A Superfast Direct Solver for Nonuniform Discrete Fourier Transform of Type III

Jingyu Liu

School of Mathematical Sciences, Fudan University

July 10

Joint work with Prof. Yingzhou Li

NUDFT

Nonuniform discrete Fourier transform (NUDFT):

$$f_j = \sum_{k=0}^{N-1} \mathrm{e}^{2\pi \mathrm{i} x_j \omega_k} u_k, \quad 0 \leq j \leq M-1.$$

- $M \geq N$.
- Sample points: $\{x_j\}_{j=0}^{M-1} \subset [0,1)$.
- Frequencies: $\{\omega_k\}_{k=0}^{N-1} \subset [-1/2, N-1/2)$.
- Coefficients: $\{u_k\}_{k=0}^{N-1} \subset \mathbb{C}$.
- Target values: $\{f_j\}_{j=0}^{M-1} \subset \mathbb{C}$.

NUFFT

Fast forward computation:

- Uniform case (FFT): $M = N$, $x_j = j/M$ and $\omega_k = k$.
  Complexity: $\mathscr{O}(N \log N)$ [Cooley and Tukey 1965]
- Nonuniform case (NUFFT)
  - Type I: Equispaced sample points, i.e., $x_j = j/M$;
  - Type II: Integer frequencies, i.e., $\omega_k = k$;
  - Type III: Nonequispaced sample points and noninteger frequencies.

  Complexity: $\mathscr{O}(M + N \log N)$
  - Local expansions: [Anderson and Dahleh 1996], [Ruiz-Antolín and Townsend 2018]
  - Gridding algorithms: [Dutt and Rokhlin 1993], [Greengard and Lee 2004], [A. Barnett, Magland, and Klinteberg 2019].

INUDFT

Problem (Inverse NUDFT Problem, INUDFT)

Given sample points $\{x_j\}$ *and frequencies* $\{\omega_k\}$, *determine the coefficients* $\{u_k\}$ *from the target values* $\{f_j\}$.

- Can be modeled as a least-squares problem associated with the NUDFT matrix $\mathbf{A} = \left(\mathrm{e}^{2\pi \mathrm{i} x_j \omega_k}\right)_{j,k}$.
- The pseudoinverse of a NUDFT matrix does not simpliy equal to its adjoint.
- Accuracy is heavily affected by the distribution of sample points and frequencies.

Existing methods:

- Iterative methods: CG + NUFFT.
- Direct methods: [Kircheis and Potts 2019], [Wilber, Epperly, and A. H. Barnett 2025].

Displacement Structure of NUDFT-II Matrix

Method in [Wilber, Epperly, and A. H. Barnett 2025]:
NUDFT-II matrix

$$\mathbf{A}(j,k) = \mathrm{e}^{2\pi \mathrm{i} x_j k} = \gamma_j^k$$

is a Vandermonde matrix and satisfies the following Sylvester equation:

$$\mathbf{\Gamma A} - \mathbf{AC} = \mathbf{a e}_{N-1}^*,$$

where

- $\mathbf{\Gamma} = \mathrm{diag}(\gamma_0, \gamma_1, \ldots, \gamma_{M-1})$ is diagonal;
- $\mathbf{C} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \cdots \ \mathbf{e}_0]$ is a circulant matrix;
- $a_j = \gamma_j^N - 1$.

