A Sparse Data Fast Fourier Transform – Algorithm and Implementation

Alaeddin A. Aydıner, Weng Cho Chew, Jiming Song, and Tie Jun Cui

Center for Computational Electromagnetics (CCEM)
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign (UIUC)

Outline

- Introduction
- SDFFT in 1D
- The Algorithm in Higher Dimensions
- Implementation
- Parabolic Reflector Problem
- Conclusions

Introduction

Various NUFFT algorithms have been proposed by A. Dutt, V. Rokhlin, G. Beylkin, Q.-H. Liu et al. These methods all employ the regular FFT at some point, which imposes a regularity upon the spectral domain, i.e., it is not possible to *choose* an arbitrary set of spectral points along with an arbitrary set of spatial points. SDFFT allows for both, provided the Nyquist criterion is satisfied.

A. Dutt and V. Rokhlin, "Fast Fourier transforms for nonequispaced data," *SIAM Journal on Scientific Computing*, vol. 14, pp. 1368–1393, Nov. 1993.

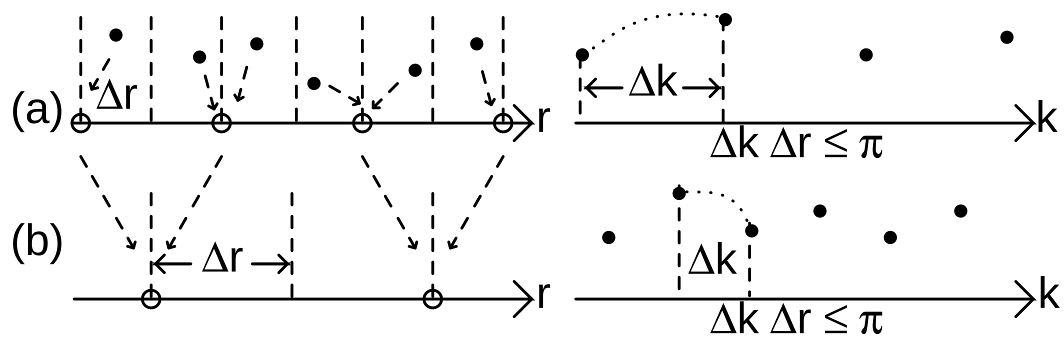
A. Dutt and V. Rokhlin, "Fast Fourier transforms for nonequispaced data, II," *Applied and Computational Harmonic Analysis*, vol. 2, pp. 85–100, Jan. 1995.

G. Beylkin, "On the fast Fourier transform of functions with singularities," *Appl. Computat. Harmonic Anal.*, vol. 2, pp. 363–382, 1995.

Q. H. Liu and N. Nguyen, "An accurate algorithm for nonuniform fast Fourier transforms (NUFFT's)," *IEEE Microwave and Guided Wave Letters*, vol. 8, pp. 18–20, July 2000.

Q. H. Liu, X. M. Xu, B. Tian, and Z. Q. Zhang, "Applications of nonuniform fast transform algorithms in numerical solutions of differential and integral equations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, pp. 1551–1560, July 2000.

