Fast Algorithms for the Computation of Oscillatory Integrals

Emmanuel Candès

California Institute of Technology

EPSRC Symposium Capstone Conference
Warwick Mathematics Institute, July 2009

Collaborators

- Lexing Ying
- Laurent Demanet

Our problem

Evaluate numerically a Fourier integral operator (FIO)

$$(Tf)(x) = \int_{\mathbb{R}^d} a(x,k) e^{2\pi i \Phi(x,k)} \hat{f}(k)\, \mathrm{d}k$$

at points x given on a Cartesian grid

- $k \in \mathbb{R}^d$: frequency variable ($\hat{f}(k) = \int_{\mathbb{R}^d} e^{-2\pi i x \cdot k} f(x)\, \mathrm{d}x$)
- $a(x,k)$: (smooth) amplitude
- $\Phi(x, \lambda k)$: homogeneous (smooth) phase function as large as $|k|$

$$\Phi(x, \lambda k) = \lambda\, \Phi(x,k), \qquad \lambda > 0$$

e.g. $\Phi(x,k) = g(x)|k|$

A motivating example: wave propagation

$$\frac{\partial^2 u}{\partial t^2}(x,t) = c^2 \Delta u(x,t), \qquad \begin{array}{rcl} u(x,0) & = & u_0(x) \\ \partial u/\partial t(x,0) & = & 0 \end{array}$$

A motivating example: wave propagation

$$\frac{\partial^2 u}{\partial t^2}(x,t) = c^2 \Delta u(x,t), \qquad \begin{array}{rcl} u(x,0) & = & u_0(x) \\ \partial u/\partial t(x,0) & = & 0 \end{array}$$

Solution operator is

$$u(x,t) = \frac{1}{2}\left(\int_{\mathbb{R}^2} e^{2\pi i(x\cdot k + c|k|t)}\hat{u}_0(k)\,\mathrm{d}k + \int_{\mathbb{R}^2} e^{2\pi i(x\cdot k - c|k|t)}\hat{u}_0(k)\,\mathrm{d}k\right)$$

Two FIOs with phase functions

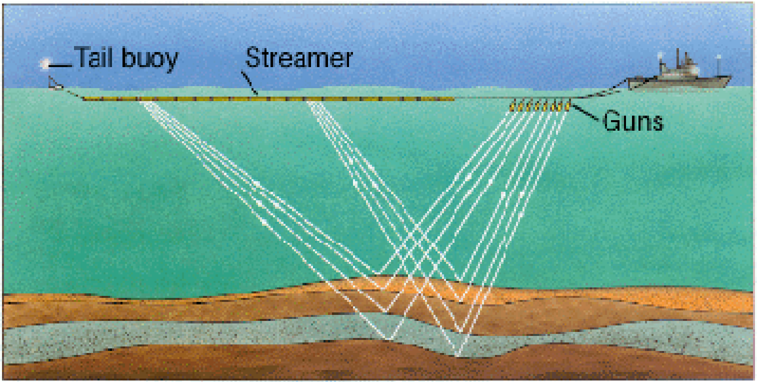
$$\Phi_\pm(x,k) = x\cdot k \pm c|k|t$$

Inhomogeneous medium $c(x) \to$ solution operator $=$ sum of two FIOs (small times)

Importance of FIOs

- Arise in many (inverse) problems
- Applying FIOs is often the computational bottleneck

Example in seismics

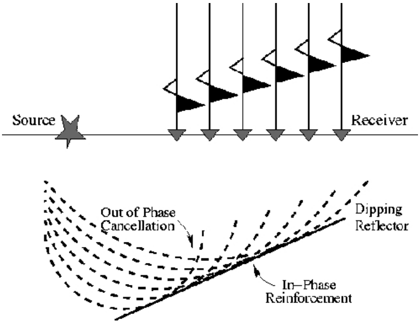


Marine survey

Kirchhoff migration

Wave measurements $f_s(t, x_r)$ parametrized by

- time t
- receiver location x_r
- source coordinate x_s



- Forward map: $F\delta c = \delta p$
- Imaging operator is F^*: FIO under general assumptions
- Approximations by generalized Radon transform (GRT): integration of f_s over fixed set of curves parametrized by travel times

$$g_s(x) = \int \delta(t - \tau(x, x_r) - \tau(x, x_s))\, f_s(t, x_r)\, \mathrm{d}t\, \mathrm{d}x_r$$

Followed by stack operation over the s index

Other examples

- Transmission electron microscopy
- Radar imaging
- Ultrasound imaging

Discrete computational problem

Discrete grids

$$X = \{(i_1/N, i_2/N) : 0 \leq i_1, i_2 < N\} \subset [0,1]^2$$
$$\Omega = \{(k_1/N, k_2/N) : -N/2 \leq k_1, k_2 < N\} \subset [-1/2, 1/2]^2$$

