NUFFT, Discontinuous Fast Fourier Transform, and Some Applications

Qing Huo Liu

Department of Electrical and Computer Engineering

Duke University

Durham, NC 27708

Outline

- Motivation
- NUFFT Algorithms
- FFT for Discontinuous Functions
- Applications in Solution of Wave Equations, Sensing, and Imaging
- Summary

MOTIVATION

- **Develop a fast method to calculate Discrete Fourier Transform (DFT) of nonuniformly sampled data**
  - Regular FFT algorithms do not apply
  - Straightforward DFT requires $O(N^2)$ for the forward transform, and $O(N^3)$ for the inverse transform

- **Develop a fast Fourier transform algorithm for discontinuous functions**
  - Regular DFT and FFT has slow convergence of $O(1/N)$
  - The "discontinuous" FFT (DFFT) method has exponential convergence while requiring only $O(N \log N)$ operations

- **Engineering applications of the NUFFT and DFFT algorithms**

  SAR, GPR, CT, MRI

Outline

- Motivation
- NUFFT Algorithms
- FFT for Discontinuous Functions
- Applications in Solution of Wave Equations, Sensing, and Imaging
- Summary

NUFFT Algorithms

- **Regular FFT Algorithms: A fast method for DFT**

  - First proposed by Cooley and Tukey (1965)

  - Direct calculation of Discrete Fourier Transform

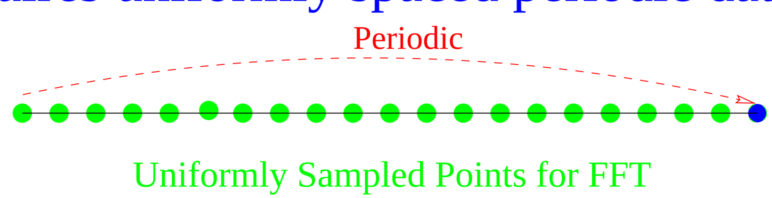
$$f_j = \sum_{k=-N/2}^{N/2-1} \alpha_k e^{i2\pi kj} \quad \text{for} \quad j = -N/2, \cdots, N/2-1$$

  requires N^2 arithmetic operations.

  - In FFT, number of arithmetic operations $0.5N\log_2 N$.

- **Limitation of Regular FFT Algorithms**

  - FFT requires uniformly spaced periodic data



- **The Nonuniform Discrete Fourier Transform**

$$f_j = F(\alpha)_j = \sum_{k=-N/2}^{N/2-1} \alpha_k e^{it_k \cdot \omega_j} \quad \text{for} \quad j = -N/2, \cdots, N/2-1,$$

  - Frequency samples $\omega = \{\omega_{-N/2}, \cdots, \omega_{N/2-1}\}$, $\omega_j = 2\pi j/N \in [-\pi, \pi]$ are uniform
  - Time samples $t = \{t_{-N/2}, \cdots, t_{N/2-1}\}$, $t_k \in [-N/2, N/2]$ are nonuniform
  - Regular FFT does not apply
  - Nonuniform data is common in applications
  - Direct calculation is very expensive

- **NUFFT algorithms are fast methods with $O(mN \log_2 N)$ arithmetic operations**

PREVIOUS METHODS

- **Dutt & Rokhlin's Method (1993)**

  - Interpolation involving a Gaussian function

$$F(\omega) = e^{-b\omega^2} e^{i\omega\tau} \quad \text{for} \quad \omega \in [-\pi, \pi]$$

  (where $b > 1/2$ and τ is a real number) by a small number of equally spaced points on the unit circle.

- **Beylkin's Method (1995)**

  - Interpolation using multiresolution analysis (MRA)

- **Liu & Nguyen—Least Square Interpolation (1997, 1998)**

  - Optimal in the least-square sense
  - A new class of matrices: **Regular Fourier Matrices**
  - Highly accurate and with the same complexity

Nonuniform Fast Fourier Transforms (NUFFT)

- **Fast Algorithm for Summation**

$$f_j = F(\alpha)_j = \sum_{k=-N/2}^{N/2-1} \alpha_k e^{it_k \cdot \omega_j} \quad \text{for} \quad j = -N/2, \cdots, N/2-1,$$

- Frequency samples $\omega = \{\omega_{-N/2}, \cdots, \omega_{N/2-1}\}$,
  $\omega_j = 2\pi j/N \in [-\pi, \pi]$ are uniform
- Time samples $t = \{t_{-N/2}, \cdots, t_{N/2-1}\}$, $t_k \in [-N/2, N/2]$
  are nonuniform

- **Our NUFFT Algorithm:** - Introduce a finite sequence

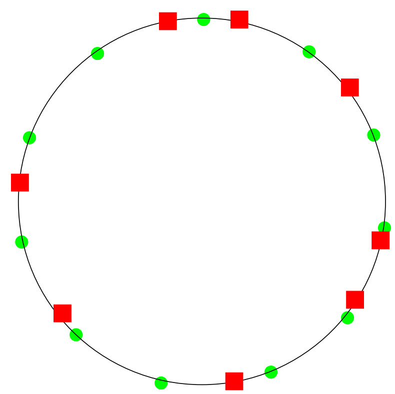
$$S(j) = s_j e^{i2\pi\tau j/N} \equiv s_j z^{jm\tau} \quad \text{for} \quad j = -N/2, \cdots, N/2-1$$

where the "accuracy factors" $0 < s_j \leq 1$ are chosen to minimize the approximation error. Use least-square to approximate this sequence by a small number of uniform points.

LEAST SQUARE INTERPOLATION

Interpolation of Unequally Spaced Points by

Uniform Points on a Unit Circle



- **Find the least square solution x_ℓ of**

$$s_j z^{jm\tau} = \sum_{\ell=-q/2}^{+q/2} x_\ell(\tau) z^{j([m\tau]+\ell)}$$

where $z = e^{i2\pi/mN}$. The **oversampling factor** $m \geq 2$.

- **We use** $(q+1)$ **uniform points to interpolate one point**
  - Number of unknowns $(q+1)$
  - Number of equations N
- **Since** $(q+1) << N$**, this is an over-determined system**

$$Ax(\tau) = v(\tau)$$

$$A_{j\ell} = z^{j(\ell+[m\tau])}, \quad v_j(\tau) = s_j z^{jm\tau}$$

- **Note** A_{jk} **is a function of** τ
- **Least square solution**

$$x(\tau) = F^{-1}a(\tau)$$

$$F = A^{\dagger}(\tau)A(\tau), \quad a(\tau) = A^{\dagger}(\tau) \cdot v(\tau)$$

- **Is** F **a function of** τ **?**

- **Matrix** $F = A^\dagger(\tau)A(\tau) =$

$$\begin{bmatrix} N & \frac{z^{-N/2} - z^{N/2}}{1-z} & \cdots & \frac{z^{-qN/2} - z^{qN/2}}{1-z^q} \\ \frac{z^{N/2} - z^{-N/2}}{1-z^{-1}} & N & \cdots & \frac{z^{-(q-1)N/2} - z^{(q-1)N/2}}{1-z^{q-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{z^{qN/2} - z^{-qN/2}}{1-z^{-q}} & \frac{z^{(q-1)N/2} - z^{-(q-1)N/2}}{1-z^{-(q-1)}} & \cdots & N \end{bmatrix}$$

- **The regular Fourier matrices** $F(m, N, q)$
  - It has a remarkable property:

    $F(m, N, q)$ is **independent** of τ.