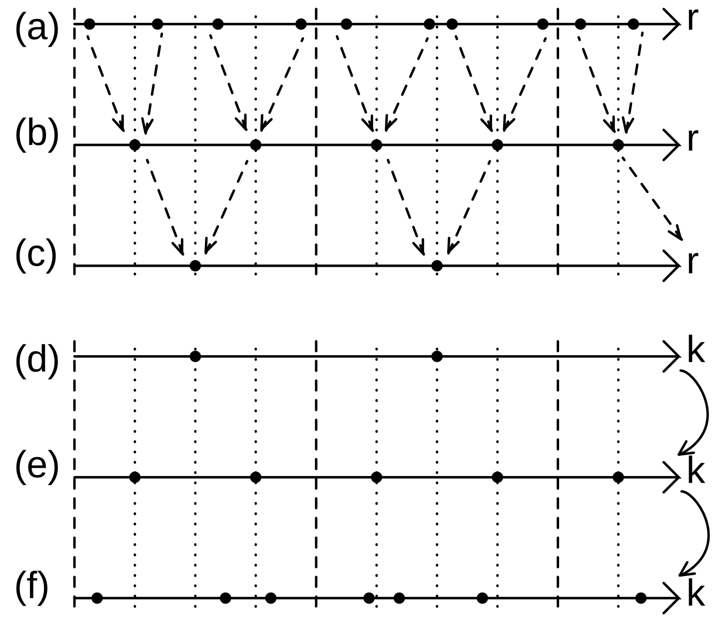
SDFFT in 1D



- Start with the dense spatial grid in (a) and compute the Fourier data on the corresponding sparse spectral grid by aggregating all input data with the relevant phases.

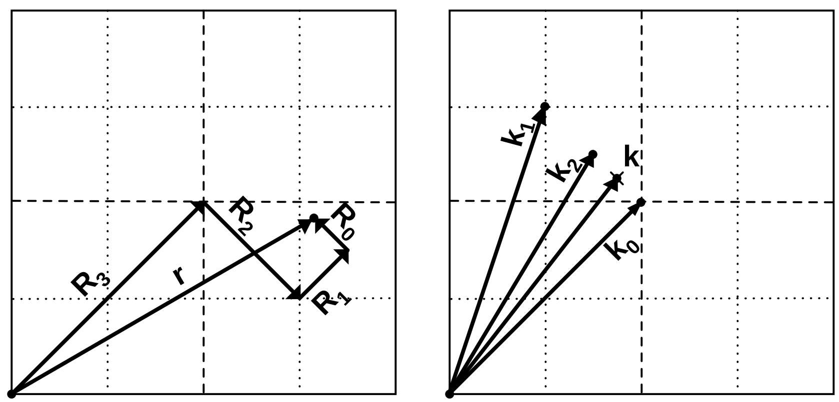
- Interpolate the Fourier data in the sparse spectral grid in (a) to the denser spectral grid in (b).

- Update the Fourier data at each spectral point in (b) by spatially translating the data, i.e., by applying the phase shifts corresponding to the translation from the denser spatial grid in (a) to the sparser one in (b).



The steps are performed recursively until the desired spectral data are obtained; i.e., starting with (a), the procedure arrives at (f) after a series of aggregations and interpolations. Also, the process starts with N_r r-space data mapped to $O(1)$ k-space points and ends with $O(1)$ r-space data mapped to N_k k-space points. Hence, the computational complexity is $O((N_r + N_k) \log N_r)$.

The Algorithm in Higher Dimensions



(a) (b)

Given multidimensional vectors $\mathbf{r}$ and $\mathbf{k}$ to denote spatial and spectral points, we have

$$\mathbf{r} = \sum_l \mathbf{R}_l.$$

Hence, we can compute the Fourier transform in a multilevel fashion:

$$\begin{aligned} \tilde{f}(\mathbf{k}) &= \sum_{\mathbf{r}} e^{-i\mathbf{k}\cdot\mathbf{r}} f(\mathbf{r}), \ \forall \mathbf{k} \in \mathcal{K} \\ &= \sum_{\mathbf{r}} \prod_l e^{-i\mathbf{k}\cdot\mathbf{R}_l} f(\mathbf{r}), \ \forall \mathbf{k} \in \mathcal{K}. \end{aligned} \quad (1)$$

The algorithm starts by computing

$$\tilde{f}(\mathbf{k_0}) = \sum_{\mathbf{r}} e^{-i\mathbf{k}_0\cdot\mathbf{R}_0} f(\mathbf{r}), \ \forall \mathbf{k_0} \in \mathcal{K},$$

which is *interpolated* to the denser mesh represented by $\mathbf{k_1}$, yielding $\tilde{f}_I(\mathbf{k_1})$.