Given input $\{f(k)\}_{k\in\Omega}$, evaluate

$$(Tf)(x) := \frac{1}{N} \sum_{k\in\Omega} a(x,k)\, e^{2\pi i N \Phi(x,k)} f(k), \quad \text{at } \textit{all} \quad x \in X$$

with Φ smooth and homogeneous in k

Peek at the results

$$(Tf)(x) := \frac{1}{N} \sum_{k \in \Omega} a(x,k)\, e^{2\pi i N \Phi(x,k)} f(k), \quad x \in X$$

Kernel is not analytic and is highly oscillatory

- Naive evaluation $O(N^4)$ ($O(N^2)$ inputs/outputs)
- Algorithm for fast summation $O(N^{2.5} \log N)$ (C., Demanet and Ying, '06)

Peek at the results

$$(Tf)(x) := \frac{1}{N} \sum_{k \in \Omega} a(x,k)\, e^{2\pi i N \Phi(x,k)} f(k), \quad x \in X$$

Kernel is not analytic and is highly oscillatory

- Naive evaluation $O(N^4)$ ($O(N^2)$ inputs/outputs)
- Algorithm for fast summation $O(N^{2.5} \log N)$ (C., Demanet and Ying, '06)

Today:

Novel algorithm with optimal complexity for *accurate* summation

- $O(N^2 \log N)$ flops
- $O(N^2)$ storage

Agenda

- The butterfly structure
- Fast butterfly algorithm for the evaluation of FIOs

The Butterfly Structure

Butterfly algorithm

General algorithmic structure for evaluating certain types of integrals

$$u_i = \sum_j K(x_i, p_j) f_j$$

- Introduced by Michielssen and Boag ('96)
- Generalized by O'Neil and Rokhlin ('07)

Butterfly algorithm

General algorithmic structure for evaluating certain types of integrals

$$u_i = \sum_j K(x_i, p_j) f_j$$

- Introduced by Michielssen and Boag ('96)
- Generalized by O'Neil and Rokhlin ('07)

Example: $K(x,p) = e^{2\pi i N x p}$

- $\{x_i\}$: N points in $[0,1]$
- $\{p_j\}$: N points in $[0,1]$
- $\{f_j\}$: sources at $\{p_j\}$

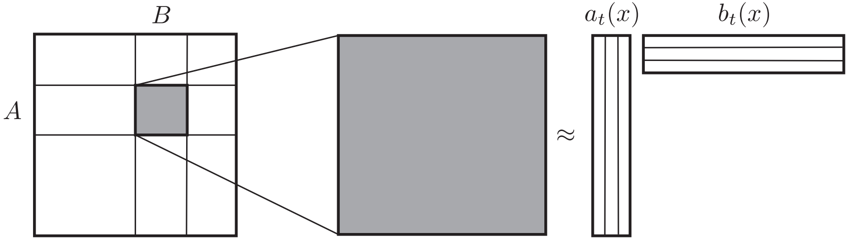
Applications

- FFT
- *nonuniform* FFTs
- many others

Kernel is dense and oscillatory

Low-rank approximation ($K(x,p) = \exp(2\pi i N xp)$)

A interval in x
B interval in p obeying length$(A) \times$ length$(B) \leq 1/N$



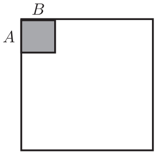
The submatrix $\{K(x_i, p_j) : x_i \in A, p_j \in B\}$ has approximately low rank:

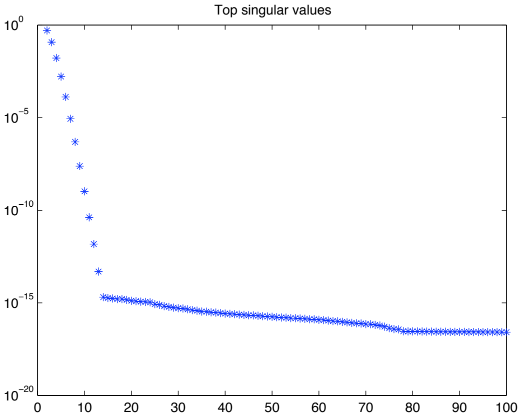
$$|K(x,p) - \sum_{t=1}^{r} a_t(x) b_t(p)| \leq \epsilon$$

with $r = O(\log(1/\epsilon))$

Example

- $10^6 \times 10^6$ DFT
- Top left $10^3 \times 10^3$ block



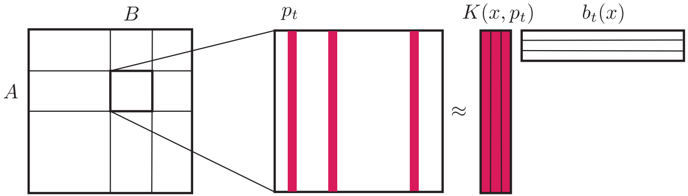


Interpolative decompositions

O'Neil and Rokhlin suggest using interpolative decompositions

$$a_t(x) := K(x, p_t), \quad p_t \in B$$

- Rank-revealing QR decomposition: Gu and Eisenstat ('96)
- Interpolative decomp.: Cheng, Gimbutas, Martinsson and Rokhlin ('05)



