Unequally spaced FFT and fast Radon transform

Gregory Beylkin

University of Colorado at Boulder

IPAM
September 14, 2004

Plan

1. The Radon Transform
2. Algorithms for the USFFT
3. Stolt migration
4. Directional filtering of seismograms using Slant Stack (Radon) transform.
5. Various extensions (as time permits).

The Radon Transform

Let us consider the classical Radon transform

$$(Ru)(s,\nu) = \int_{\mathbb{R}^d} u(x)\,\delta(s-\nu\cdot x)\,dx,$$

where $s \in \mathbb{R}$ and $||\nu|| = 1$ and its dual

$$(R^* v)(y) = \int_{||\nu||=1} v(s,\nu)_{|s=\nu\cdot y}\,d\nu.$$

Inversion of the Radon transform using the projection slice theorem: we have

$$\int_{-\infty}^{\infty} (Ru)(s,\nu)\,e^{isp}ds = \int_{\mathbb{R}^d} u(x)\,e^{ip\nu\cdot x}\,dx = \hat{u}(p\nu),$$

where $p \in \mathbb{R}$ and use the Fourier transform (easy on paper!).

Radon's inversion formula

We have (J. Radon, 1917)

$$R^* K R = I,$$

where K is a convolution with the kernel

$$K(s) = \frac{1}{2(2\pi)^d} \int_{-\infty}^{+\infty} |r|^{d-1} \mathrm{e}^{irs}\, dr.$$

This inversion formula was derived in applications (on more than one occasion) without knowing Radon's result.

It is the so-called back-projection algorithm

The generalized Radon Transform

Define

$$(Ru)(s,\nu) = \int_{\mathbb{R}^d} u(x)\, a(x,\nu)\, \delta(s - \phi(x,\nu))\, dx,$$

and its dual

$$(R^*v)(y) = \int_{||\nu||=1} b(y,\nu)\, v(s,\nu)_{|s=\phi(y,\nu)}\, d\nu.$$

With an appropriate selection of b, we have

$$R^*KR = I + T,$$

where T is a smoothing pseudo-differential operator.

Discretization of integrals

Consider the Fourier integral

$$g(\xi) = \int_{-\infty}^{\infty} f(x) e^{ix\xi} dx,$$

or the coefficients of the Fourier series

$$c_k = \int_{-\pi}^{\pi} f(x) e^{ixk} dx,$$

where, for some reason (e.g., discontinuities of f), equal spacing of nodes results in a low order approximation.

Trigonometric sums

Using FFT requires sampling on an equally spaced grid: a significant limitation in many applications.

The direct evaluation of trigonometric sums

$$\hat{g}_n = \sum_{l=1}^{N_p} g_l \, \mathrm{e}^{-2\pi \mathrm{i} N x_l \xi_n},$$

$n = 1, \ldots, N_f$, where $g_l \in \mathbb{C}$, $|\xi_n|, |x_l| \leq \frac{1}{2}$ requires $O(N_f \cdot N_p)$ operations.

Typically $N_f \approx N_p \approx N$ and $N_f \cdot N_p = O(N^2)$. The cost in 2D is $O(N^4)$ and in 3D $O(N^6)$.

USFFT

Computation of the sum may be viewed as an application of the matrix

$$F^0_{ln} = \mathrm{e}^{\pm 2\pi \mathrm{i} N x_l \xi_n},$$

$l = 1, \ldots, N_p$, $n = 1, \ldots, N_f$ to a vector.
A special case is the matrix

$$F_{ln} = \mathrm{e}^{\pm 2\pi \mathrm{i} l \xi_n},$$

$l = -N/2, \ldots, N/2-1$, $n = 1, \ldots, N_f$ and its adjoint.

Algorithms for the fast application of these matrices and their adjoints to vectors (as well as their multidimensional generalizations) constitute Unequally Spaced Fast Fourier Transform (USFFT) algorithms.

References

$\mathcal{O}(N \log N)$ algorithms with a (relatively) large constant

- Low accuracy interpolation for SAR
- Press and Rybicki: "Fast algorithm for spectral analysis of unevenly sampled data", The Astrophysical Journal, 338, 1989 (interpolation)
- Sullivan: "A Technique for Convolving Unequally Spaced Samples Using Fast Fourier Transforms", Sandia Report, Jan.1990 (Taylor expansion)
- Brandt: "Multilevel computations of integral transforms and particle interactions with oscillatory kernels", Comp. Phys. Commun. 1991 (interpolation)

References

$\mathcal{O}(N \log N)$ algorithms with a small constant (2 FFTs $+ C_0 N$)