Then, we incorporate the contributions due to $\mathbf{R_1}$ of each $\mathbf{r}$ by computing

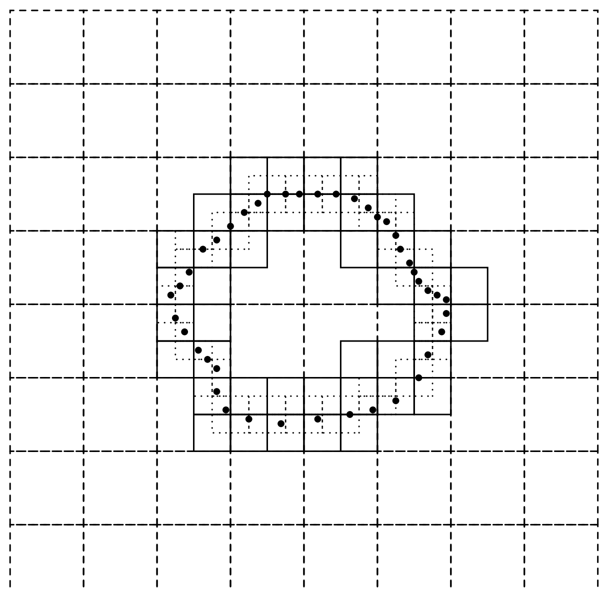
$$\tilde{f}(\mathbf{k_1}) = \sum_{\mathbf{r}} e^{-i\mathbf{k_1}\cdot\mathbf{R_1}} \tilde{f}_I(\mathbf{k_1}), \ \ \forall \mathbf{k_1} \in \mathcal{K}. \tag{2}$$

The interpolations and aggregations

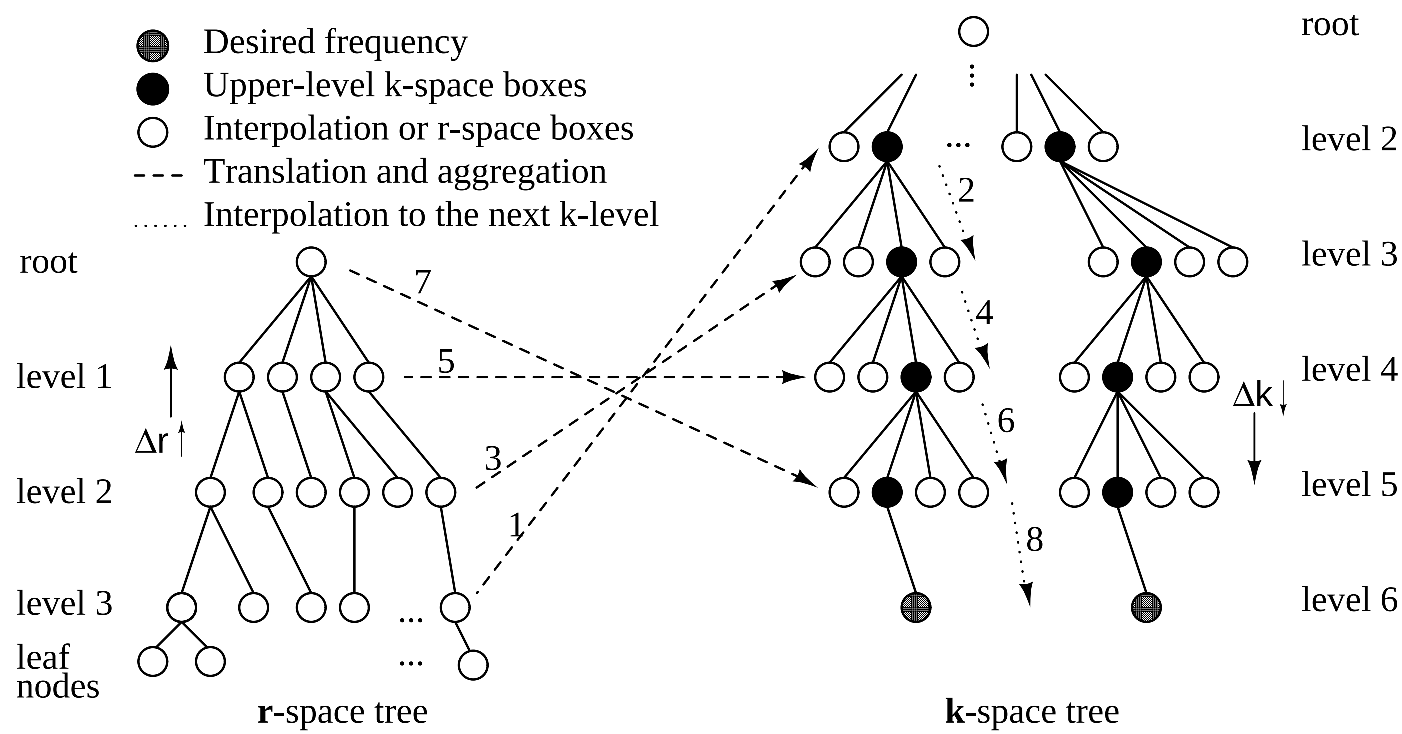
$$\tilde{f}(\mathbf{k_i}) \longrightarrow \tilde{f}_I(\mathbf{k_{i+1}}) \tag{3}$$