Displacement Structure of NUDFT-II Matrix

Fact: $\mathbf{C} = \mathbf{F}^{-1}\mathbf{D}\mathbf{F}$, where $\mathbf{F}$ is the standard $N \times N$ DFT matrix and $\mathbf{D} = \mathrm{diag}(\mathbf{F}\mathbf{e}_1)$.
The Sylvester equation thus becomes

$$\mathbf{\Gamma}\mathbf{A} - \mathbf{A}\mathbf{F}^{-1}\mathbf{D}\mathbf{F} = \mathbf{a}\mathbf{e}_{N-1}^*.$$

Let $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{F}^{-1}$ and $\mathbf{b} = \mathbf{F}^{-*}\mathbf{e}_{N-1}$, then

$$\mathbf{\Gamma}\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\mathbf{D} = \mathbf{a}\mathbf{b}^*,$$

where $b_k = \mathrm{e}^{-2\pi \mathrm{i} k/N}/N$.
[Beckermann and Townsend 2017]: Displacement structure $\Rightarrow$ Low-rankness.
Approximate $\tilde{\mathbf{A}}$ by a hierarchical semiseparable (HSS) matrix: Fast construction based on the factorized alternating direction implicit (fADI) method [Wilber 2021].

HSS Tree

Definition (HSS Tree)

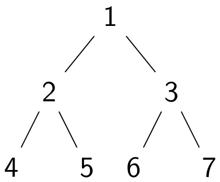
A tree T is called an HSS tree if

- T is a full binary tree with root node 1.
- There are two index sets I_τ and J_τ associated with each node τ of T, satisfying
  - $I_1 = [0, \ldots, M-1]$, $J_1 = [0, \ldots, N-1]$.
  - For a non-leaf node τ with children α_1 and α_2, $I_\tau = I_{\alpha_1} \sqcup I_{\alpha_2}$ and $J_\tau = J_{\alpha_1} \sqcup J_{\alpha_2}$.

Level 0

Level 1

Level 2



$J_1 = [0, \ldots, 255]$

$J_2 = [0, \ldots, 127]$, $J_3 = [128, \ldots, 255]$

$J_4 = [0, \ldots, 63]$, $J_5 = [64, \ldots, 127]$, ...

Definition (HSS Matrix)

A matrix $\mathbf{H}$ is said to be an HSS matrix with respect to an HSS tree T if there are matrices $\mathbf{D}_\tau$, $\mathbf{U}_\tau$, $\mathbf{V}_\tau$, $\mathbf{R}_\tau$, $\mathbf{W}_\tau$, $\mathbf{B}_{\tau,\sigma}$ (called HSS generators) associated with each node τ and its silbing σ, satisfying the following recursion:

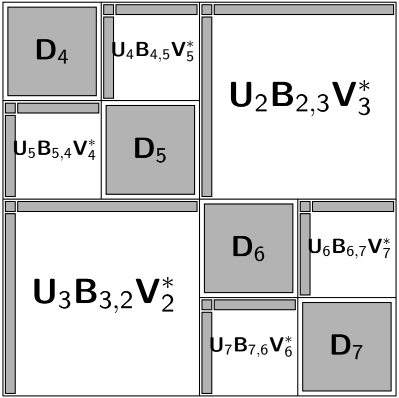
1. For a non-leaf node τ with children α_1 and α_2,

$$\mathbf{D}_\tau = \mathbf{H}(I_\tau, J_\tau) = \begin{bmatrix} \mathbf{D}_{\alpha_1} & \mathbf{U}_{\alpha_1}\mathbf{B}_{\alpha_1,\alpha_2}\mathbf{V}_{\alpha_2}^* \\ \mathbf{U}_{\alpha_2}\mathbf{B}_{\alpha_2,\alpha_1}\mathbf{V}_{\alpha_1}^* & \mathbf{D}_{\alpha_2} \end{bmatrix}.$$

2. For a non-leaf and non-root node τ with children α_1 and α_2,

$$\mathbf{U}_\tau = \begin{bmatrix} \mathbf{U}_{\alpha_1}\mathbf{R}_{\alpha_1} \\ \mathbf{U}_{\alpha_2}\mathbf{R}_{\alpha_2} \end{bmatrix}, \quad \mathbf{V}_\tau = \begin{bmatrix} \mathbf{V}_{\alpha_1}\mathbf{W}_{\alpha_1} \\ \mathbf{V}_{\alpha_2}\mathbf{W}_{\alpha_2} \end{bmatrix}.$$