- **Therefore, for all time sample points,** F **only need to be calculated once.**

- **Vector** a **is, for** $\ell = -q/2, \cdots, q/2$

$$a_\ell(\tau) = \sum_{j=-N/2}^{N/2-1} s_j e^{i\frac{2\pi}{Nm}(\{m\tau\}-\ell)j}$$

- **In general, vector** a **has to be evaluated by the above series.**

  - For some special accuracy factors s_j, closed form is possible

Accuracy Factors

- **Accuracy factors s_j are needed in**

$$a_\ell(\tau) = \sum_{j=-N/2}^{N/2-1} s_j e^{i\frac{2\pi}{Nm}(\{m\tau\}-\ell)j}$$

- **Three Different Accuracy Factors Are Used**

(1) Gaussian accuracy factors

$$s_j = e^{-b(\frac{2\pi j}{Nm})^2}$$

Then a_ℓ has to be found by the series.

(2) Cosine accuracy factors

$$s_j = \cos\frac{\pi j}{Nm}$$

then $a_\ell(\tau)$ can be found in closed form

$$a_\ell(\tau) = -i \sum_{\gamma=-1,1} \frac{\sin[\frac{\pi}{m}(\{m\tau\} - \ell + \gamma/2)]}{1 - e^{i\frac{2\pi}{Nm}(\{m\tau\}-\ell+\gamma/2)}}$$

(3) Trivial accuracy factors

$$s_j = 1$$

then $a_\ell(\tau)$ can also be found in closed form

$$a_\ell(\tau) == \frac{e^{-i\frac{\pi}{m}(\{mc\}+q/2-k)} - e^{i\frac{\pi}{m}(\{mc\}+q/2-k)}}{e^{i\frac{2\pi}{Nm}(\{mc\}+q/2-k)}}$$

- **Cosine accuracy factors are more efficient and accurate.**

PROCEDURES OF THE NUFFT ALGORITHM

- **Preprocessing: Compute** $x_\ell(t_k)$ **for all** ℓ **and** k
- **Interpolation: Calculate Fourier coefficients**

$$\eta_n = \sum_{\ell,k,[mt_k]+\ell=n} \alpha_k \cdot x_\ell(t_k)$$

- **Regular FFT: Use uniform FFT to evaluate**

$$T_j = \sum_{n=-mN/2}^{mN/2-1} \eta_n \cdot e^{2\pi inj/mN}$$

- **Scaling: Scale the values to arrive at the approximated NUFFT**

$$\tilde{f}_j = T_j \cdot s_j^{-1}$$

- **The number of arithmetic operations is** $O(mN \log_2 N)$**, where** $m \ll N$**. (Usually** $m = 2$ **and** $q = 8$**.)**

Accuracy of the NUFFT Algorithm

- **L_2 and L_∞ Errors**

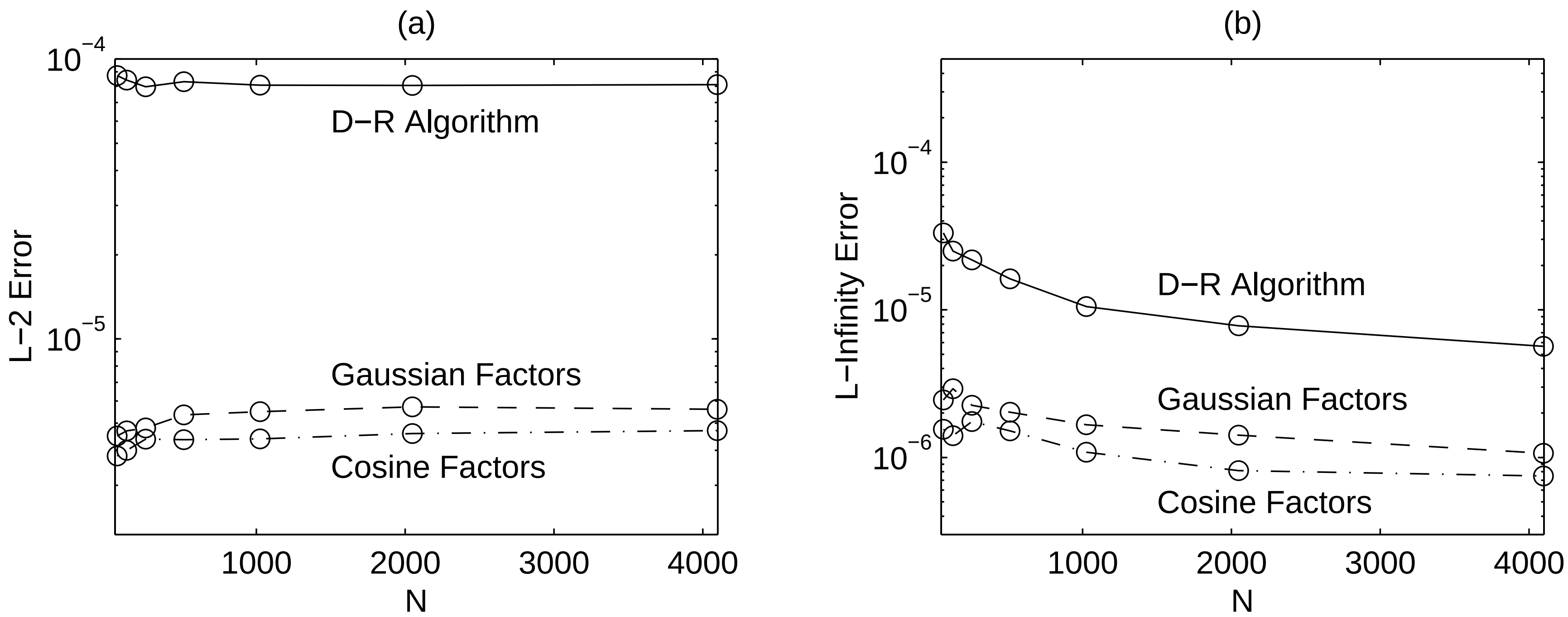
$$E_2 = \sqrt{\sum_{j=-N/2}^{N/2-1} |\tilde{f}_j - f_j|^2 \Big/ \sum_{j=0}^{N-1} |f_j|^2}.$$

$$E_\infty = \max_{-N/2 \le j \le N/2-1} |\tilde{f}_j - f_j| \Big/ \sum_{j=-N/2}^{N/2-1} |\alpha_j|$$

