The Polynomial Resampling Method for Non-Uniform Fourier Data

Alex Gutierrez
Rodrigo Platte and Anne Gelb

Arizona State University

May 23, 2012
Goal: Reconstruct Piecewise Smooth Images from Non-uniformly Sampled Fourier Data

- MRI machines take data in the Fourier domain, we must convert to the physical domain.
- Jump discontinuities in the spatial domain
- Fourier information is global, hard to resolve highly localized features
- Non-uniform sampling patterns are becoming increasingly popular
Example Function
2 Interpolation Nodes

- $f(x)$
- PRM Reconstruction
- Reconstruction from Integer Samples

Sampled Nodes

$\hat{f}(\omega)$

$\hat{f}(\omega)_{PRM}$
5 Interpolation Nodes

![Graph showing f(x), PRM Reconstruction, and Reconstruction from Integer Samples]

- f(x)
- PRM Reconstruction
- Reconstruction from Integer Samples

![Graph showing Sampled Nodes and their Fourier Transforms]

- $\hat{f}(\omega)$
- $\hat{f}(\omega)_{PRM}$

Sampled Nodes
5 Interpolation Nodes, Zoom

![Graph showing interpolation nodes and samples]

- Red line: $f(x)$
- Blue dashed line: PRM Reconstruction
- Green dotted line: Reconstruction from Integer Samples

Sampled Nodes

- $\hat{f}(\omega)$
- $\hat{f}(\omega)_\text{PRM}$
Convergence at Integer Coefficients

\[ \| \hat{f}(\omega) - \hat{f}_{\text{Poly}}(\omega) \|_{\infty} \]

Number of Interpolation Nodes $N$
The Idea Behind the PRM

- Given a specific jump discontinuity \( \xi \) we integrate by parts

\[
\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} \, dx
\]

- Separating the integral:

\[
\int_{-\infty}^{\xi} f(x) e^{-i\omega x} \, dx + \int_{\xi}^{\infty} f(x) e^{-i\omega x} \, dx
\]

- Integrating by parts:

\[
\hat{f}(\omega) = (f(\xi^+) - f(\xi^-)) \frac{e^{-i\omega \xi}}{i\omega} + \frac{1}{i\omega} \left( \int_{-\infty}^{\xi} f'(x) e^{-i\omega x} \, dx + \int_{\xi}^{\infty} f'(x) e^{-i\omega x} \, dx \right)
\]
Integration by parts, cont.

- Repeating this process just for this new integral yields that

\[
\int_{-\infty}^{\xi} f'(x)e^{-i\omega x} \, dx + \int_{\xi}^{\infty} f'(x)e^{-i\omega x} \, dx =
\]

\[
(f'(\xi^+) - f'(\xi^-)) \frac{e^{-i\omega \xi}}{i\omega} + \frac{1}{i\omega} \left( \int_{-\infty}^{\xi} f''(x)e^{-i\omega x} \, dx + \int_{\xi}^{\infty} f''(x)e^{-i\omega x} \, dx \right)
\]

- Define \([f^{(k)}](x) := f^{(k)}(x^+) - f^{(k)}(x^-)\) and substitute into the first iteration to find that

\[
\hat{f}(\omega) = [f(\xi)] \frac{e^{-i\omega \xi}}{i\omega} + [f'(\xi)] \frac{e^{-i\omega \xi}}{(i\omega)^2} + \frac{1}{(i\omega)^2} \left( \int_{-\infty}^{\xi} f''(x)e^{-i\omega x} \, dx + \int_{\xi}^{\infty} f''(x)e^{-i\omega x} \, dx \right)
\]
Polynomial Interpolation

- After $M$ iterations we get

$$\hat{f}(\omega) = e^{-i\xi\omega} \sum_{k=0}^{M} \frac{[f^{(k)}](\xi)}{(i\omega)^{k+1}} + \epsilon_M(\omega)$$

- Let:

$$\lambda_k = [f^{(k)}](\xi)(-i)^{k+1}$$

- Using the change of variables $s(\omega) := \frac{1}{\omega}$ yields:

$$\hat{f}(\omega) = \hat{f} \left( \frac{1}{s} \right) \approx e^{-i\xi/s} \sum_{k=0}^{M} \lambda_k s^{k+1} = e^{-i\xi/s} s \sum_{k=0}^{M} \beta_k T_k(\tilde{s})$$
How Should We Sample In Fourier Space?