- Dutt and Rokhlin: "Fast Fourier transform for nonequispaced data", SIAM J. Sci. Comput., 1993 (Gaussian bells)

- Beylkin: "On the fast Fourier transform of functions singularities", 1995, (splines, tight estimates)

$\mathcal{O}(N \log N)$ algorithms with a small constant (1 FFT $+C_1 N$), $C_1 > C_0$

- Dutt, Gu, Rokhlin: Fast algorithms for polynomial interpolation, integration, and differentiation, SIAM J. Numer. Anal., 1996 (an approximation of the Dirichlet kernel)

- Beylkin, Monzon: On approximation of functions by exponential sums, to appear 2004, (an alternative approximation of the Dirichlet kernel)

Orthogonalization of B-splines

Let β^m be the m-th order central B-spline and $\hat{\beta^m}$ its Fourier transform,

$$\hat{\beta^m}(\xi) = \left(\frac{\sin \pi\xi}{\pi\xi}\right)^{m+1}.$$

Consider the periodic function $a^{(m)}$,

$$a^{(m)}(\xi) = \sum_{l=-\infty}^{l=\infty} |\hat{\beta^m}(\xi + l)|^2 = \sum_{l=-m}^{l=m} \beta^{2m+1}(l)\ \mathrm{e}^{2\pi \mathrm{i} l\xi},$$

and the Fourier transform of the Battle-Lemarié scaling function,

$$\hat{\varphi}^{(m)}(\xi) = \frac{\hat{\beta^m}(\xi)}{\sqrt{a^{(m)}(\xi)}}.$$

An approximation of the ideal filter

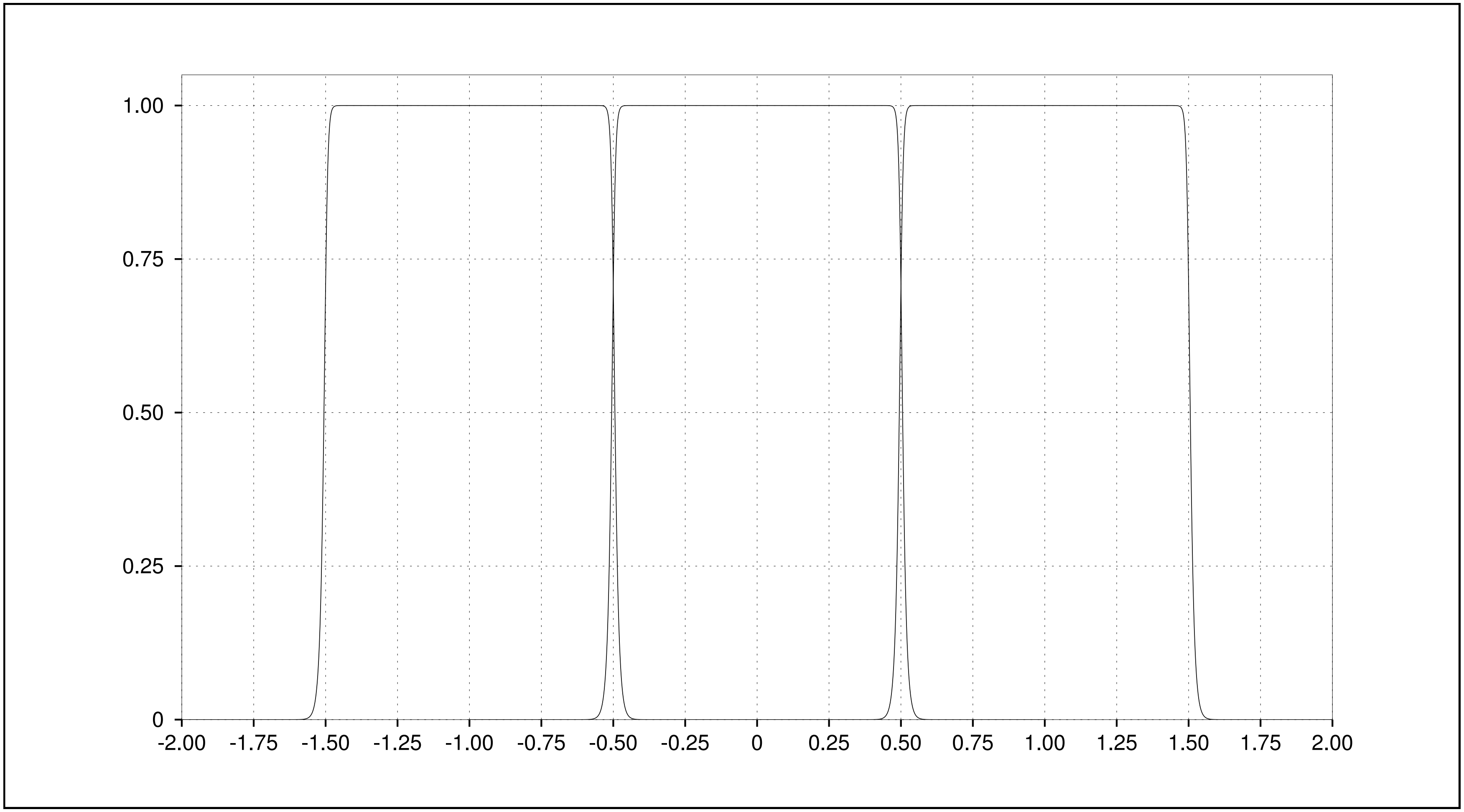
For large m the Battle-Lemarié scaling function is a good approximation to the ideal filter,