An Example of the HSS Matrix



$$\mathbf{U}_2 = \begin{bmatrix} \mathbf{U}_4\mathbf{R}_4 \\ \mathbf{U}_5\mathbf{R}_5 \end{bmatrix}, \quad \mathbf{V}_2 = \begin{bmatrix} \mathbf{V}_4\mathbf{W}_4 \\ \mathbf{V}_5\mathbf{W}_5 \end{bmatrix},$$

$$\mathbf{U}_3 = \begin{bmatrix} \mathbf{U}_6\mathbf{R}_7 \\ \mathbf{U}_6\mathbf{R}_7 \end{bmatrix}, \quad \mathbf{V}_3 = \begin{bmatrix} \mathbf{V}_6\mathbf{W}_6 \\ \mathbf{V}_7\mathbf{W}_7 \end{bmatrix}.$$

Shared basis (nested basis) property: $\mathbf{U}_\tau$ is the column basis of $\mathbf{A}(I_\tau, J_\tau^{\mathrm{c}})$.

A Least-Squares Solver for the HSS Matrix

URV factorization [Xi et al. 2014]:

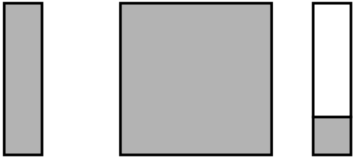
$$\mathbf{H} = \mathbf{Z}^{(L)} \cdots \mathbf{Z}^{(1)} \mathbf{Z}^{(0)} \mathbf{T}^{(0)} \mathbf{T}^{(1)} \mathbf{W}^{(1)*} \cdots \mathbf{T}^{(L)} \mathbf{W}^{(L)*},$$

where

- $\{\mathbf{Z}^{(\ell)}\}$, $\{\mathbf{W}^{(\ell)}\}$ are block unitary matrices;
- $\{\mathbf{T}^{(\ell)}\}$ are block upper-triangular matrices;
- Each block contains at most $\mathscr{O}(1)$ nonzeros.

Idea: Introduce zeros by a reverse QR factorization.

$$\mathbf{V}_4 = \mathbf{Q}_4 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{V}}_4 \end{bmatrix}$$



URV Factorization: Process

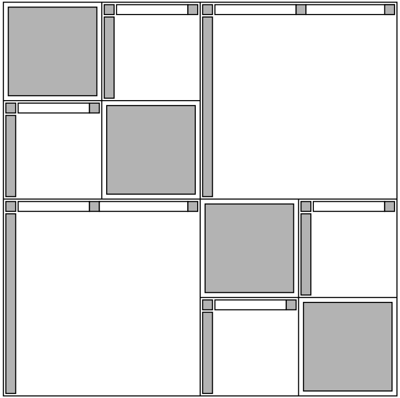


Figure: Introducing zeros into off-diagonal columns.

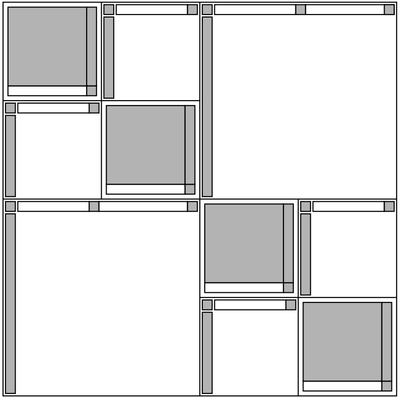


Figure: QR factorization on diagonal blocks.

Algorithm 1: A direct solver for INUDFT-II

Input: Samples $\{x_j\}_{j=0}^{M-1}$ and target values $\{f_j\}_{j=0}^{M-1}$.

Output: Coefficients $\{u_k\}_{k=0}^{N-1}$.