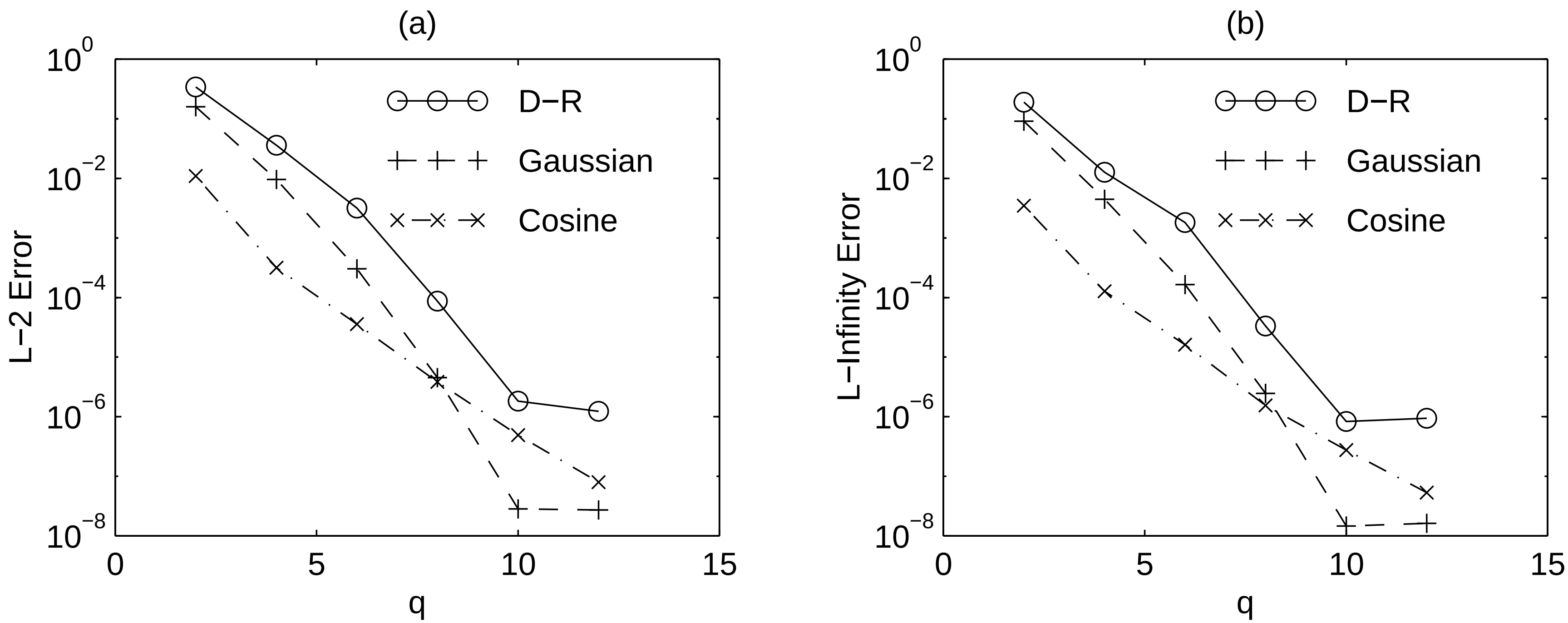
- **For following tests**

  - The time sample points t_k and the data α_k are obtained by a pseudorandom number generator with large variations

E_2 and E_∞ as Functions of N ($q = 8$)



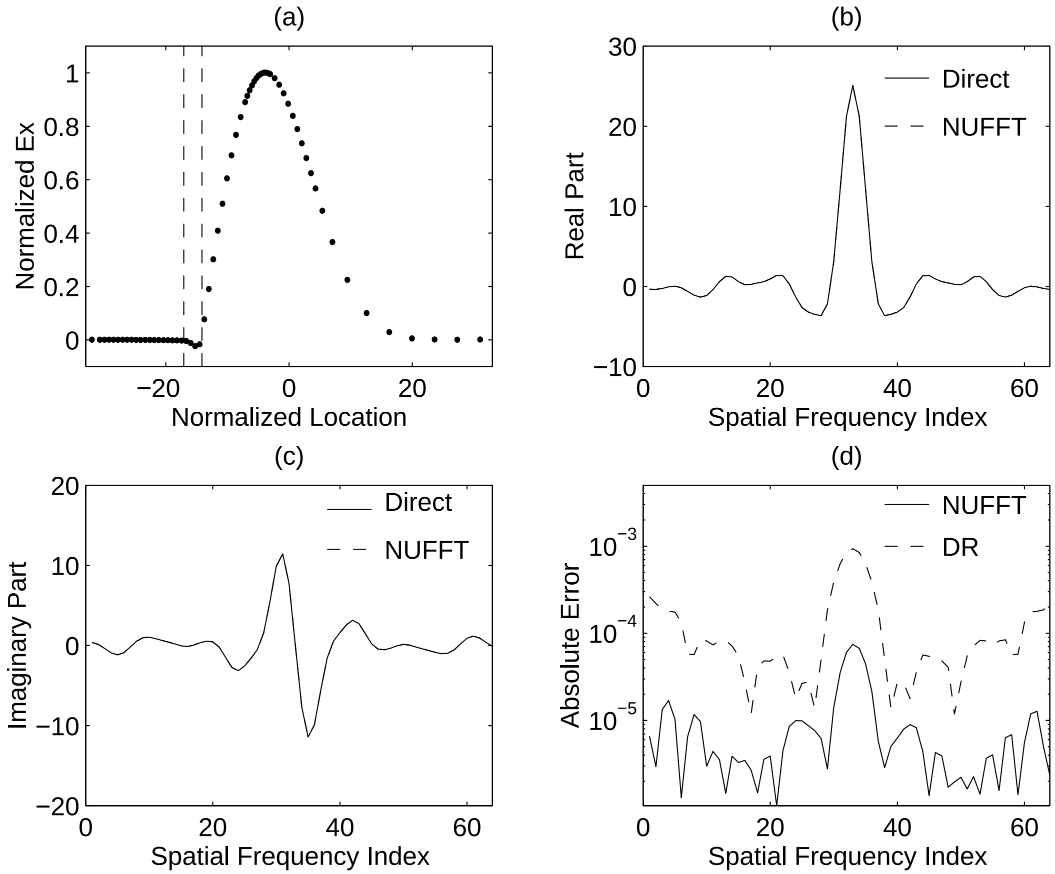
E_2 and E_∞ as Functions of q ($N = 64$)