$$\hat{\varphi}^{(m)}(\xi) = 1 + O(\xi^{2m+2}),$$

where for the B-spline we have

$$\hat{\beta^m}(\xi) = 1 - \frac{(1+m)\pi^2}{6}\xi^2 + O(\xi^4).$$

Example



The Fourier transform of the Battle-Lemarié scaling function of order $m = 23$.
Shown are functions $\hat{\varphi}^{(m)}(\xi)$, $\hat{\varphi}^{(m)}(\xi + 1)$ and $\hat{\varphi}^{(m)}(\xi - 1)$.

The main points

- We compute a bandlimited version of the function that preserves frequencies within the band, i.e., we multiply the Fourier Transform of the function by an approximate ideal filter

- In the original domain the filter has a large support

- Key point: the filter is applied *partially* in the original domain and *partially* in the Fourier domain

- The convolution with B-splines in the original domain is the projection on B-splines; it accounts for the numerator of the approximate ideal filter

- The denominator is applied in the Fourier domain and amounts to orthogonalization of the expansion (the basis changes from the B-splines to the Battle-Lemarié scaling functions)

Theorem

Let E_∞ be the error in approximating the Fourier transform of the generalized function f by a periodic function $2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}$, for some $j < 0$,

$$E_\infty = \sup_{|\xi|\le\alpha} |2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}} - \hat{f}(2^{-j}\xi)| \;/\; \sup_{|\xi|\le\alpha} |\hat{f}(2^{-j}\xi)|,$$

where F is the Fourier series,

$$F(\xi) = \sum_{k\in\mathbb{Z}} f_k \mathrm{e}^{-2\pi \mathrm{i} k\xi},$$

with coefficients $f_k = \int_{-\infty}^{\infty} f(x)\,{\beta^m}_{kj}(x)\,\mathrm{d}x$.

If $|\hat{f}(\xi)| \le C(1+|\xi|)^\sigma, \quad \sigma < m$, then for any $\epsilon > 0$ we can choose m, the order of the central B-spline, and the parameter $\alpha > 0$ so that for $|\xi| \le \alpha$

$$E_\infty \le \epsilon$$

Algorithm that uses only one FFT (no oversampling)

Let us find g_l, $l = -n, \cdots, n$, such that

$$\hat{g}_k = \sum_{j=-n}^{n} f_j e^{2\pi i k x_j} = \sum_{l=-n}^{n} g_l e^{2\pi i k l/(2n+1)}.$$

We have $(N = 2n+1))$

$$g_l = \frac{1}{N} \sum_{k=-n}^{n} e^{-2\pi i k l/N} \sum_{j=-n}^{n} f_j e^{2\pi i k x_j} = \sum_{j=-n}^{n} f_j \frac{1}{N} \sum_{k=-n}^{n} e^{2\pi i k (x_j - \frac{l}{N})},$$

or

$$g_l = \sum_{j=-n}^{n} D_n(x_j - \frac{l}{N})\, f_j.$$

The Dirichlet kernel

The periodic Dirichlet kernel,

$$D_n(x) \;=\; \frac{1}{N} \sum_{k=-n}^{n} e^{2\pi i k x} \;=\; \frac{\sin N\pi x}{N\,\sin \pi x}\,,$$

where $N = 2n + 1$, can be written as

$$D_n(x) \;=\; G_n(x) + G_n(1-x)\,,$$

where

$$G_n(x) = \frac{\sin(N\pi x)}{N\pi} \sum_{k \geq 0} \frac{(-1)^k}{x+k} = \sum_{k \geq 0} \frac{\sin(N\pi(x+k))}{N\pi(x+k)}.$$