$$\tilde{f}(\mathbf{k_{i+1}}) = \sum_{\mathbf{r}} e^{-i\mathbf{k_{i+1}}\cdot\mathbf{R_{i+1}}} \tilde{f}_I(\mathbf{k_{i+1}}), \ \ \forall \mathbf{k_{i+1}} \in \mathcal{K} \tag{4}$$

follow each other until the particular set of k-vectors are computed. Thus, the algorithm builds cover-sets over the desired spectral points as shown below.



Implementation



Our implementation of the above algorithm involves two possibly unbalanced trees for the two domains as above. In the k-space tree, the SDFFT process starts at a level governed by the Nyquist criterion. While Δr becomes larger as we go up the r-space tree, Δk becomes accordingly smaller as we go down the k-space tree.

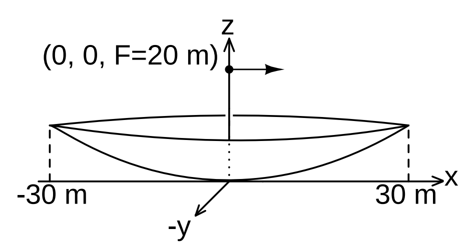
Parabolic Reflector Problem

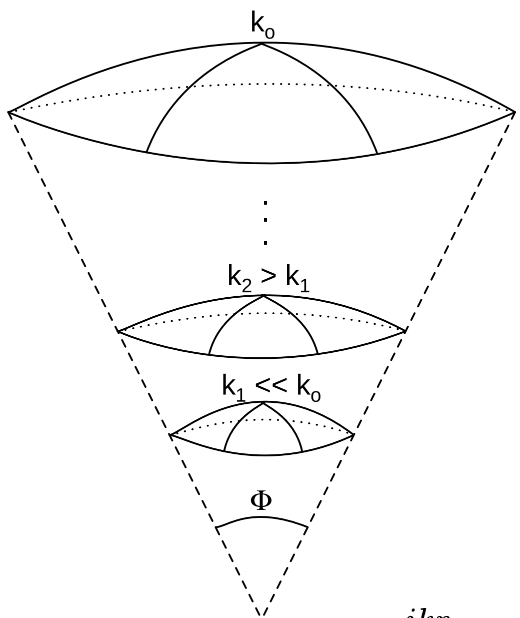
Although the physical optics (PO) approximation is useful to express antenna far-fields in terms of Fourier transforms, the resulting transforms have been difficult to evaluate efficiently due to the arbitrary domain over which they are defined. SDFFT offers an efficient way of computing such transforms.