`/* Construction. */`

Approximate $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{F}^{-1}$ by an HSS matrix $\tilde{\mathbf{A}}_{\mathrm{HSS}}$;

`/* Factorization. */`

Compute the URV factorization of $\tilde{\mathbf{A}}_{\mathrm{HSS}}$;

`/* Solution. */`

Solve $\mathbf{v} = \arg\min_{\mathbf{w}} \|\tilde{\mathbf{A}}_{\mathrm{HSS}}\mathbf{w} - \mathbf{f}\|_2$;

Compute $\mathbf{u} = \mathbf{F}\mathbf{v}$ using FFT;

Fast structure in INUDFT-II:

$$\mathbf{A}_{\mathrm{fast}} = \tilde{\mathbf{A}}_{\mathrm{HSS}}\mathbf{F} \quad \text{and} \quad \mathbf{A}_{\mathrm{fast}}^{\dagger} = \mathbf{F}^{-1}\tilde{\mathbf{A}}_{\mathrm{HSS}}^{\dagger}.$$

Complexity

Stage	Operation	Complexity	Total
Construction	Construct $\tilde{\mathbf{A}}_{\mathrm{HSS}}$	k^2M	k^2M
Factorization	Factorize $\tilde{\mathbf{A}}_{\mathrm{HSS}}$	k^2M	k^2M
Solution	Apply $\tilde{\mathbf{A}}_{\mathrm{HSS}}^{\dagger}$ Apply $\mathbf{F}$	kM $N\log N$	$kM + N\log N$

Table: Complexities in the INUDFT–2 algorithm ($\mathscr{O}$ omitted).

Motivation

Fact: $\{\mathrm{e}^{2\pi \mathrm{i}\ell x}\}_{\ell\in\mathbb{Z}}$ forms an orthogonal basis of $L^2[0,1]$.
From direct calculation,

$$\mathrm{e}^{2\pi \mathrm{i}\omega_k x} = \sum_{\ell\in\mathbb{Z}} Q_{\ell,k}\mathrm{e}^{2\pi \mathrm{i}\ell x},$$

where

$$Q_{\ell,k} = \int_0^1 \mathrm{e}^{2\pi \mathrm{i}(\omega_k-\ell)x}\mathrm{d}x = \mathrm{e}^{\pi \mathrm{i}(\omega_k-\ell)}\frac{\sin\left(\pi(\omega_k-\ell)\right)}{\pi(\omega_k-\ell)} = G(\ell-\omega_k).$$

By truncating both sides to the R-th term,

$$\mathrm{e}^{2\pi \mathrm{i}\omega_k x} = \underbrace{\sum_{|\ell-\omega_k|\le R} Q_{\ell,k}\mathrm{e}^{2\pi \mathrm{i}\ell x}}_{\text{Approximation}} + \underbrace{\sum_{|\ell-\omega_k|>R} Q_{\ell,k}\mathrm{e}^{2\pi \mathrm{i}\ell x}}_{\text{Error}}.$$

The error term has the estimate: $\|\mathrm{err}_k\|_{L^2}^2 \le \mathscr{O}(1/R)$.

Error Bound When Sample Points Are Uniform Random Variables

For the NUDFT-III matrix:

$$\mathbf{A}(j,k) = \mathrm{e}^{2\pi \mathrm{i} x_j \omega_k} \approx \sum_{-R \le \ell \le N-1+R} \mathrm{e}^{2\pi \mathrm{i} x_j \ell} Q_{\ell,k}.$$

Or, in matrix form,

$$\mathbf{A} = \mathbf{B}\mathbf{Q} + \mathbf{E}.$$

Proposition

Suppose $\{x_j\}$ are i.i.d. uniform random variables in $[0,1)$ and $R \ge 2$, then $\mathbf{E}$ is a random matrix satisfying $\mathbb{E}\mathbf{E} = \mathbf{0}$ and

$$\mathbb{E}\|\mathbf{E}\|_{\mathrm{F}}^2 \le \|\mathbf{A}\|_{\mathrm{F}}^2 \frac{2}{\pi^2} \frac{1}{R - 3/2}.$$