Interpolative representation

$$K(x, p) \approx \sum_{t=1}^{r} K(x, p_t) b_t(x)$$

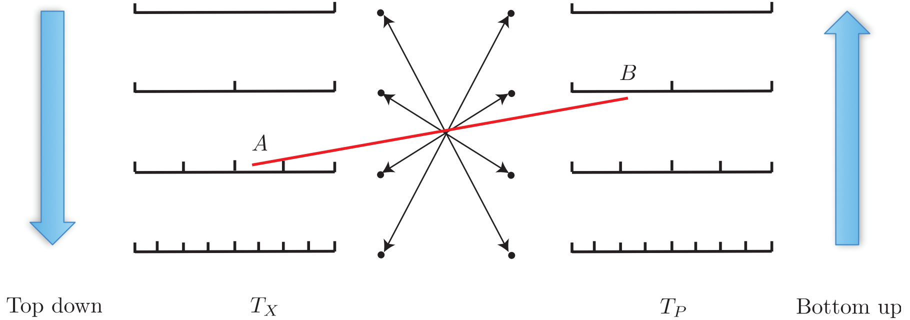
Cost for an $m \times n$ matrix is $O(mn^2)$

Multiscale decompositions

- Compute low-rank approximations of all submatrices obeying

$$\text{length}(A) \times \text{length}(B) = 1/N$$

- Use two-scale relations for efficiency



Definition of partial sums and equivalent sources

- Partial sums

$$u^B(x) = \sum_{p_j \in B} K(x, p_j) f_j$$

- Approximation for $x \in A$ and length$(A) \times$ length$(B) \leq 1/N$

$$u^B(x) \approx \sum_{t=1}^{r} K(x, p_t^{AB}) \left(\sum_{p_j \in B} b_t^{AB}(p_j) f_j \right), \quad x \in A$$

Definition of partial sums and equivalent sources

- Partial sums

$$u^B(x) = \sum_{p_j \in B} K(x, p_j) f_j$$

- Approximation for $x \in A$ and length$(A) \times$ length$(B) \leq 1/N$

$$u^B(x) \approx \sum_{t=1}^{r} K(x, p_t^{AB}) \left(\sum_{p_j \in B} b_t^{AB}(p_j) f_j \right), \quad x \in A$$

- *Equivalent sources* for (A, B): $\{f_t^{AB}\}_{1 \leq t \leq r}$

$$f_t^{AB} = \sum_{p_j \in B} b_t^{AB}(p_j) f_j$$

Definition of partial sums and equivalent sources

- Partial sums

$$u^B(x) = \sum_{p_j \in B} K(x, p_j) f_j$$

- Approximation for $x \in A$ and length$(A) \times$ length$(B) \leq 1/N$

$$u^B(x) \approx \sum_{t=1}^{r} K(x, p_t^{AB}) \left(\sum_{p_j \in B} b_t^{AB}(p_j) f_j \right), \quad x \in A$$

- *Equivalent sources* for (A, B): $\{f_t^{AB}\}_{1 \leq t \leq r}$

$$f_t^{AB} = \sum_{p_j \in B} b_t^{AB}(p_j) f_j$$

- Compact representation of $K^{AB} : B \to A$

$$u^B(x) = \sum_{t=1}^{r} K(x, p_t^{AB}) f_t^{AB}$$

Definition of partial sums and equivalent sources

- Partial sums

$$u^B(x) = \sum_{p_j \in B} K(x, p_j) f_j$$

- Approximation for $x \in A$ and length$(A) \times$ length$(B) \leq 1/N$

$$u^B(x) \approx \sum_{t=1}^{r} K(x, p_t^{AB}) \left(\sum_{p_j \in B} b_t^{AB}(p_j) f_j \right), \quad x \in A$$

- *Equivalent sources* for (A, B): $\{f_t^{AB}\}_{1 \leq t \leq r}$

$$f_t^{AB} = \sum_{p_j \in B} b_t^{AB}(p_j) f_j$$

- Compact representation of $K^{AB} : B \to A$

$$u^B(x) = \sum_{t=1}^{r} K(x, p_t^{AB}) f_t^{AB}$$

Butterfly structure

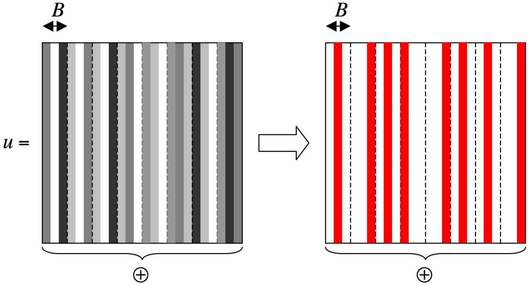
Recursive computation of $\{p_t^{AB}\}$, $\{f_t^{AB}\}$ for length$(A) \times$ length$(B) = 1/N$