Observations

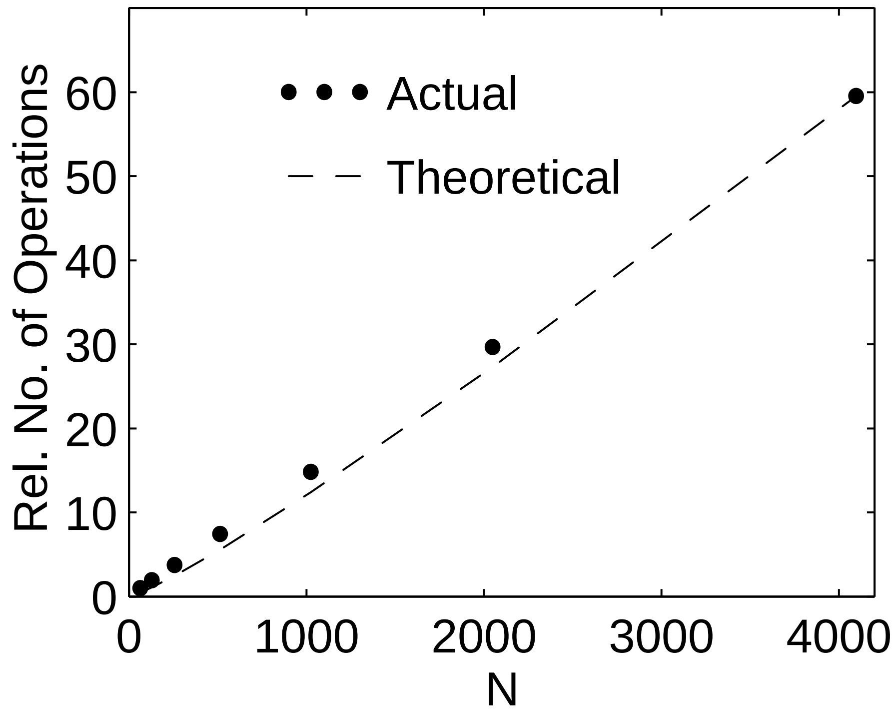
- **NUFFT is optimal in the least square sense.**
  Our algorithm always obtains a higher accuracy than the previous algorithm, while the number of operations is comparable.
- Cosine accuracy factors are more efficient than Gaussian accuracy factors since $a(\tau)$ can be found in closed form.
- Cosine accuracy factors are more accurate than Gaussian accuracy factors for $q \leq 8$.

EM Field Near A Sharp Discontinuity



(a) Spatial distribution of transient EM field near a conductive dielectric slab. **(b)** Real and **(c)** imaginary parts of the (spatial) spectrum. **(d)** Absolute errors of NUFFT.

Computation Complexity



Relative number of operations as a function of N. Both input data and the locations of the sampling points are random. The dashed curve is the theoretically predicted curve $O(N \log_2 N)$ passing through the last point.

Summary of NUFFT

- Direct evaluation of nonuniform DFT is expensive, requiring $O(N^2)$ arithmetic operations.
- Through least-square interpolation, we discover a new class of matrices, the **Regular Fourier Matrices** $F(m, N, q)$.
- The NUFFT algorithm proposed is accurate as it has a least-square error in the interpolation of the basis.
- Other related forward and inverse NUDFTs can be also calculated by the NUFFT.
- The NUFFT algorithm is a fundamental technique useful to many other applications.

Outline

- Motivation
- NUFFT Algorithms
- FFT for Discontinuous Functions
- Applications in Solution of Wave Equations, Sensing, and Imaging
- Summary

FFT for Discontinuous Functions: Motivation

- For smooth periodic functions, the FFT provides a high accuracy.
- FFT results have greatly reduced accuracy for discontinuous functions.
- Examples: Electromagnetic field in a discontinuous medium.
- **The source of inaccuracy**

  -Trapezoidal rule in the Fourier integration.

  -Error is proportional to $O(\frac{1}{N})$
- **Methods for FFT of discontinuous functions** (Fan/Liu, 2001; 2004)

  -Sorets (1995) treats piecewise constant functions

  -This work is an extension to piecewise smooth functions

Formulation of DFFT

- **Fourier Transform** of $f(x)$ (a piecewise smooth function)

$$\hat{f}(n) = \int_0^1 f(x) e^{-i2\pi nx} dx, \quad -\frac{N}{2} < n \leq \frac{N}{2} - 1$$

- **Integration** in L sections

$$\hat{f}(n) = \sum_{l=1}^{L} \int_{x_{l-1}}^{x_l} f(x) e^{-i2\pi nx}\, dx$$

- By change of variables, each section can be evaluated by **Gaussian Legendre quadrature**

$$\int_{-1}^{1} y(t)\, dt \cong \sum_{k=1}^{q} y(t_k)\, \omega_k$$

- **Summation**

$$\hat{f}(n) \cong \sum_{l=1}^{L} b^l \sum_{k=1}^{q} \omega_k \, f(t_k^l) \, e^{-i2\pi n t_k^l}$$

- **However, here $\{t_k^l\}$ are nonuniform.**

- **Lagrange interpolation to a uniform grid**

$$g(x) = \sum_{m=1}^{p} g(x_m)\,\delta_m(x), \qquad \delta_m(x) = \prod_{\substack{n=1\\ n\neq m}}^{p} \frac{x - x_n}{x_m - x_n}$$

- **Double interpolation**

$$\hat{f}(n) = \sum_{l=1}^{L} b^l \sum_{k=1}^{q} \omega_k^l \left(\sum_{m_1=1}^{p_1} f(t_{m_1}^l)\delta_{m_1}(t_k^l) \right) \sum_{m=1}^{p} e^{-i2\pi n t_m^l} \delta_m(t_k^l)$$

- **Then it can be evaluated by the standard FFT**

$$\hat{f}(n) = \sum_{m=1}^{\nu N} g_m\, e^{-i2\pi n x_m}$$

$$g_m = \sum_{l=1}^{L} b^l \sum_{k=1}^{q} \left(\sum_{m_1=1}^{p_1} f(t_{m_1}^l)\delta_{m_1}(t_k^l) \right) \omega_k^l\, \delta_m(t_k^l)$$

ν is sampling factor ($\nu = 2$ in our calculation)

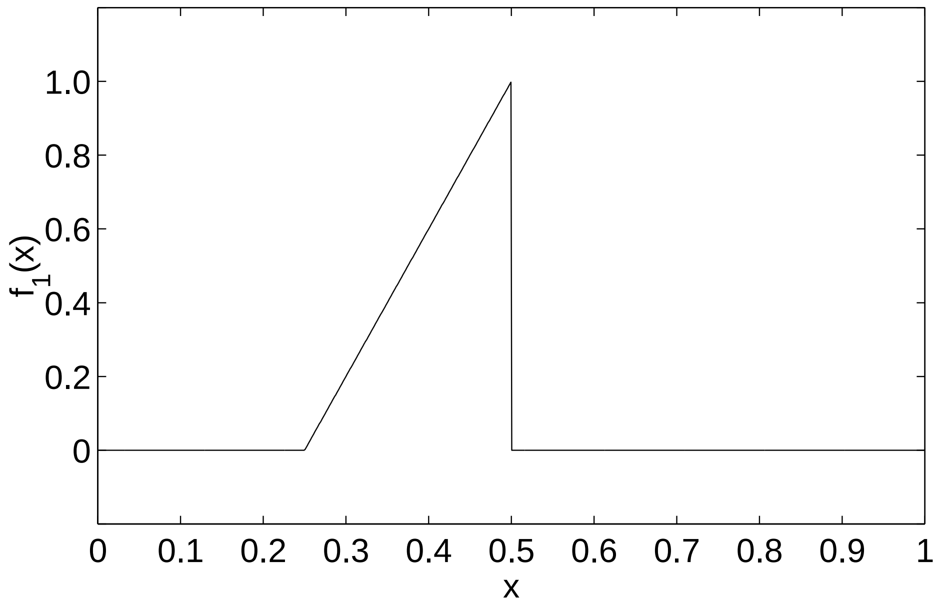
- **Advantages of double interpolation procedure**
  - Nonuniform FFT;
  - Allows a lower order interpolation for the slowly varying function $f(x)$;
  - Allows other efficient algorithms for interpolation of $f(x)$, if needed.