Practical Algorithm

Denoting $\mathbf{H} = \mathbf{B}^{\dagger}\mathbf{A}$, we have

$$\begin{aligned} \mathbf{H} &= \mathbf{Q} + \mathbf{B}^{\dagger}\mathbf{E}, \\ \mathbf{A} &= \mathbf{B}\mathbf{H} + \left(\mathbf{I} - \mathbf{B}\mathbf{B}^{\dagger}\right)\mathbf{E}. \end{aligned}$$

- $\mathbf{B}$ is a NUDFT-II matrix having a fast structure;
- $\mathbf{Q}$ is a kernel matrix, can be compressed into an HSS matrix;
- $\mathbf{H}$ can be empirically compressed into an HSS matrix;
- $\mathbf{I} - \mathbf{B}\mathbf{B}^{\dagger}$ is a projection matrix.

The Construction of the Direct Solver

Construction on $\mathbf{B}_{\text{fast}}$: Type II direct solver.

Goal: Find an efficient way to construct $\mathbf{H} = \mathbf{B}_{\text{fast}}^{\dagger}\mathbf{A}$.

- Fast forward computation on $\mathbf{A}$: NUFFT.
- Fast forward computation on $\mathbf{B}_{\text{fast}}^{\dagger}$: URV factorization and FFT.

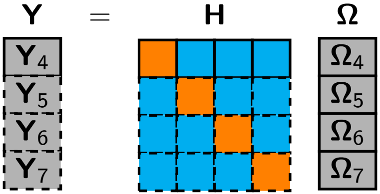
Black-box setting: Compress a matrix into its HSS form through matrix-vector multiplication.

Black-Box Construction of the HSS Matrix

Method in [Levitt and Martinsson 2024]: Recover the HSS structure using random samplings.

- Target rank: r.
- Sample number: $s \geq \max\{m + r, 3r\}$ where m is maximum size of the leaf node.

Their algorithm constructs an HSS matrix using $\{\mathbf{\Omega}, \mathbf{H}\mathbf{\Omega}, \mathbf{\Psi}, \mathbf{H}^*\mathbf{\Psi}\}$ where $\mathbf{\Omega}$ and $\mathbf{\Psi}$ are Gaussian random matrices of size $N \times s$.

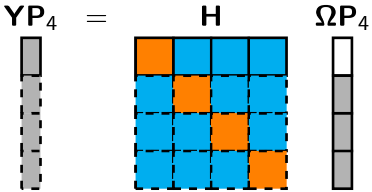


Main step: Find a basis $\mathbf{U}_4$ of $\mathbf{H}(I_4, J_4^c)$.

Black-Box Construction of the HSS Matrix

Idea: Eliminate the contribution of diagonal blocks.

Suppose $\mathbf{P}_4 = \mathrm{null}(\mathbf{\Omega}_4, r)$, then



Therefore, we have

$$\underbrace{\mathbf{Y}_4\mathbf{P}_4}_{\text{sample matrix}} = \sum_{\sigma=5}^{7} \mathbf{H}_{4,\sigma} \underbrace{\mathbf{\Omega}_\sigma \mathbf{P}_4}_{\text{test matrix}} .$$

Then construct $\mathbf{U}_4$ using $\mathbf{Y}_4\mathbf{P}_4$.

Complexity: $\mathscr{O}(k^2 N + k T_{\mathrm{mult}})$.

A Superfast Direct Solver of INUDFT-III

Algorithm 2: A direct solver for INUDFT-III.

Input: Sample points $\{x_j\}_{j=0}^{M-1}$, frequencies $\{\omega_k\}_{k=0}^{N-1}$, target values $\{f_j\}_{j=0}^{M-1}$.