Approximation via exponentials

We approximate G_n in $[0,1]$,

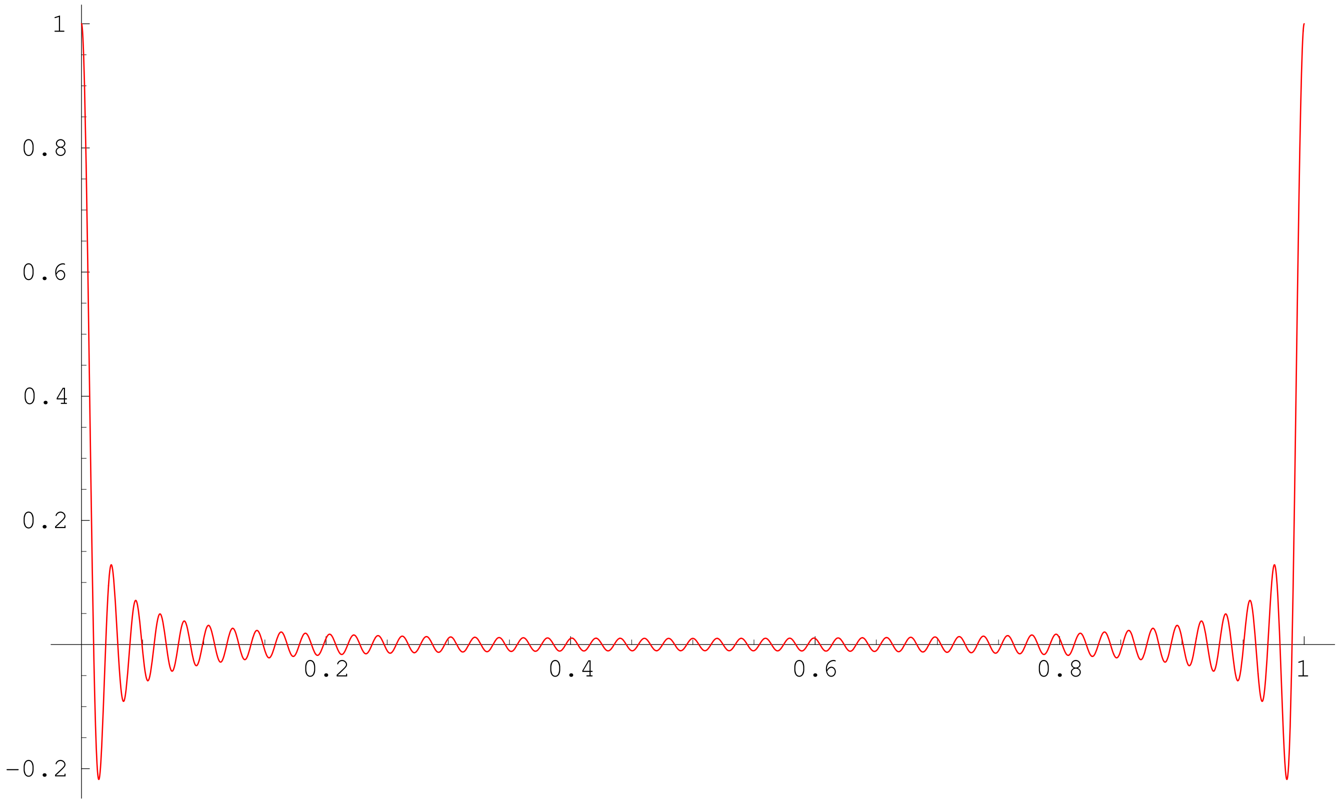
$$|G_n(x) - \sum_{m=1}^{M} \rho_m e^{t_m x}| \leq \epsilon\,,$$

where weights and nodes are complex and $|e^{t_m}| < 1$. The number of terms grows logarithmically with the accuracy and with n, $M = \mathcal{O}(\log n) + \mathcal{O}(\log \epsilon)$.