With the PO approximation, $\mathbf{J}_s(\mathbf{r}) = 2\hat{\mathbf{n}} \times \mathbf{H}(\mathbf{r})$, the far-field pattern of a parabolic antenna is given by

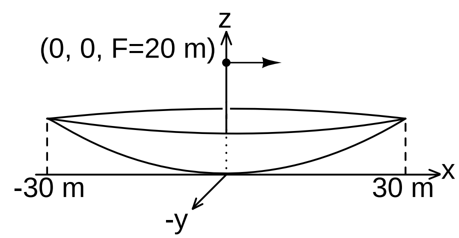
$$\mathbf{E} = i\omega\mu_o \frac{e^{ikr}}{4\pi r} \left(\widehat{\theta}\widehat{\theta} + \widehat{\phi}\widehat{\phi}\right) \cdot \int_{S'} d\mathbf{r}' e^{-i\mathbf{k}\cdot\mathbf{r}'} \left[2\widehat{\mathbf{n}}' \times \mathbf{H}(\mathbf{r}')\right], \tag{5}$$

in which $\mathbf{H}(\mathbf{r}) = \mathrm{I}l\boldsymbol{\nabla} \times \frac{e^{ik_o|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}\widehat{\boldsymbol{\alpha}}$, k_o is the dipole frequency, $\widehat{\boldsymbol{\alpha}}$ is the orientation of the source dipole located at the focus of the parabolic surface.





In order to compute broadband data as in the figure, we need to find the k_0 dependency of the transform. The field can be rewritten as:



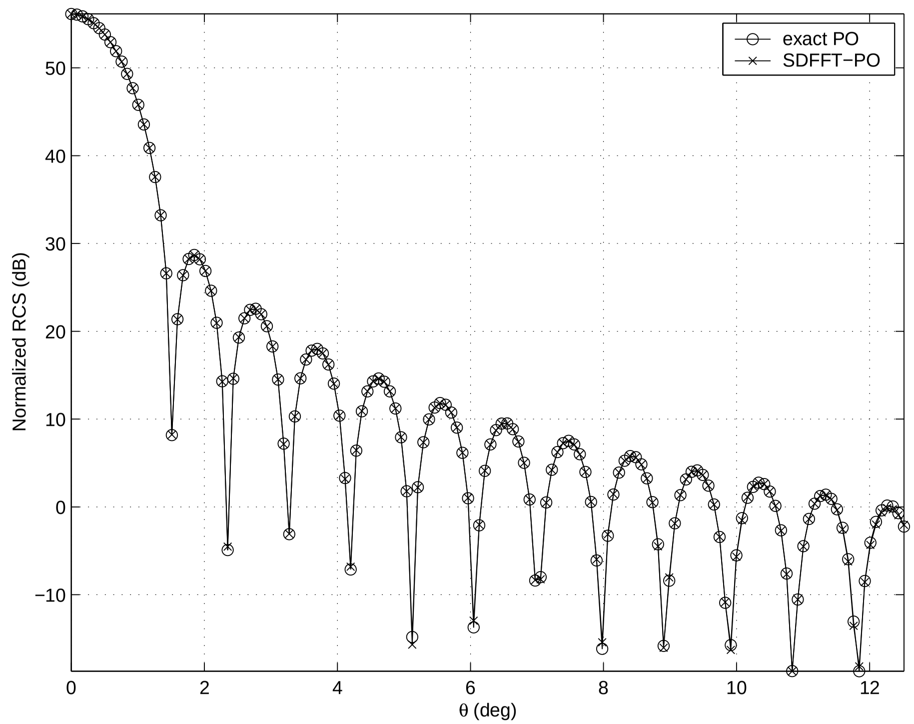
$$\begin{aligned}\mathbf{E} &= i\omega\mu_o \frac{e^{ikr}}{4\pi r}\left(\widehat{\theta}\widehat{\theta}+\widehat{\phi}\widehat{\phi}\right)\cdot\int_{S'} d\mathbf{r}' e^{-i\mathbf{k}\cdot\mathbf{r}'} \times \\ &\quad \left\{\frac{\mathrm{I}l}{2\pi}\frac{ik_oR-1}{R^2}\frac{e^{ik_oR}}{R}\left[\left(y'^2-2F(z'-F)\right)\widehat{x}-x'y'\widehat{y}-x'(z'-F)\widehat{z}\right]\right\} \\ &= i\omega\mu_o \frac{e^{ikr}}{4\pi r}\left(\widehat{\theta}\widehat{\theta}+\widehat{\phi}\widehat{\phi}\right)\cdot\int_{S'} d\mathbf{r}' \underbrace{e^{ik_oF}e^{-i(\mathbf{k}-k_o\widehat{z})\cdot\mathbf{r}'}}_{e^{-i\mathbf{k}\cdot\mathbf{r}'+ik_oR}}\left(ik_oR-1\right)\mathbf{f}(F,\mathbf{r}') \qquad (6)\end{aligned}$$

where $R = z' + F$ is the distance of a point on the parabolic reflector to the focus.