Initialization

For all (A,B), $\ell(A)=1$ & $\ell(B)=1/N$, construct $\{p_t^{AB}\}_{1\leq t\leq r}$ and $\{f_t^{AB}\}_{1\leq t\leq r}$

$$f_t^{AB} = \sum_{p_j \in B} b_t^{AB}(p_j) f_j$$

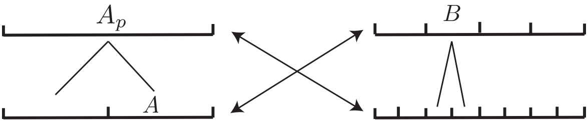
cost of constructing $\{p_t^{AB}\}$ is $O(r^2 N)$/pair
cost of constructing $\{f_t^{AB}\}$ is $O(r^2)$/pair
$\Rightarrow$ tot. cost is $O(r^2 N^2)$



Interpretation: $\{p_t^{AB}\}$ selected columns, $\{f_t^{AB}\}$ column weights

Recursion: "merge, split, compress"

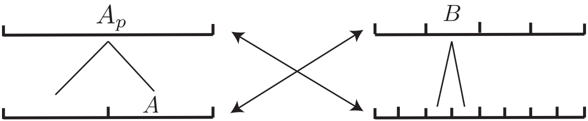
For all (A, B), $\ell(A) = 1/2$ & $\ell(B) = 2/N$, get $\{p_t^{AB}\}_{1 \le t \le r}$ and $\{f_t^{AB}\}_{1 \le t \le r}$



$$u^B(x) \approx \sum_{i=1}^{2} \sum_{t=1}^{r} K(x, p_t^{A_p B_i}) f_t^{A_p B_i}, \quad x \in A \subset A_p$$

Recursion: "merge, split, compress"

For all (A,B), $\ell(A)=1/2$ & $\ell(B)=2/N$, get $\{p_t^{AB}\}_{1\le t\le r}$ and $\{f_t^{AB}\}_{1\le t\le r}$



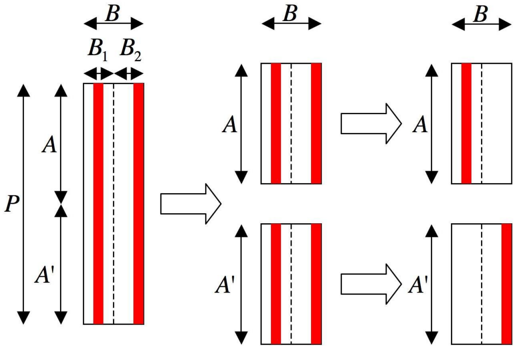
$$u^B(x) \approx \sum_{i=1}^{2}\sum_{t=1}^{r} K(x,p_t^{A_pB_i})f_t^{A_pB_i}, \quad x\in A\subset A_p$$

- Can reduce the rank of $K(x,p_t^{A_pB_i}) : x\in A, \{p_t^{A_pB_i}\}_{t,i} \to K(x,p_t^{AB})$
- Treat $\{f_t^{A_pB_i}\}_{t,i}$ as sources at $\{p_t^{A_pB_i}\}_{t,i}\subset B$

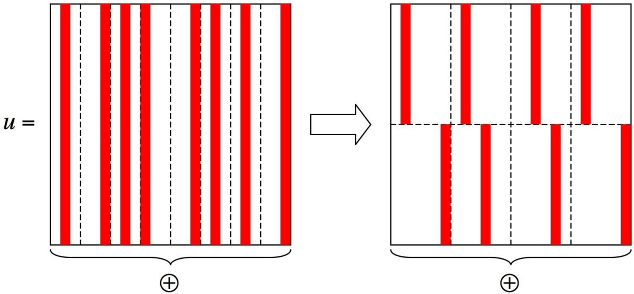
$$f_t^{AB} = \sum_{i=1}^{2}\sum_{s=1}^{r} b_t^{AB}(p_s^{A_pB_i})f_s^{A_pB_i}$$

cost of constructing $\{p_t^{AB}\}$ is $O(r^2N)$/pair
cost of constructing $\{f_t^{AB}\}$ is $O(r^2)$/pair $\Rightarrow$ tot. cost is $O(r^2N^2)$