As a result we obtain the approximation for the Dirichlet kernel,

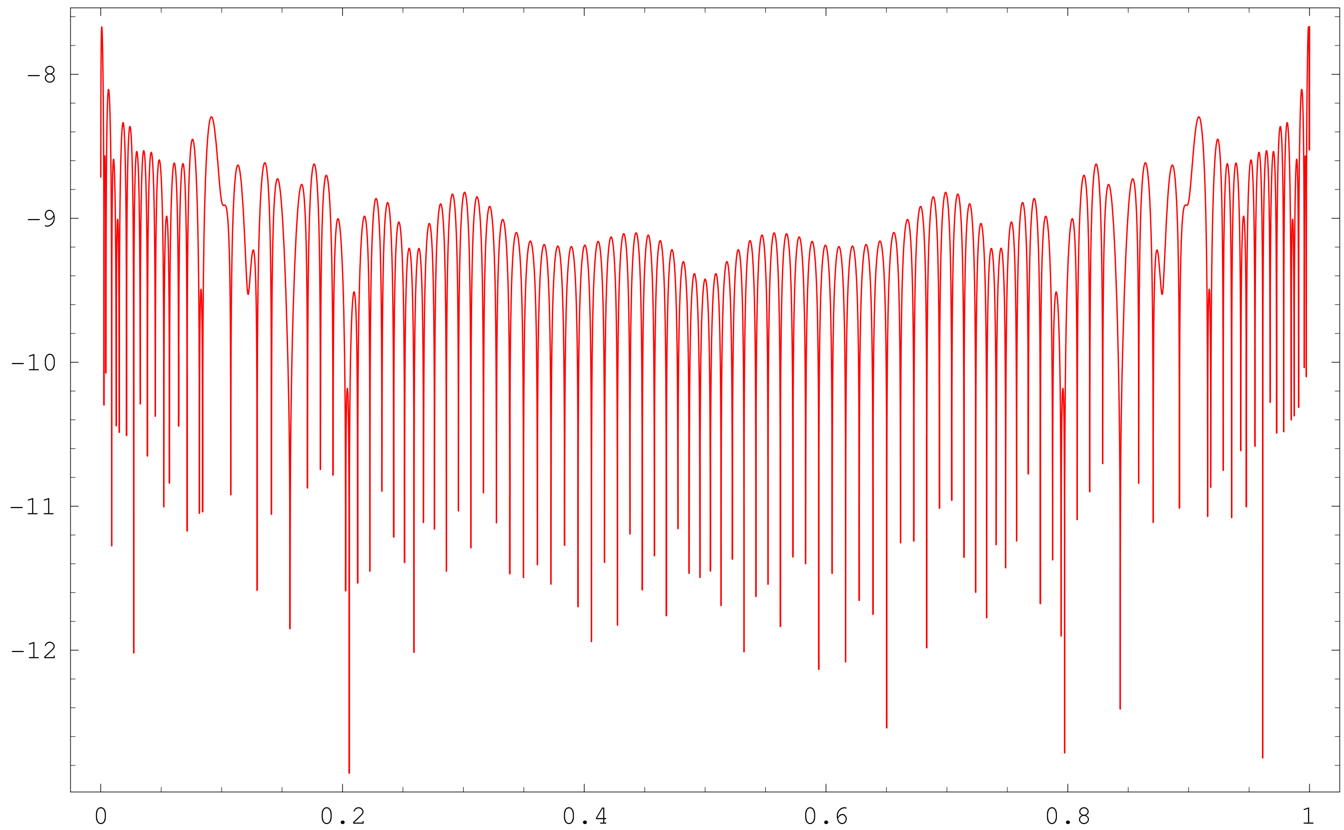
$$|D_n(x) - \sum_{m=1}^{M} \rho_m e^{t_m x} - \sum_{m=1}^{M} \rho_m e^{t_m (1-x)}| \leq 2\epsilon\,.$$

Approximation via exponentials: example



Dirichlet kernel D_{50}

Approximation via exponentials: example



The error ($\log_{10}$) of 44-term approximation of D_{50}.

A fast algorithm

We need to compute the sum

$$g(x_n) = \sum_{l=1}^{L} K(x_n - y_l) f(y_l).$$

Using M-term exponential approximation of the kernel, an elegant algorithm of Rokhlin computes the sum with accuracy ϵ in $\mathcal{O}(2M \cdot (L + N))$ operations, where M is the number of terms in

$$|K(s) - \sum_{m=1}^{M} \rho_m e^{t_m s}| \leq \epsilon \text{ for } s \in [0, 1].$$

(Assuming that the kernel K is an even function, $K(-s) = K(s)$. If not, then approximate on $[-1, 0]$ separately).

Recursion

Split the sum as

$$g(x_n) = \sum_{0 \le y_l \le x_n} K(x_n - y_l) f(y_l) + \sum_{x_n \le y_l \le 1} K(x_n - y_l) f(y_l),$$

approximate the first term as $\sum_{m=1}^{M} w_m q_{n,m}$, where $q_{n,m} = \sum_{0 \le y_l \le x_n} e^{t_m (x_n - y_l)} f(y_l)$. and, similarly, the second term. We observe that

$$q_{n+1,m} = e^{t_m (x_{n+1} - x_n)} \sum_{0 \le y_l \le x_n} e^{t_m (x_n - y_l)} f(y_l) + \sum_{x_n < y_l \le x_{n+1}} e^{t_m (x_{n+1} - y_l)} f(y_l),$$

and, thus, $q_{n,m}$ is computed via the recursion

$$q_{n+1,m} = e^{t_m (x_{n+1} - x_n)} q_{n,m} + \sum_{x_n < y_l \le x_{n+1}} e^{t_m (x_{n+1} - y_l)} f(y_l).$$