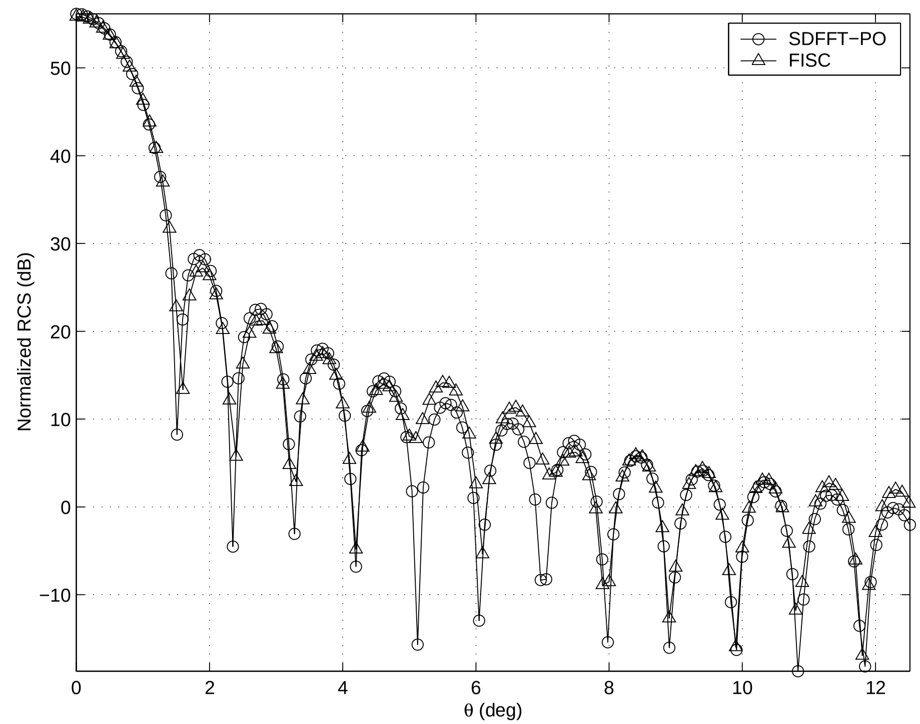
- A stationary phase analysis reveals that the main contribution to the integral is due to $\mathbf{k} \approx k_o \hat{z}$.

- To obtain broadband data, it suffices to perform the Fourier transforms in terms of a shifted z-component, $k'_z = k_z - k_o$.

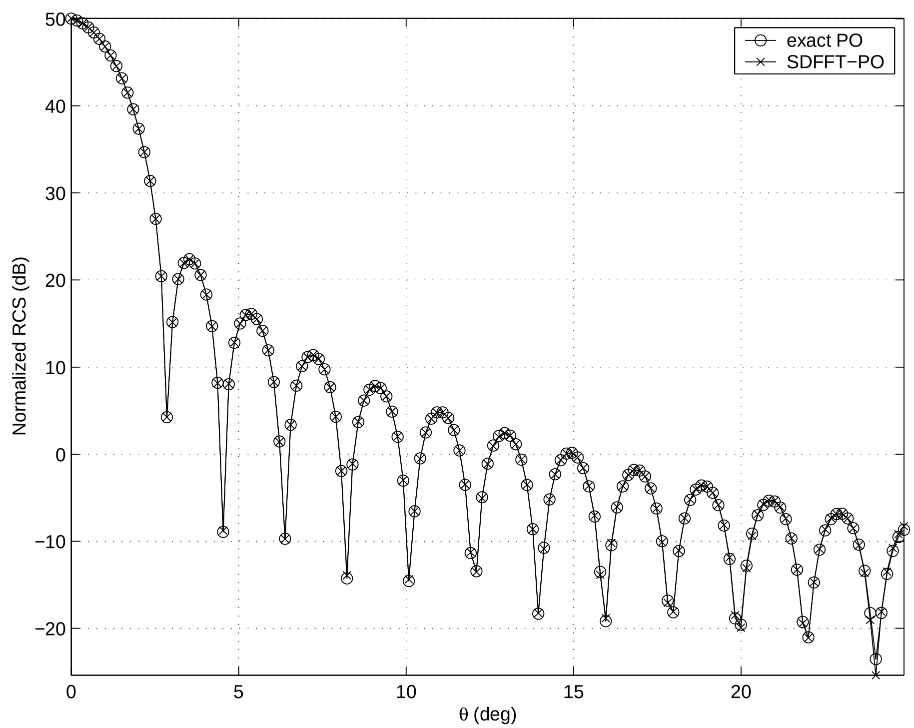
- Because of the $(ik_oR - 1)$ factor, two Fourier transforms need to be computed for each dimension. However, since $k_oR \gg 1$, the latter term may be neglected.