Implementation and complexity of the DFFT algorithm

- **Steps:**
  - Initialization of $\delta_{m_1}(t_k^l)$ and $\delta_m(t_k^l)$. (This preprocessing is needed only once). Complexity $O(Np^2)$.
  - Calculation of g_m. The complexity is $O(Np)$.
  - Calculation of $\hat{f}(n)$ in (9) by a standard FFT. The complexity is $O(\nu N \log N)$.
- **The total complexity is $O(Np + \nu N \log N)$ for last two steps.** The preprocessing need be done only once.

Numerical Examples of DFFT

Example 1: Triangle Function



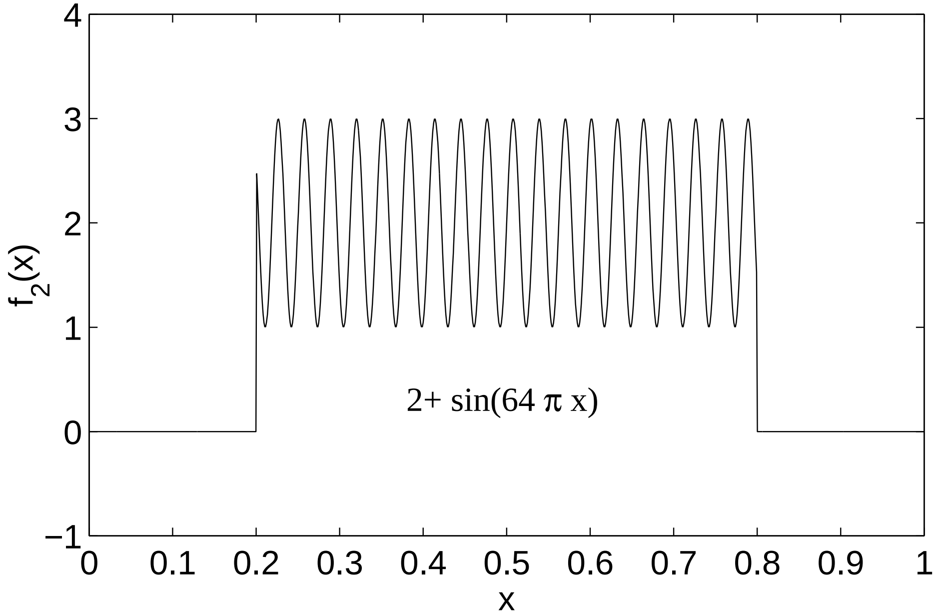
$$f_1(x) = \begin{cases} \frac{x - x_1}{x_2 - x_1} & x_1 \leq x \leq x_2 \\ 0 & \text{elsewhere} \end{cases}$$

Table 1. Errors and Run Times for Example 1

(Double Precision)

N	Errors (E_∞)		Timings (ms)			
	This paper	Direct	Init.	Eval.	FFT	Direct
64	1.130e-13	9.683e-03	51.0	1.30	0.51	0.13
128	1.120e-13	4.824e-03	102.	2.60	1.11	0.26
256	1.130e-13	2.408e-03	203.	5.21	2.47	0.58
512	1.120e-13	1.203e-03	406.	10.5	5.89	1.24

Example 2: Sinusoidal Function



$$f_2(x) = \begin{cases} \alpha + \sin(2\pi\beta x) & x_1 \le x \le x_2 \\ & \text{elsewhere} \end{cases}$$

Table 2. Errors and Run Times for Example 2

(Double Precision)

N	Errors (E_∞)		Timings (ms)			
	This paper	Direct	Init.	Eval.	FFT	Direct
512	1.471e-11	5.820e-02	158.	4.28	5.81	1.25
1024	1.080e-12	2.920e-02	333.	8.41	13.0	2.67

Example 3: 2-D Function

$$f(x, y) = f_1(x)\, f_2(y)$$

Table 3. Errors for the 2-D Problem in Example 3

(Double Precision)

N $\times$ M	Errors (E_∞)	
	This paper	Direct
128 $\times$ 512	2.916e-12	1.697e-03
256 $\times$ 1024	2.700e-13	8.511e-04

Summary of the DFFT

- **A fast DFFT algorithm has been developed for the evaluation of Fourier transform of piecewise smooth functions.**
- **DFFT can achieve NUFFT: It is applicable to both uniformly and nonuniformly sampled data.**
- **The complexity of algorithm is** $O(Np + \nu N \log N)$ **plus** $O(Np^2)$ **for precalculation.**
- **Numerical results demonstrate the efficiency and accuracy.**

Outline

- Motivation
- NUFFT Algorithms
- FFT for Discontinuous Functions
- Applications in Solution of Wave Equations, Sensing, and Imaging
- Summary

Application 1: Integral Equation Solution by the CGFFT Method

- **1-D EM scattering problem** Plane wave scattering from a slab of finite width
- **Integral equation**