Stolt migration for seismics (and SAR)

Stolt migration (solution of the linearized inverse scattering problem within the single scattering assumption)

$$U(z,x,0) = \int_{-\infty}^{\infty} \int_{-\frac{\omega}{c}}^{\frac{\omega}{c}} e^{iz\sqrt{\frac{4\omega^2}{c^2}-k_x^2}+ixk_x}\,\hat{U}(0,k_x,\omega)\,dk_x d\omega,$$

where $\hat{U}(0,k_x,\omega)$ is obtained by taking the Fourier transform of measured data $U(0,x,t)$,

$$\hat{U}(0,k_x,\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U(0,x,t)\,e^{-i\omega t}\,e^{-ik_x x}\,dx\,dt.$$

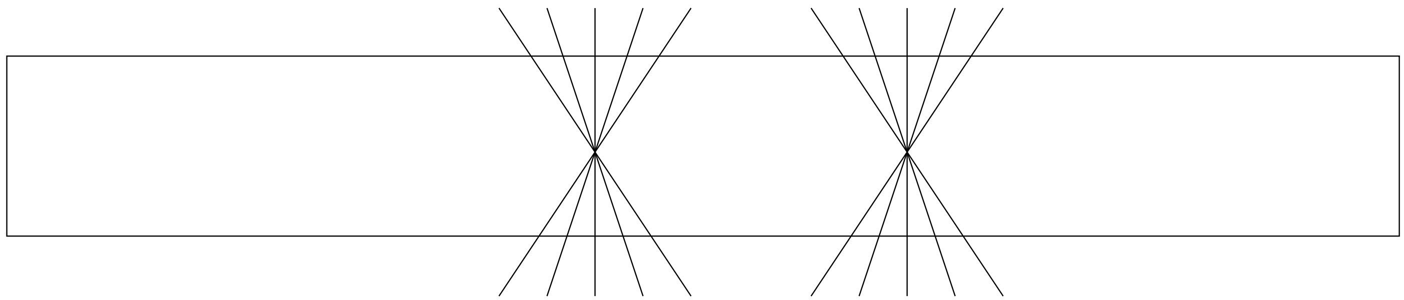
Stolt migration is based on the change of variables $\omega \longrightarrow k_z$,

$$k_z = \sqrt{\frac{4\omega^2}{c^2} - k_x^2}.$$

To discretize k_z using equal spacing we need $\hat{U}(0,k_x,\omega)$ at non-equally spaced nodes.

Slant Stack

Consider functions with the support elongated in one direction (e.g. time)