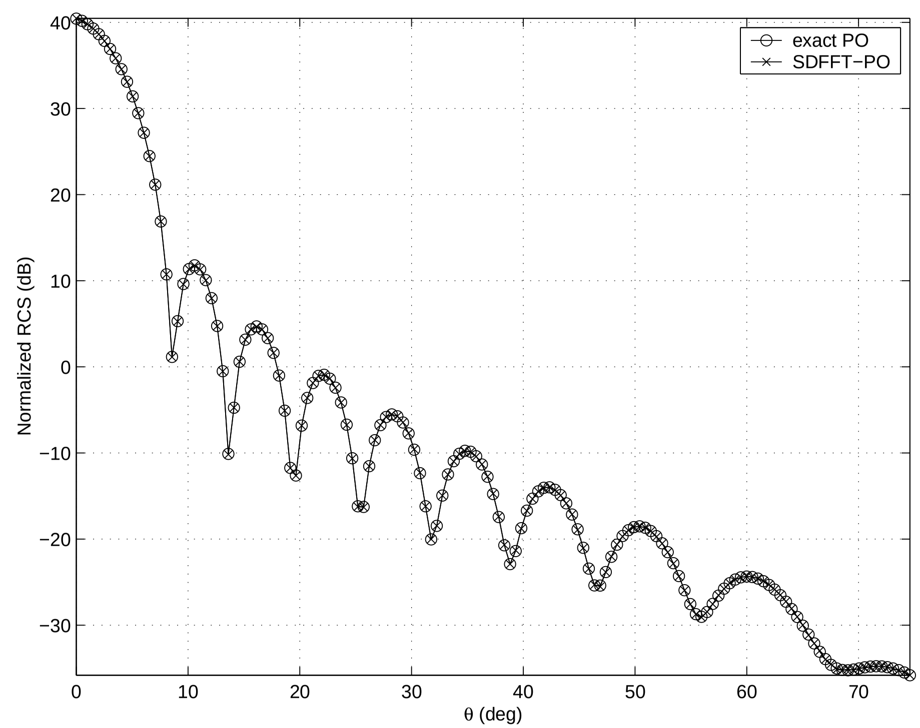
300 MHz



Comparison with FISC



150 MHz



50 MHz

N_r	N_k	Max. Frequency (MHz)	Memory (MB)	CPU Time (s)
15587	15519	300	118	87
31415	30397	600	181	156
65798	60761	1200	373	346

Conclusions

- A sparse data fast Fourier transform (SDFFT) algorithm that can efficiently map sparse spatial data to arbitrary points in the spectral domain has been presented.

- The 3D parabolic antenna problem is solved to demonstrate its $O(N \log N)$ complexity and its ability to produce broadband data.

  Our future work may involve accelerating the current code, improving interpolation complexity, and applying the method to solving large-scale integral equations and inverse problems. A specialization for the Ewald's sphere may be worthwhile to avoid extra computational and memory costs related to 3D interpolation.