$$E^{inc}(x) = E(x) + \int_0^l G(x - x')J(x')\,dx$$

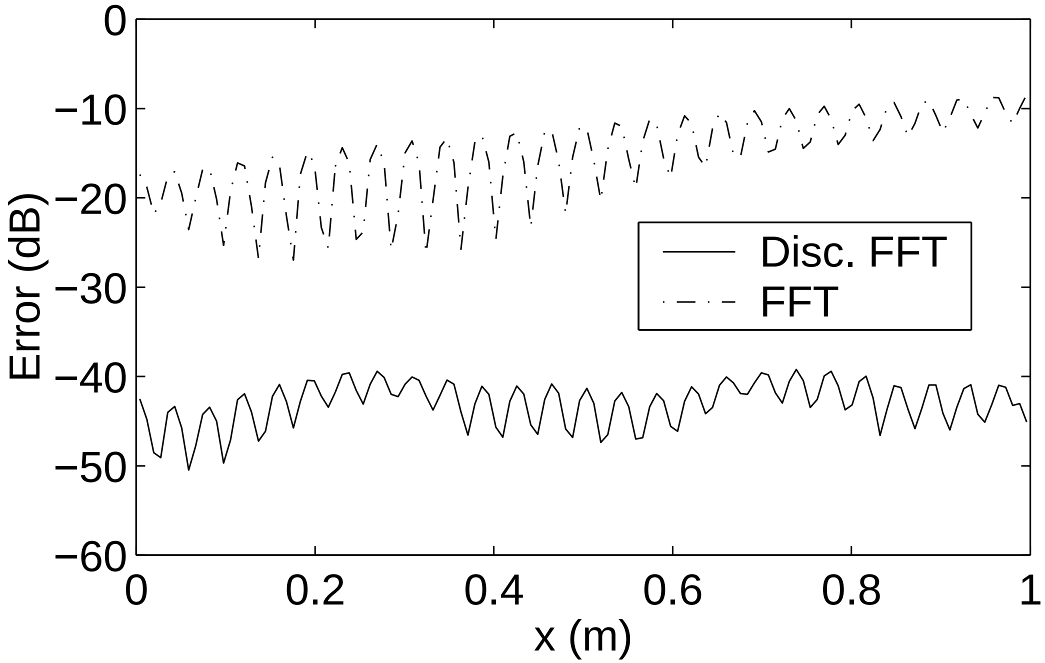
  - $J(x) = k_0^2[\epsilon_r(x) - 1]E(x)$ is the unknown equivalent current
  - $G(x - x')$ is the 1-D Green's function in free space
- **Convolution integral is evaluated by Fourier transform**

$$E^{inc}(x) = E(x) + \mathcal{F}^{-1}\left\{\mathcal{F}\{G(x)\}\,\mathcal{F}\{J(x)\}\right\}$$

$\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the forward and inverse Fourier transform.

Example 1:

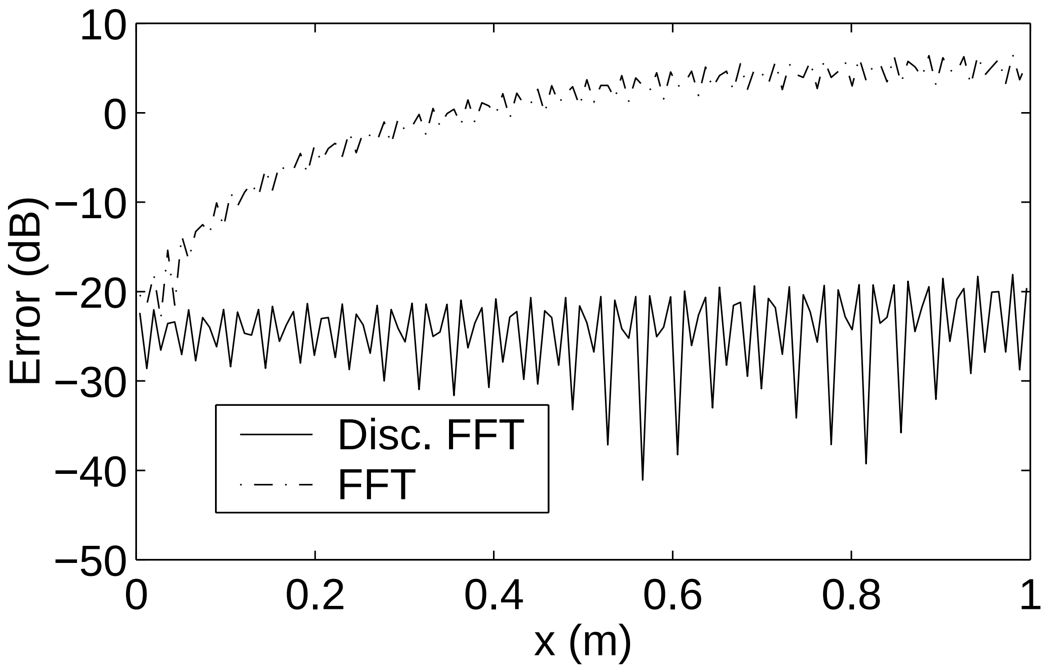
Comparison of CGFFT and DFFT at a High Sampling Rate



Comparison of CGFFT algorithms for dielectric slab with a low ϵ_r contrast. $\epsilon_r = 2$, $f = 2.75$ GHz.

- **Sampling density: 10 PPW**

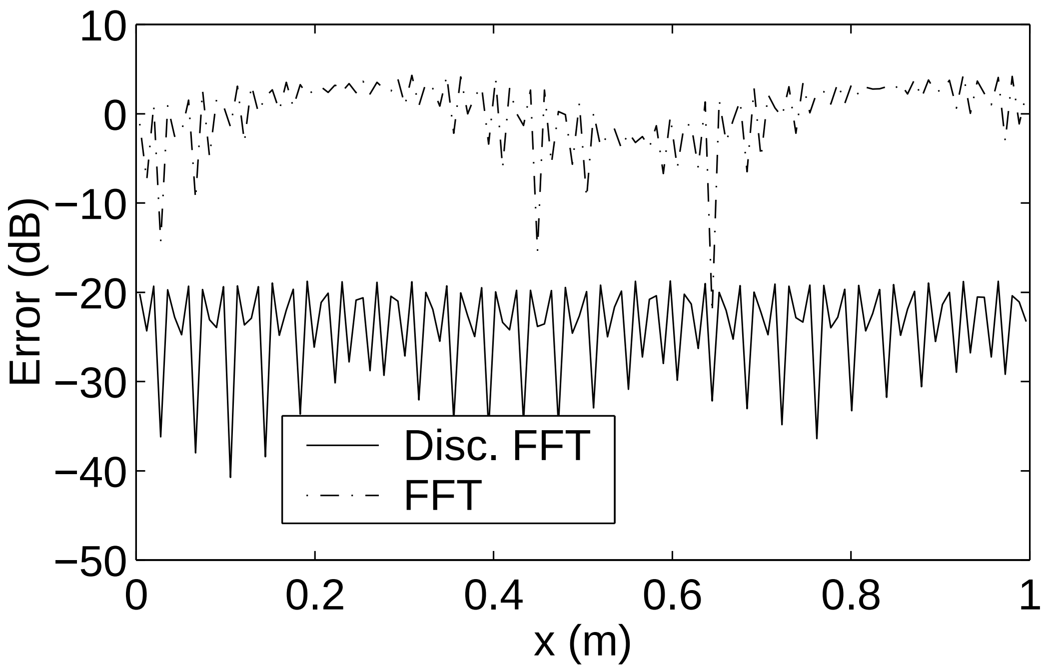
Example 2: Lower Sampling Rate (Higher Frequency)



Same as Example 1 except $f = 5.5$ GHz.