Algorithm for inversion in the Fourier domain involves

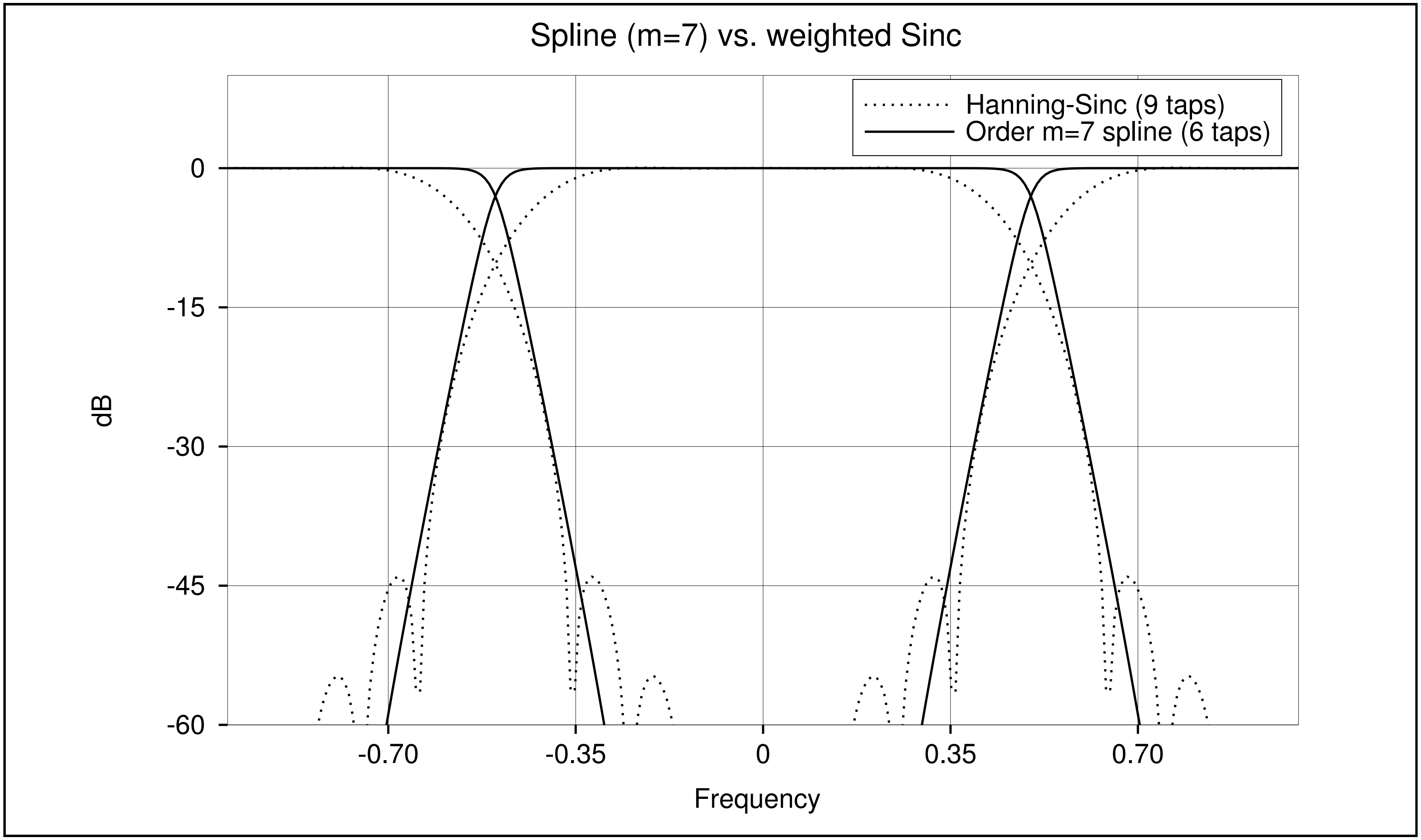
1. USFFT

2. Solving linear systems with positive definite (badly conditioned) self-adjoint Toeplitz matrices.

Since the inverse of a Toepitz matrix can be applied in $\mathcal{O}(N \log N)$ operations (actually, 6 FFTs for the self-adjoint Toeplitz), we obtain Fast Radon Transform

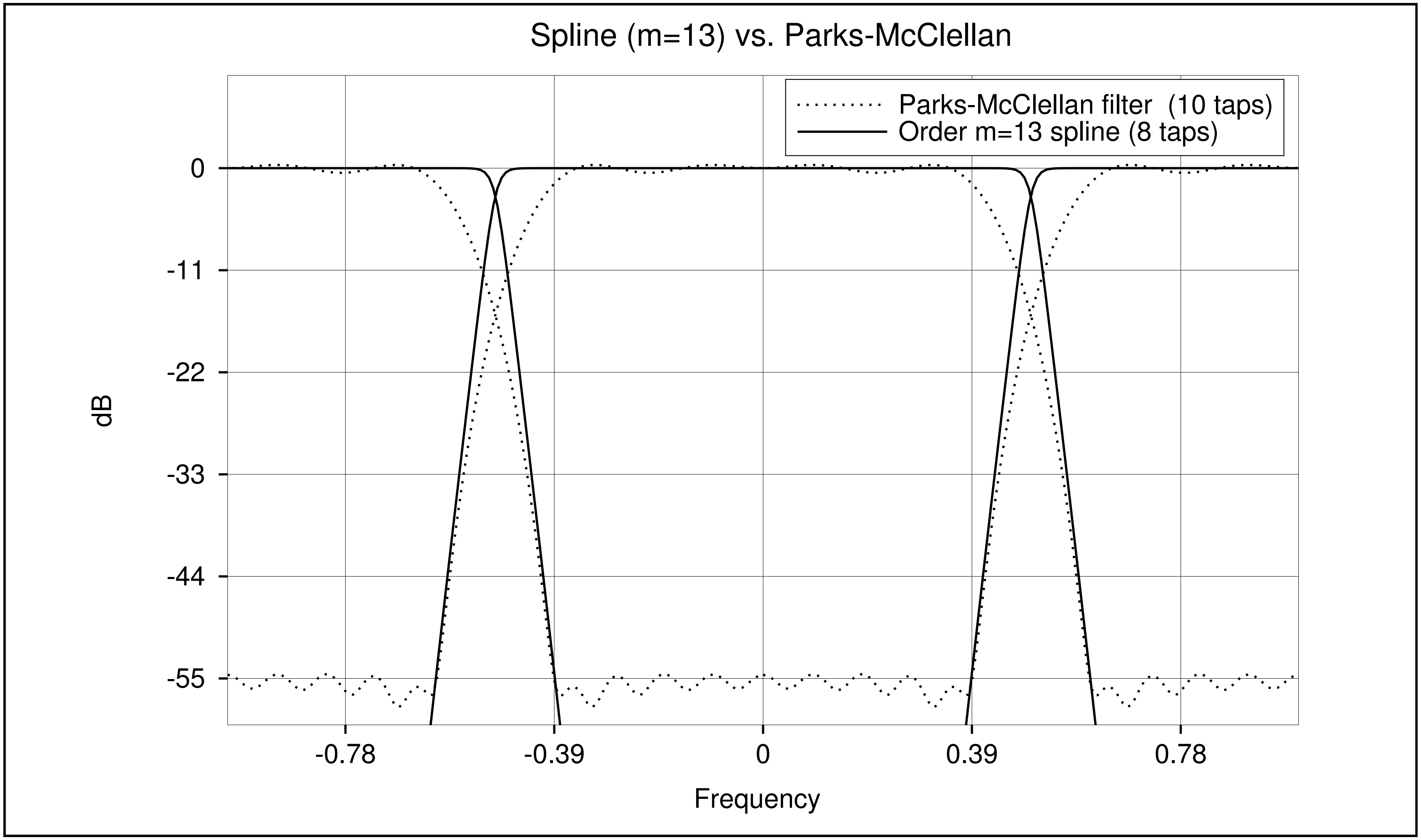
A similar algorithm for the electron microscope tomography (Sandberg, Mastronarde, Beylkin 2003)