Schematic representation



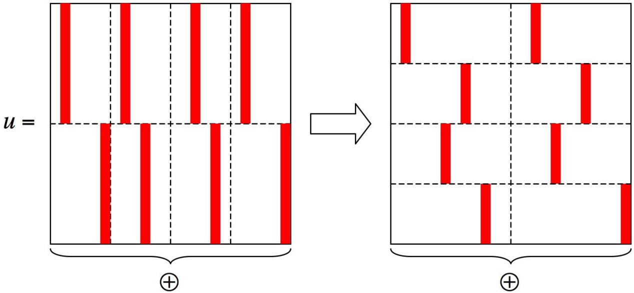
Schematic representation



$$u^{B_i}(x) = \sum_{t=1}^{r} K(x, p_t^{A_p B_i}) f_t^{A_p B_i}$$

$$u^{B}(x) = \sum_{t=1}^{r} K(x, p_t^{AB}) f_t^{AB}$$

Next step

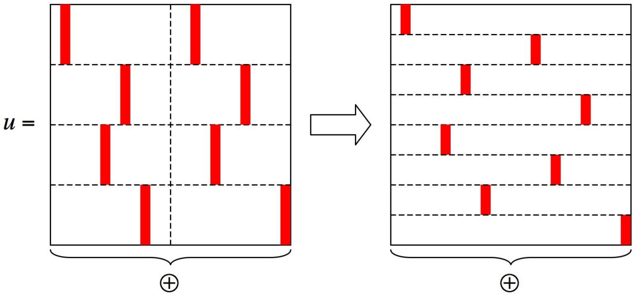


... until $\ell(B) = 1$ and $\ell(A) = 1/N$

Termination

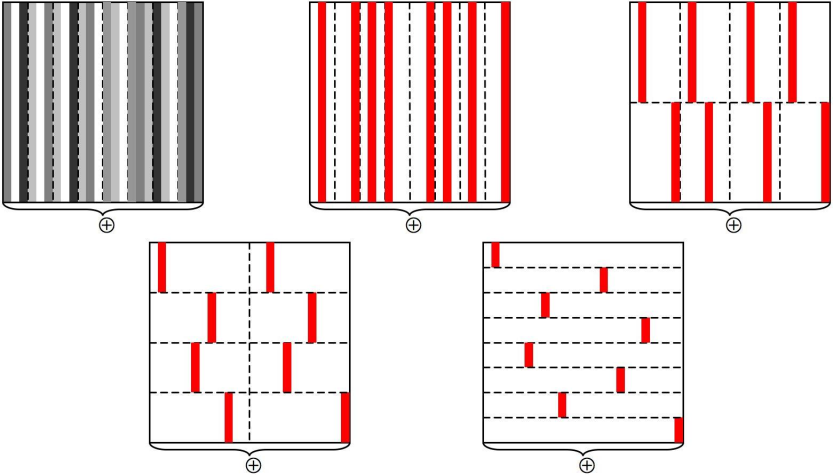
In the end, $\ell(A) = 1/N$ and $\ell(B) = 1$, and