- **Sampling density: 5 PPW**

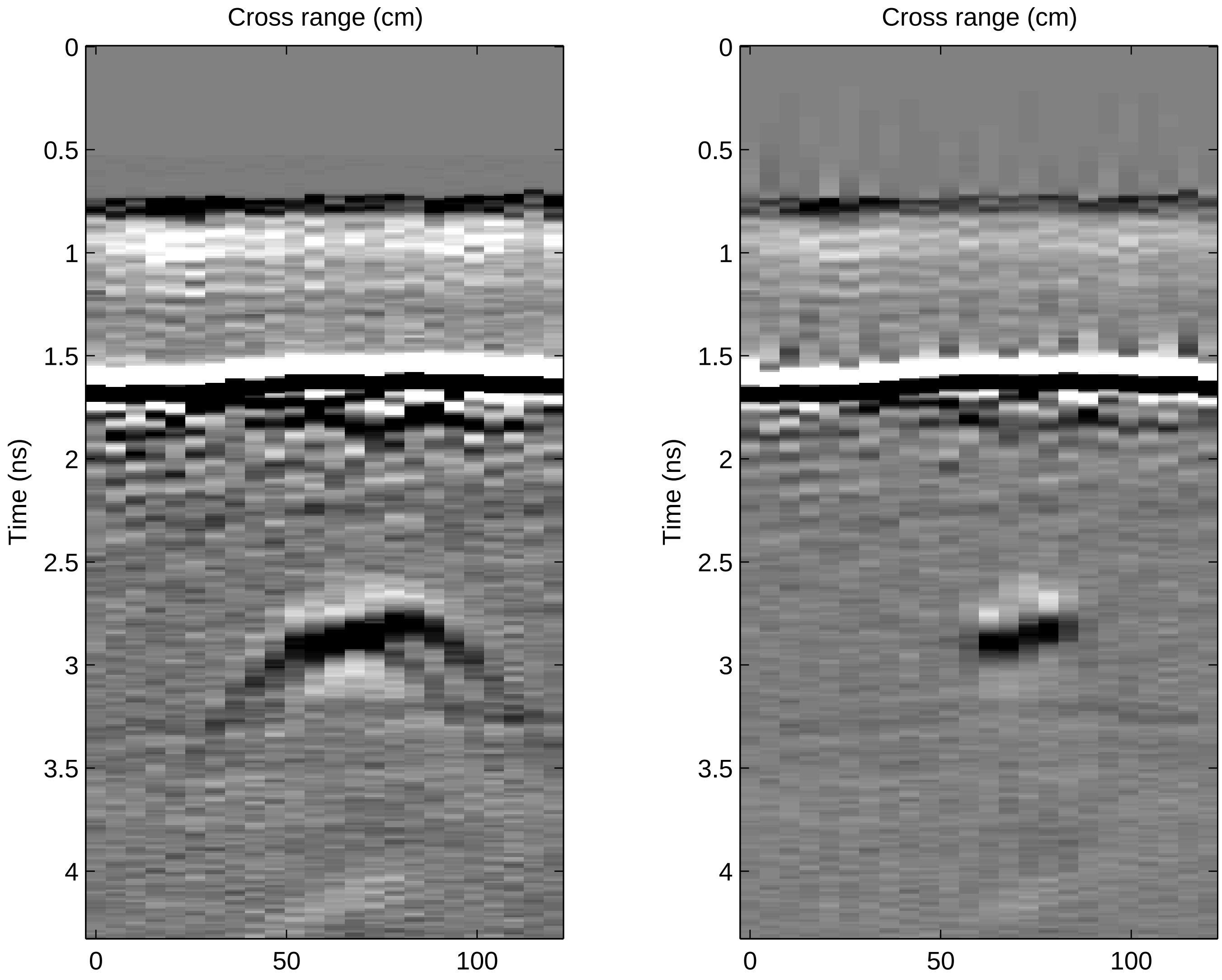
Example 3: Lower Sampling Rate and High Contrast



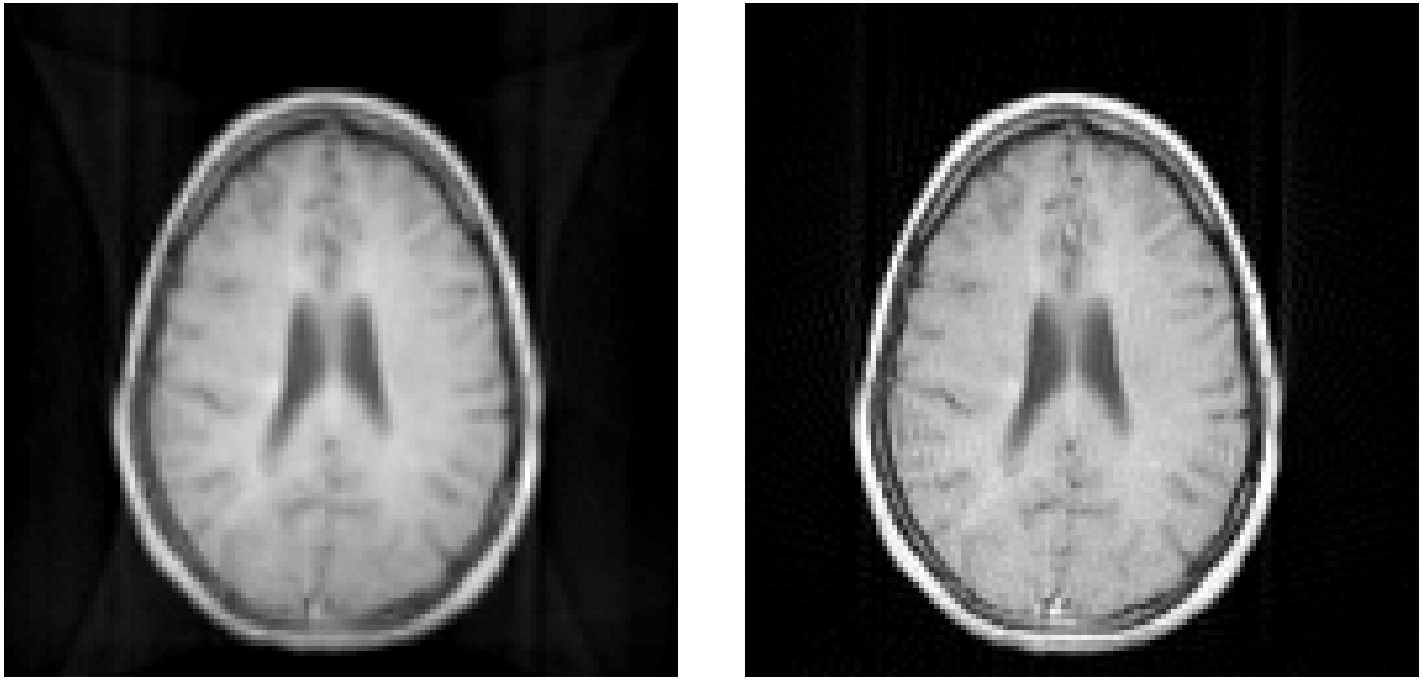
Same as Example 1 except with a higher contrast, $\epsilon_r = 8$.