Filter design



Comparison of Sinc filter with spline (USFFT) design (9 vs. 6 taps).

Filter design



Comparison Parks-McClellan filters with spline (USFFT) design (10 vs. 8 taps).

Inverse USFFT

Solve the linear system

$$g(x_k) = \sum_{l=-N/2}^{N/2} f_l \, e^{\pm 2\pi i l x_k},$$

or

$$g(x_k, y_k) = \sum_{l=-N_x/2}^{N_x/2} \sum_{l'=-N_y/2}^{N_y/2} f_{ll'} \, e^{\pm 2\pi i l x_k} \, e^{\pm 2\pi i l' y_k},$$

for f_l or $f_{ll'} \in \mathbb{C}$, where $k = 1, 2, \ldots, N_p$.

Points $x_k, y_k \in [-1/2, 1/2]$ and values $g(x_k)$, $g(x_k, y_k)$ are given.

Inverse USFFT

In order to solve

$$g_l = \sum_{k=-N/2}^{k=N/2} f_k \, e^{2\pi i k x_l}$$

where points x_l, $|x_l| < 1/2$, $l = 1, \dots, L$, $N < L$, are not necessarily equally spaced, consider

$$g(x) = \sum_{k=-N/2}^{k=N/2} f_k \, e^{2\pi i k x}.$$

We have

$$f_k = \frac{1}{2\pi} \int_0^{2\pi} g(x) e^{-2\pi i k x},$$

and view g_l as values of $g(x)$ at points x_l.

Let us find quadrature coefficients c_l at nodes x_l.

Inverse USFFT

We compute

$$w_m = \sum_{l=1}^{L} g_l c_l e^{-2\pi i m x_l},$$

where

$$w_m = \sum_{k=0}^{N-1} f_k T_{k-m} \text{ and } T_m = \sum_{l=1}^{L} c_l e^{2\pi i m x_l}.$$

From the point of view of linear algebra,

$$g = A f,$$

where $A_{kl} = e^{2\pi i k x_l}$.

We apply the diagonal matrix $D = \text{diag}\{c_l\}$ and then the adjoint matrix A^*,

$$w = A^* Dg = A^* D A f,$$

and then solve for f using Toeplitz structure of the matrix $T = A^* D A$.

Trigonometric interpolation of measured data

Consider function

$$g(x,y,z) = \sum_{l,l'} f_{ll'}\, e^{ilx}\, e^{il'y}\, e^{iz\sqrt{p^2-l^2-l'^2}}$$

that solves the Helmholtz equation,

$$(\nabla^2 + p^2)\, g = 0,$$

or

$$g(x,y,z) = \sum_{l,l'} f_{ll'}\, e^{\mathrm{i}lx}\, e^{\mathrm{i}l'y}\, e^{-z\sqrt{l^2+l'^2}}$$

that solves the Laplace equation,

$$\nabla^2 g = 0.$$

If we measure quantities which satisfy these equations, and measurements are on some unequally spaced grid, we have a setup for trigonometric (or harmonic) interpolation.

B.K. Alpert, M.H. Francis, and R.C. Wittmann, IEEE Trans., May 1998.