$$u^B(x) \approx \sum_{t=1}^{r} K(x, p_t^{AB}) f_t^{AB}$$



cost of evaluating $u^B(x)$ is $O(r)$/pair $\Rightarrow$ tot. cost is $O(rN)$

Multiscale recursion



Summary: complexity analysis

If low-rank expansions available

- $O(r^2 N \log N)$ evaluation time
- $O(r^2 N \log N)$ storage

If not

- $O(r^2 N^2)$ evaluation time
- $O(r^2 N \log N)$ storage

Summary: complexity analysis

If low-rank expansions available

- $O(r^2 N \log N)$ evaluation time
- $O(r^2 N \log N)$ storage

If not

- $O(r^2 N^2)$ evaluation time
- $O(r^2 N \log N)$ storage

Powerful architecture but

- Precomputation time may be prohibitive
- Storage may be prohibitive

Summary: complexity analysis

If low-rank expansions available

- $O(r^2 N \log N)$ evaluation time
- $O(r^2 N \log N)$ storage

If not

- $O(r^2 N^2)$ evaluation time
- $O(r^2 N \log N)$ storage

Powerful architecture but

- Precomputation time may be prohibitive
- Storage may be prohibitive

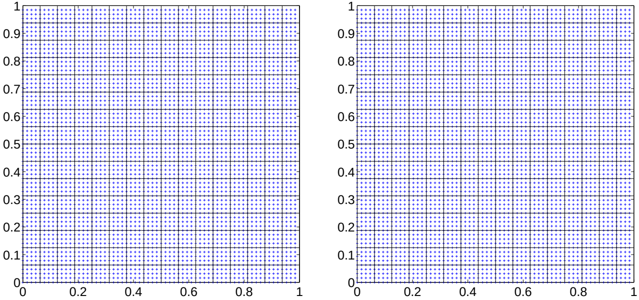
Our contribution

$O(r^2 N \log N)$ evaluation time and storage complexity is $O(r^2 N)$

Fast Evaluation of FIOs

Recall oscillatory integral

$$u(x) = \sum_{k \in \Omega} e^{2\pi i N \Phi(x,k)} f_k$$



$x \in X$ $k \in \Omega$

Polar coordinates for frequency variable k

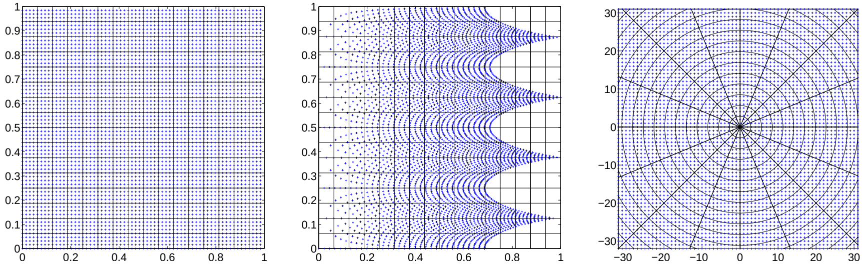
Phase may be singular at $k = 0$ because of homogeneity (e.g. $\Phi(x, k) = |k|$)

Polar coordinates for frequency variable k

Phase may be singular at $k = 0$ because of homogeneity (e.g. $\Phi(x, k) = |k|$)

Polar coordinates $p = (p_1, p_2)$

$$k_1 = p_1 \cos(2\pi p_2) \quad k_2 = p_2 \sin(2\pi p_2)$$

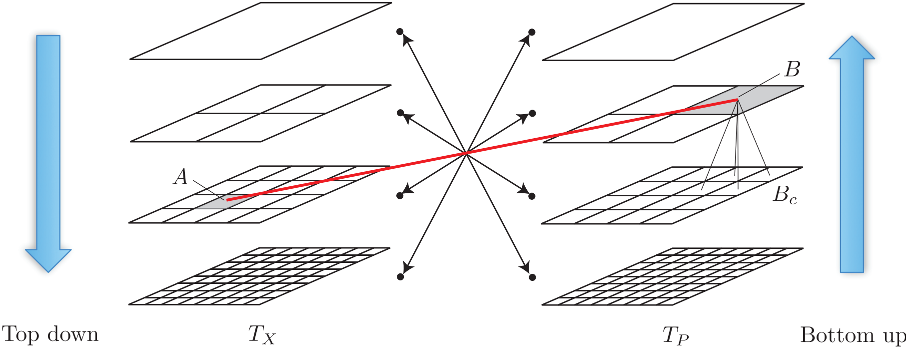


Cartesian — Polar — Partitioning in k

Slight abuse of notations

$$u(x) = \sum_{p \in \Omega} e^{2\pi i N \Psi(x,p)} f_p, \quad \Rightarrow \quad \text{smooth phase } \Psi$$

Hierarchical structure



$$\ell(A) \times \ell(B) = 1/N$$

Low-rank interactions

If $\ell(A)\ell(B) \leq 1/N$, kernel

$$e^{2\pi i N \Psi(x,p)} : x \in A, p \in B$$

has approx. low rank

Assume wlog A and B centered at 0

Low-rank interactions

If $\ell(A)\ell(B) \leq 1/N$, kernel

$$e^{2\pi i N \Psi(x,p)} : x \in A, p \in B$$

has approx. low rank

Assume wlog A and B centered at 0

- Residual phase in 1D (for simplicity)

$$\begin{aligned} R^{AB}(x,p) &= \Psi(x,p) - \Psi(0,p) - \Psi(x,0) + \Psi(0,0) \\ &= \partial_x \partial_p \Psi(x^*, p^*)\, xp \\ &= O(1/N) \end{aligned}$$

Same calculation in higher dimensions

Low-rank interactions

If $\ell(A)\ell(B) \leq 1/N$, kernel

$$e^{2\pi i N \Psi(x,p)} : x \in A, p \in B$$

has approx. low rank

Assume wlog A and B centered at 0

- Residual phase in 1D (for simplicity)

$$\begin{aligned} R^{AB}(x,p) &= \Psi(x,p) - \Psi(0,p) - \Psi(x,0) + \Psi(0,0) \\ &= \partial_x \partial_p \Psi(x^*, p^*)\, xp \\ &= O(1/N) \end{aligned}$$

  Same calculation in higher dimensions
- Shows that $e^{2\pi i N R^{AB}(x,p)}$ approx. low rank