Output: Coefficients $\{u_k\}_{k=0}^{N-1}$.

`/* Construction. */`

Construct a superfast direct solver $\mathbf{B}_{\rm fast}$ for the NUDFT-II matrix $\mathbf{B} \in \mathbb{C}^{M\times N}$;

Compress $\mathbf{B}_{\rm fast}^{\dagger}\mathbf{A}$ into an HSS matrix $\mathbf{H}_{\rm HSS}$ using the black-box method;

`/* Factorization. */`

Compute the URV factorization of $\mathbf{H}_{\rm HSS}$;

`/* Solution. */`

Calculate $\mathbf{u} = \mathbf{H}_{\rm HSS}^{-1}\mathbf{B}_{\rm fast}^{\dagger}\mathbf{f}$;

Fast structure in INUDFT-III:

$$\mathbf{A}_{\rm fast} = \mathbf{B}_{\rm fast}\mathbf{H}_{\rm HSS} \quad \text{and} \quad \mathbf{A}_{\rm fast}^{\dagger} = \mathbf{H}_{\rm HSS}^{-1}\mathbf{B}_{\rm fast}^{\dagger}.$$

Complexity

Stage	Operation	Complexity	Total
Construction	Construct $\mathbf{B}_{\text{fast}}$	$k^2 M$	$k^2 M + kN \log N$
	Apply $\mathbf{A}$	$M + N \log N$	
	Apply $\mathbf{B}_{\text{fast}}^\dagger$	kM	
	Compress $\mathbf{H}_{\text{HSS}}$	$k^2 N$	
Factorization	Factorize $\mathbf{H}_{\text{HSS}}$	$k^2 N$	$k^2 N$
Solution	Apply $\mathbf{B}_{\text{fast}}^\dagger$	$kM + N \log N$	$kM + N \log N$
	Apply $\mathbf{H}_{\text{HSS}}^\dagger$	kN	

Table: Complexities in the INUDFT-III algorithm ($\mathscr{O}$ omitted).

Experiment Setting

- $M = 4N$.
- Samples are i.i.d. uniform random variables in $[0, 1)$.
- Frequencies are perturbed integers given by

$$\omega_k = k + \alpha\psi_k, \quad 0 \leq k \leq N - 1,$$

where $\{\psi_k\}$ are IID uniform random variables on $[-1, 1]$ and $0 \leq \alpha < 1/2$ controls the non-uniformity.

The Low-Rank Property of $\mathbf{H}$

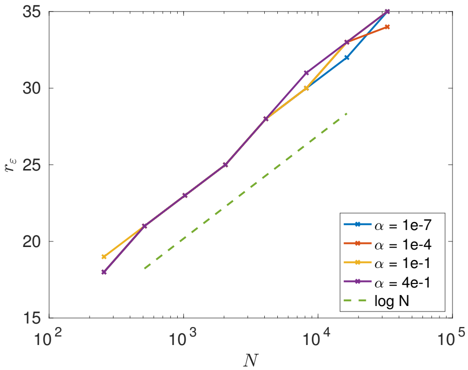
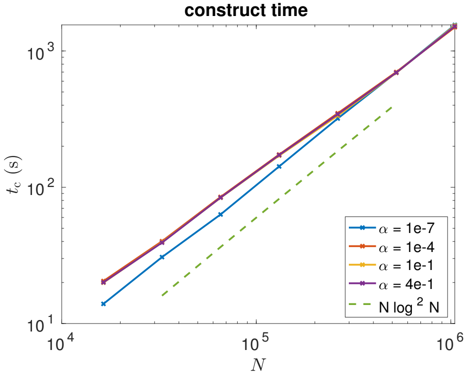
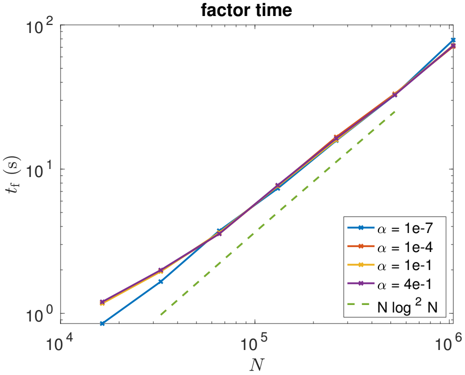