- **Sampling density: 5 PPW**

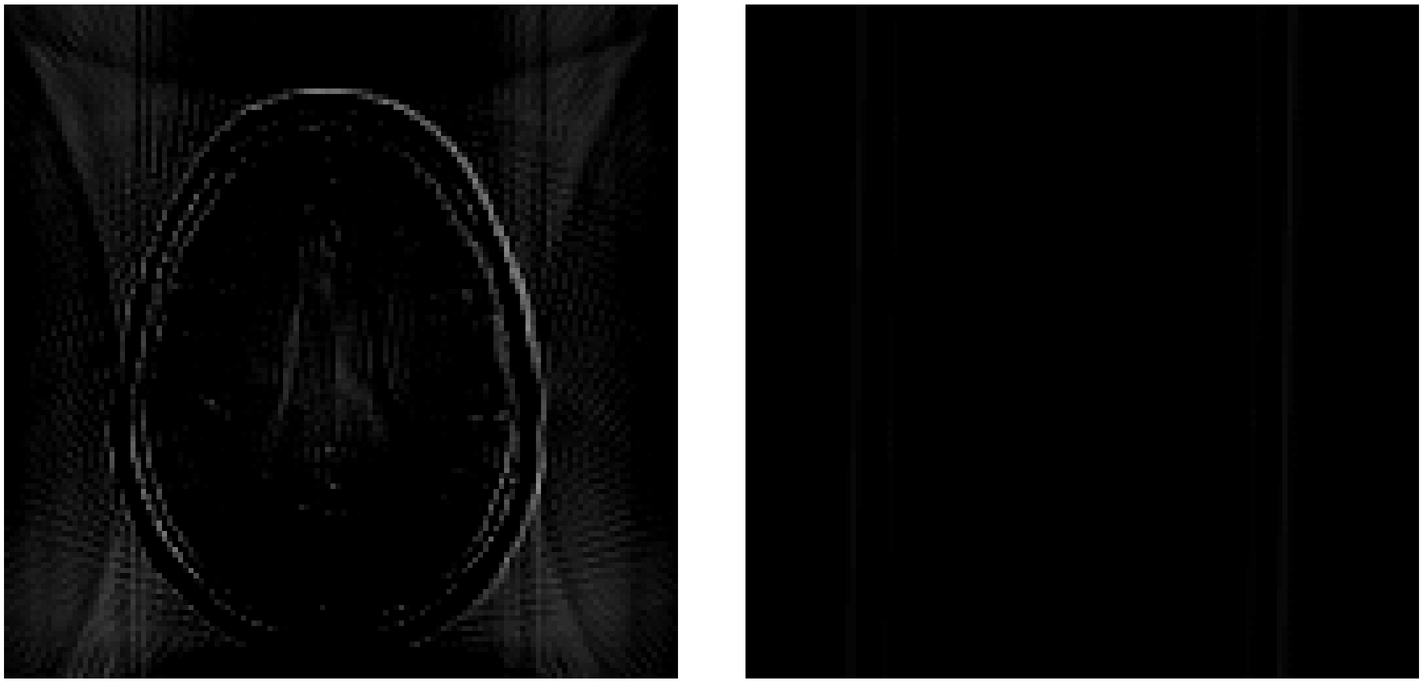
Application 2: Ground Penetrating Radar Using NUFFT



Application 3: MRI Image Reconstruction

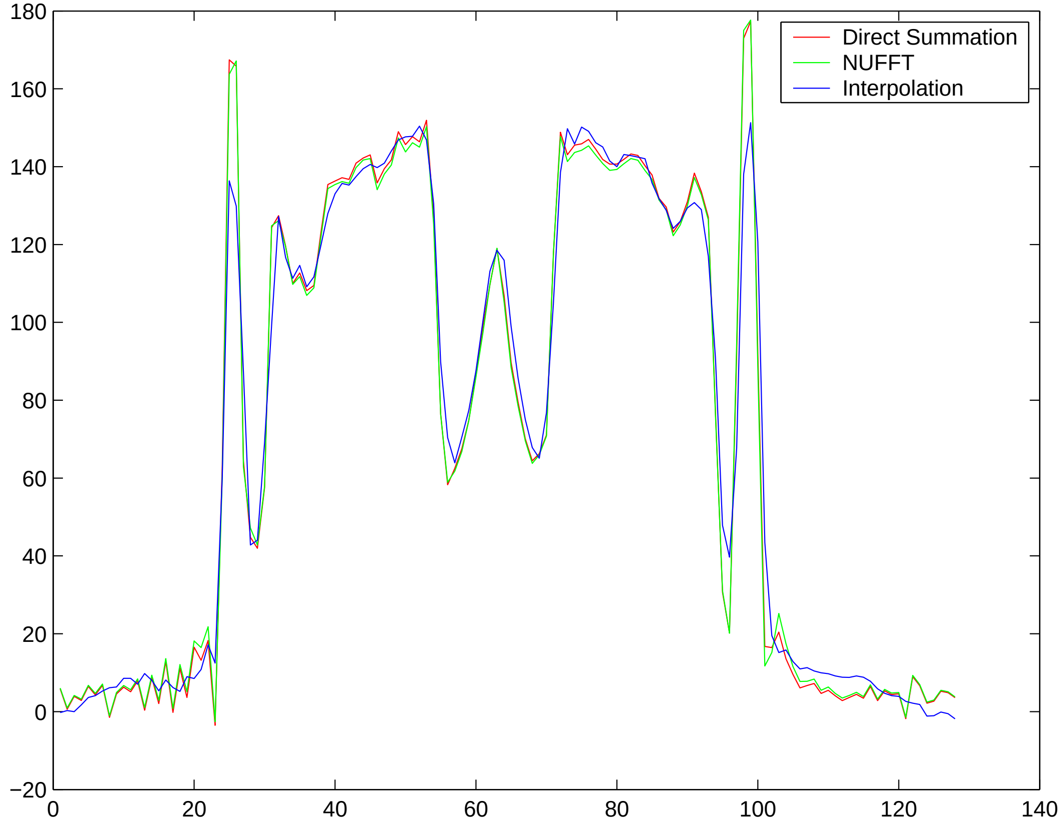


Conventional and NUFFT reconstructed results



Error in the conventional method and in the NUFFT

Error Comparison



Comparison of the conventional and NUFFT reconstructions in L_2 error.
NUFFT: 1.49%, Interpolation: 12.25%

Summary and Conclusions

- **NUFFT algorithms with** $O(N \log N)$ **operations have been developed in recent years and received considerable attention.** We presented a simple method based on least-square interpolation of the basis, inspired by the original work by Dutt and Rokhlin.
- **A fast DFFT algorithm has been developed for the Fourier transform for discontinuous functions with** $O(Np + \nu N \log N)$ **operations.**
- **Both NUFFT and DFFT algorithms have many applications:**
  - Numerical solution of wave equations
  - Ground penetrating radar and synthetic aperture radar processing
  - CT and MRI image reconstruction

Acknowledgment

Collaborators:

G. Fan, N. Nguyen, N. Pitsianis, J. Song, X. Sun, X. Tang