Low-rank interactions

If $\ell(A)\ell(B) \leq 1/N$, kernel

$$e^{2\pi i N \Psi(x,p)} : x \in A, p \in B$$

has approx. low rank

Assume wlog A and B centered at 0

- Residual phase in 1D (for simplicity)

$$\begin{aligned} R^{AB}(x,p) &= \Psi(x,p) - \Psi(0,p) - \Psi(x,0) + \Psi(0,0) \\ &= \partial_x \partial_p \Psi(x^*, p^*)\, xp \\ &= O(1/N) \end{aligned}$$

  Same calculation in higher dimensions

- Shows that $e^{2\pi i N R^{AB}(x,p)}$ approx. low rank
- Factorization

$$e^{2\pi i N \Psi(x,p)} = e^{-2\pi i N \Psi(0,0)} \left[e^{2\pi i N \Psi(0,x)} e^{2\pi i N R^{AB}(x,p)} e^{2\pi i N \Psi(0,p)} \right]$$

Oscillatory Chebyshev interpolation

Chebyshev interpolation of $e^{2\pi i N R^{AB}(x,p)}$ in

- x when $\ell(A) \leq 1/\sqrt{N}$
- p when $\ell(B) \leq 1/\sqrt{N}$

E.g. interpolation in p

- $\{p_t\}^B$: tensor-Chebyshev grid in B
- $L_t^B(p)$: Lagrange interpolant with inputs at $\{p_t^B\}$, $L_t^B(p_{t'}^B) = \delta(t = t')$

Oscillatory Chebyshev interpolation

Chebyshev interpolation of $e^{2\pi i N R^{AB}(x,p)}$ in

- x when $\ell(A) \leq 1/\sqrt{N}$
- p when $\ell(B) \leq 1/\sqrt{N}$

E.g. interpolation in p

- $\{p_t\}^B$: tensor-Chebyshev grid in B
- $L_t^B(p)$: Lagrange interpolant with inputs at $\{p_t^B\}$, $L_t^B(p_{t'}^B) = \delta(t = t')$

With grid of logarithmic size in $1/\epsilon$

$$\left| e^{2\pi i N R^{AB}(x,p)} - \sum_t e^{2\pi i N R^{AB}(x,p_t^B)} L_t^B(p) \right| \leq \epsilon \quad \text{on } A \times B$$

When interpolation in x, low-rank approximation is $\sum_t e^{2\pi i N R^{AB}(x_t^A,p)} L_t^A(x)$

Demodulation/Interpolation/Remodulation

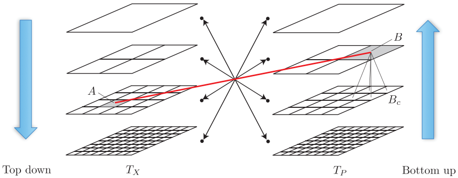
From $R^{AB}(x,p) = \Psi(x,p) - \Psi(0,p) - \Psi(x,0) + \Psi(0,0)$

$$\left| e^{2\pi i N \Psi(x,p)} - \sum_t \underbrace{e^{2\pi i N \Psi(x,p_t^B)}}_{a_t(x)} \underbrace{e^{-i2\pi N \Psi(0,p_t^B)} L_t^B(p)\, e^{i2\pi N \Psi(0,p)}}_{b_t(x)} \right| \leq \epsilon$$

Similar structure (with different interpretation) when interpolating kernel in x

Overall structure and two-scale relation

1. Initialize equivalent sources
2. Propagate equivalent sources (interpolation in p) until mid-level
3. Switch representation at mid-level
4. Propagate equivalent sources (interpolation in x)
5. Terminate by evaluating output



- Step 2: propagation of equivalent sources

$$
\begin{aligned}
f_t^{AB} &= \sum_{c,t'} b_t^B(p_{t'}^{B_c}) f_{t'}^{A_p B_c} \\
&= e^{-2\pi i N \Psi(x_0(A), p_t^B)} \sum_c \sum_{t'} L_t^B(p_{t'}^{B_c})\, e^{2\pi i N \Psi(x_0(A), p_{t'}^{B_c})}\, f_{t'}^{A_p B_c}
\end{aligned}
$$

- Step 4: similar

Finer points and summary

- $\{p_t^B\}$ and polynomials $L_t(p)$ are computed *all at once*
- Only need to store equivalent sources at pairs of consecutive scales
- Separation rank higher than that of the interpolative decomposition

Finer points and summary

- $\{p_t^B\}$ and polynomials $L_t(p)$ are computed *all at once*
- Only need to store equivalent sources at pairs of consecutive scales
- Separation rank higher than that of the interpolative decomposition

