t'}^{A_p}, c(B_c))} \delta_{t'}^{A_p B_c} \right).$$



Cartesian Butterfly Algorithm: Termination

Finally, for $l = \log(N)$ and set B to be the root box of T_Ω. For each leaf box $A \in T_X$, we obtain

$$(Lf)(x) = e^{2\pi\imath\Phi(x,c(B))} \sum_t \left(L_t^A(x) e^{-2\pi\imath\Phi(x_t^A,c(B))} \delta_t^{AB} \right).$$



Multiscale Butterfly Algorithm

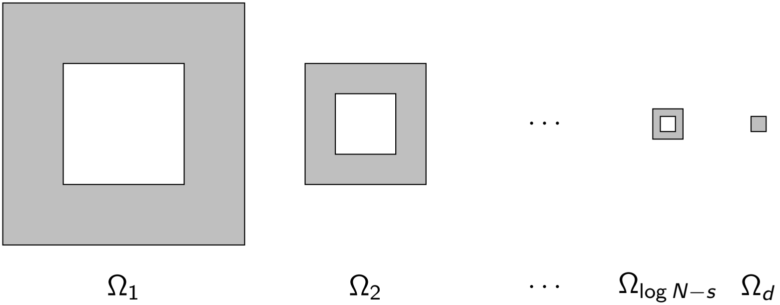


Figure : This figure shows the frequency domain decomposition of Ω. Each sub-domain Ω_j, $j = 1, \ldots, \log N - s$, is a rectangular ring and Ω_d is a small rectangular domain near the origin.

Multiscale Butterfly Algorithm

input : $\Phi, N, \widehat{f}$
output : Lf
for $j = 1, 2, \ldots, \log N - s$ **do**
  u^{Ω_j} =Cartesian-Butterfly$(\Phi, N/2^j, \widehat{f}_{\Omega_j})$
end
for $x \in X$ **do**
  for $\xi \in [-2^{s-1}, 2^{s-1})^2$ **do**
    $u^{\Omega_d} = e^{2\pi\imath\Phi(x,\xi)}\widehat{f}(\xi)$
  end
end
$Lf = u^{\Omega_d}$
for $j = 1, 2, \ldots, \log N - s$ **do**
  $Lf = Lf + u^{\Omega_j}$
end

Complexity Analysis

The cost for computing a Cartesian butterfly algorithm on a domain of size N^2 is $C = \mathcal{O}(N^2 \log N)$.

The total complexity of the multiscale butterfly algorithm is

$$\sum_{j=1}^{\log N - s} \frac{C}{2^{2(s-1)}} + 2^{2s} N^2 = C \sum_{j=1}^{\log N - s} \frac{1}{2^{2(s-1)}} + 2^{2s} N^2 = \mathcal{O}(N^2 \log N).$$

Desktop and Matlab

- Intel® Core™ i7-3770K CPU @3.50GHz
- 4 Cores and 8 Threads
- 8M Cache
- 32 GB of memory
- OS: Ubuntu 14.04 LTS

- MATLAB® R2013a

Numerical results in 2D

A generalized Radon transform whose kernel is given by

$$\Phi(x,\xi) = x\cdot\xi + \sqrt{c_1^2(x)\xi_1^2 + c_2^2(x)\xi_2^2},$$
$$c_1(x) = (2+\sin(2\pi x_1)\sin(2\pi x_2))/3,$$
$$c_2(x) = (2+\cos(2\pi x_1)\cos(2\pi x_2))/3.$$

N, q	e^h	$T_d(sec)$	$T_h(sec)$	T_d/T_h
256,7	6.44e-03	4.15e+02	7.55e+01	5.50e+00
512,7	6.58e-03	7.43e+03	3.85e+02	1.93e+01
1024,7	8.76e-03	1.55e+05	1.88e+03	8.22e+01
2048,7	8.51e-03	2.65e+06	8.87e+03	2.99e+02
256,11	2.39e-05	4.81e+02	1.74e+02	2.77e+00
512,11	1.26e-05	7.76e+03	8.17e+02	9.50e+00
1024,11	1.61e-05	1.49e+05	3.70e+03	4.04e+01
2048,11	1.82e-05	2.53e+06	1.64e+04	1.55e+02

Numerical results in 3D

Consider a simple example modeling the integration over spheres with varying radii, whose phase function is given by

$$\Phi(x,\xi) = x \cdot \xi + c(x)\sqrt{\xi_1^2 + \xi_2^2},$$
$$c(x) = (3 + \sin(2\pi x_1)\sin(2\pi x_2)\sin(2\pi x_3))/4.$$

N, q	e^h	$T_d(sec)$	$T_h(sec)$	T_d/T_h
64,5	9.41e-02	1.83e+04	2.50e+03	7.31e+00
128,5	7.57e-02	6.21e+05	2.42e+04	2.57e+01
256,5	8.23e-02	3.91e+07	2.35e+05	1.66e+02
64,7	1.20e-02	1.830e+04	7.32e+03	2.50e+00
128,7	1.04e-02	6.03e+05	4.48e+04	1.35e+01
256,7	8.13e-03	4.39e+07	3.81e+05	1.15e+02

Conclusion

- The multiscale butterfly algorithm decompose the frequency domain of an FIO to avoid possible singularity of the phase function $\Phi(x, \xi)$ at $\xi = 0$.
- The Cartesian butterfly algorithm is applied to evaluate the FIO over each sub-domain.
- The complexity is $\mathcal{O}(N^2 \log N)$ with a smaller pre-factor.

Q & A

Suggested Questions:

- How to prove the Cartesian low-rank approximation theorem?
- What is the memory complexity and practical usage?
- How to extend the algorithm to 3D problem?

The End

Thank You!

Selected References I

Candès, E., Demanet, L., and Ying, L. (2007).
Fast computation of Fourier integral operators.
SIAM Journal on Scientific Computing, 29(6):2464–2493.

Candès, E., Demanet, L., and Ying, L. (2009).
A fast butterfly algorithm for the computation of Fourier integral operators.
Multiscale Modeling and Simulation, 7(4):1727–1750.

Li, Y., Yang, H., Ho, K. L., and Ying, L. (2014).
Butterfly factorization.

Poulson, J., Demanet, L., Maxwell, N., and Ying, L. (2014).
A parallel butterfly algorithm.
SIAM Journal on Scientific Computing, 36(1):C49–C65.

Yang, H., Li, Y., and Ying, L. (2014).
Multiscale butterfly algorithm for fourier integral operators.