Figure: ε-rank of the top-right submatrix in $\mathbf{B}_{\text{fast}}^{\dagger}\mathbf{A}$ for $\varepsilon = 10^{-7}$. The size of the submatrix is $N/2$ where $N = 2^n$ for $n = 8:15$.

Direct Solver

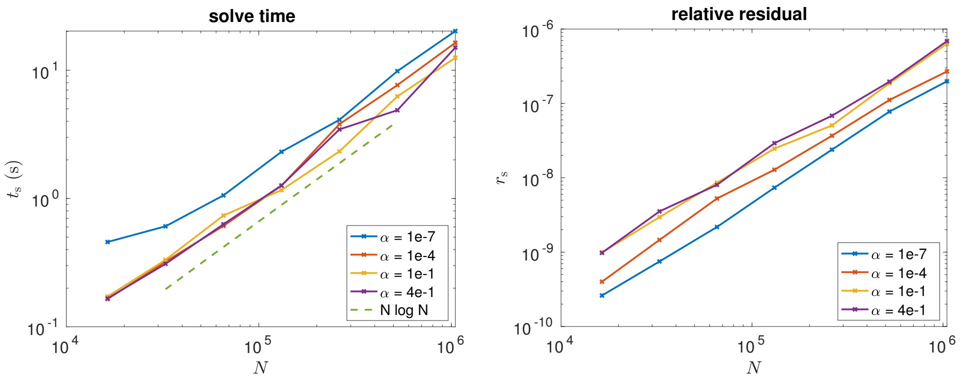


(a) Construction time.



(b) Factorization time.

Direct Solver



(a) Solution time. (b) Relative residual.

Preconditioner

N	method	α = 1e-1			
		$t_{\rm pre}$	$t_{\rm iter}$	$n_{\rm iter}$	$r_{\rm s}$
131072	CG	-	$\times$	$\times$	$\times$
	PCG	1.7e+02	1.4e+01	5	4.6e-13
262144	CG	-	$\times$	$\times$	$\times$
	PCG	3.4e+02	3.0e+01	5	7.4e-14
524288	CG	-	$\times$	$\times$	$\times$
	PCG	7.0e+02	5.6e+01	5	1.5e-13
1048576	CG	-	$\times$	$\times$	$\times$
	PCG	1.5e+03	1.4e+02	5	5.6e-14

Table: Time cost, iteration number and relative residual of CG and PCG when $\alpha = 0.1$. The marker "x" means iterative method does not converge in 500 steps.

Preconditioner

N	method	$\alpha = 4\text{e-}1$			
		t_{pre}	t_{iter}	n_{iter}	r_s
131072	CG	-	x	x	x
	PCG	1.7e+02	2.4e+01	9	1.0e-14
262144	CG	-	x	x	x
	PCG	3.4e+02	4.6e+01	8	7.1e-15
524288	CG	-	x	x	x
	PCG	7.0e+02	1.0e+02	9	4.1e-14
1048576	CG	-	x	x	x
	PCG	1.5e+03	2.1e+02	8	7.7e-13

Table: Time cost, iteration number and relative residual of CG and PCG when $\alpha = 0.4$. The marker "x" means iterative method does not converge in 500 steps.

Conclusions and Future Work

Conclusions:

- A superfast direct solver for INUDFT-III;
- The direct solver could also be used as an efficient preconditioner;
- Error bound of the forward approximation.

Future work:

- Extension to 2D and 3D problems;
- A method on regularized least-squares problem.

Thanks for listening!