- Complexity is $O(\text{polylog}(1/\epsilon)\, N^2 \log N)$
- Storage is $O(\text{polylog}(1/\epsilon)\, N^2$

for ϵ-accurate computation

Easy extensions to varying amplitudes $\rightarrow a(x,k)e^{i2\pi N\Phi(x,k)}$

Numerical results

Generalized Radon transform integrating f along ellipses

- centered at x
- and with axes of length $c_1(x)$ and $c_2(x)$

is the sum of two FIOs with phases

$$\Phi_\pm(x,k) = x \cdot k \pm \sqrt{c_1^2(x)k_1^2 + c_2^2(x)k_2^2}$$

Numerical results

Generalized Radon transform integrating f along ellipses

- centered at x
- and with axes of length $c_1(x)$ and $c_2(x)$

is the sum of two FIOs with phases

$$\Phi_\pm(x,k) = x \cdot k \pm \sqrt{c_1^2(x)k_1^2 + c_2^2(x)k_2^2}$$

First example (constant amplitude)

$$u(x) = \sum_{k\in\Omega} e^{2\pi i N \Phi_+(x,k)} \hat{f}(k)$$

with

$$c_1(x) = \frac{1}{3}(2+\sin(2\pi x_1)\sin(2\pi x_2)), \quad c_2(x) = \frac{1}{3}(2+\cos(2\pi x_1)\cos(2\pi x_2))$$

(N, Chby. grid)	Alg. Time	Dir. Time	Speedup	Error
(1024, 5)	1.48e+3	9.44e+4	6.37e+1	1.26e-2
(2048, 5)	6.57e+3	1.53e+6	2.32e+2	1.75e-2
(4096, 5)	3.13e+4	2.43e+7	7.74e+2	1.75e-2
(1024, 7)	2.76e+3	9.48e+4	3.44e+1	6.45e-4
(2048, 7)	1.23e+4	1.46e+6	1.19e+2	8.39e-4
(4096, 7)	5.80e+4	2.31e+7	3.99e+2	8.18e-4
(1024, 9)	4.95e+3	9.44e+4	1.91e+1	3.45e-5
(2048, 9)	2.21e+4	1.48e+6	6.71e+1	4.01e-5
(4096, 9)	1.02e+5	2.23e+7	2.18e+2	4.21e-5
(1024, 11)	8.33e+3	9.50e+4	1.14e+1	5.23e-7
(2048, 11)	3.48e+4	1.49e+6	4.27e+1	5.26e-7

Conclusion: scales like $O(\log(1/\epsilon)) \times O(N^2 \log N)$

Numerical results II

Second example (variable amplitude): exact integration

$$u(x) = \sum_{k\in\Omega} a_+(x,k)e^{2\pi i N\Phi_+(x,k)}\hat{f}(k) + \sum_{k\in\Omega} a_-(x,k)e^{2\pi i N\Phi_-(x,k)}\hat{f}(k)$$

$$\begin{aligned} a_\pm(x,k) &= \left(J_0(2\pi c(x)|k|) \pm iY_0(2\pi c(x)|k|)\right) e^{\mp 2\pi i c(x)|k|} \\ \Phi_\pm(x,k) &= x\cdot k + c(x)|k| \end{aligned}$$

J_0 and Y_0 are Bessel functions and spheres' radii

$$c(x) = \frac{1}{4}(3+\sin(2\pi x_1)\sin(2\pi x_2))$$

(N, Chby. grid)	Alg. Time	Dir. Time	Speedup	Error
(256, 5)	1.39e+2	3.20e+3	2.31e+1	1.48e-2
(512, 5)	7.25e+2	5.20e+4	7.17e+1	1.62e-2
(1024, 5)	3.45e+3	8.34e+5	2.42e+2	1.90e-2
(256, 7)	2.69e+2	3.21e+3	1.19e+1	4.71e-4
(512, 7)	1.38e+3	5.20e+4	3.78e+1	7.30e-4
(1024, 7)	6.43e+3	8.35e+5	1.30e+2	6.35e-4
(256, 9)	5.23e+2	3.20e+3	6.12e+0	1.59e-5
(512, 9)	2.49e+3	5.17e+4	2.08e+1	2.97e-5
(1024, 9)	1.15e+4	8.32e+5	7.25e+1	1.75e-5
(256, 11)	1.04e+3	3.18e+3	3.06e+0	8.03e-7
(512, 11)	4.10e+3	5.11e+4	1.24e+1	9.38e-7
(1024, 11)	1.84e+4	8.38e+5	4.57e+1	8.01e-7

Conclusion: scales like $O(\log(1/\epsilon)) \times O(N^2 \log N)$

Summary

Accurate and near-optimal numerical evaluation of FIOs

- Operating characteristics
  - Butterfly structure
  - Residue phase
  - Oscillatory Chebyshev interpolation
- Many applications/extensions
  - Other types of oscillatory kernels $K(x, p)$
  - Other types of problems: e.g. sparse Fourier transform (Ying, '08)

Reference: E. J. Candès, L. Demanet and L. Ying (2008). "A Fast Butterfly Algorithm for the Computation of Fourier Integral Operators," to appear in *Mult. Model. Sim*