- Chebyshev Points work extremely well with Chebyshev Polynomials
- Inverse Chebyshev Points have the right kind of distribution

Linearly Spaced Points

Chebyshev Distribution (in $s(\omega) = \frac{1}{\omega}$)

Inverse Chebyshev Distribution (in $\omega$)
Advantages of the PRM

- Very cheap to increase the range of $\omega$
Difficulties with Multiple Jump Discontinuities

- Given two jump discontinuities at the points $\xi_1$ and $\xi_2$ the expansion looks like:

$$\hat{f}(\omega) \approx e^{-i\omega \xi_1} s \sum_{k=0}^{M} \beta_{k,1} T_k(\tilde{s}) + e^{-i\omega \xi_2} s \sum_{k=0}^{M} \beta_{k,2} T_k(\tilde{s})$$

$$\begin{bmatrix}
 e^{-i\omega \xi_1} s T_k(\tilde{s}_j) & e^{-i\omega \xi_2} s T_k(\tilde{s}_j)
\end{bmatrix}
\begin{bmatrix}
 \beta_{k,1} \\
 \beta_{k,2}
\end{bmatrix} = \hat{f}(\omega_j)$$
These two terms are not necessarily independent, which causes the linear system to be ill-conditioned or rank-deficient. To mitigate this we change to least squares.

This allows us to relax the requirement that we have Chebyshev points.
Least Squares

\[
\begin{bmatrix}
    e^{-i\omega_1 \xi} s T_k(\tilde{s}_j) \\
    e^{-i\omega_2 \xi} s T_k(\tilde{s}_j)
\end{bmatrix}
\begin{bmatrix}
    \beta_{k,1} \\
    \beta_{k,2}
\end{bmatrix}
= \hat{f}(\omega_j)
\]

- These two terms are not necessarily independent, which causes the linear system to be ill-conditioned or rank-deficient. To mitigate this we change to least squares.
- This allows us to relax the requirement that we have Chebyshev points
Test Function
Reconstruction of Function from 15 Inverse Chebyshev Modes in Fourier Space
Reconstruction of Function from 45 Inverse Chebyshev Modes in Fourier Space
Convergence As A Function Of Number of Modes

L2 Error vs. \( N \)

- \( 10^0 \) to \( 10^{-6} \)
- \( 10^0 \) to \( 10^{-8} \)
- \( N \) ranges from 0 to 100.
Pointwise Error vs. Number of Fourier Samples
What if we don’t know the edge locations exactly?
Optimizing the Residual to Find Edge Locations
Multivariate optimization
Comparison to Uniform Resampling, $N = 100$
Conclusions and Future Work

- Method uses a polynomial multiplied by complex exponentials to approximate the Fourier Transform function
- Requires good edge information
- Works particularly well with logarithmic sampling patterns
- Outperforms uniform resampling and convolutional gridding
- Robust with respect to both noise and distribution.
- Weakness: difficult to implement in two dimensions.
Acknowledgments

- NSF Computational Science for Undergraduates in Mathematics grant #0703587.
- NSF-FRG: Integrated Mathematical Methods in Medical Imaging grant #0652833
- Dr. Rodrigo Platte, Dr. Anne Gelb
- Dr. Aditya Viswanathan, Dr. Douglas Cochran
A brief discussion of other methods

- Convolutional gridding
  - Performed by convolving a smooth function with a "window" function and evaluating the result at equispaced points

![Graph showing convolutional gridding](image)

- Uniform resampling
  - Essentially solves a convolution with shifted sinc interpolants
### Table: Errors as a function of number of sampled modes and data noise level

<table>
<thead>
<tr>
<th>Noise Level</th>
<th>Modes</th>
<th>Initial Edge Estimate Error</th>
<th>Optimized Edge Error</th>
<th>$\frac{1}{N} \left| \tilde{f}(\mathbf{k}) - \hat{f}(\mathbf{k}) \right|_2^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-6}$</td>
<td>32</td>
<td>0.195052</td>
<td>0.00191624</td>
<td>$3.75196 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.0900712</td>
<td>0.000144707</td>
<td>$1.11344 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>0.0480141</td>
<td>$2.49349 \times 10^{-7}$</td>
<td>$4.77307 \times 10^{-10}$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0358682</td>
<td>$2.37945 \times 10^{-10}$</td>
<td>$1.94438 \times 10^{-13}$</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>32</td>
<td>0.195052</td>
<td>0.00191477</td>
<td>$3.75151 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.0900712</td>
<td>0.000146364</td>
<td>$1.12327 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>0.0480141</td>
<td>$8.48876 \times 10^{-6}$</td>
<td>$4.94615 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0358682</td>
<td>$1.13382 \times 10^{-5}$</td>
<td>$4.39712 \times 10^{-8}$</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>32</td>
<td>0.195052</td>
<td>0.00196296</td>
<td>$3.77897 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.0900712</td>
<td>0.000140487</td>
<td>$1.18384 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>0.0480141</td>
<td>$8.23667 \times 10^{-5}$</td>
<td>$6.31158 \times 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0358682</td>
<td>0.000159401</td>
<td>$5.09944 \times 10^{-7}$</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>32</td>
<td>0.195052</td>
<td>0.00240956</td>
<td>$4.32779 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.0900712</td>
<td>0.000476132</td>
<td>$7.3156 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>0.0480141</td>
<td>$0.00595786$</td>
<td>$5.59893 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0358682</td>
<td>0.0029249</td>
<td>$5.65196 \times 10^{-6}$</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>32</td>
<td>0.195052</td>
<td>0.00385144</td>
<td>$0.000131672$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.0900712</td>
<td>0.00420933</td>
<td>$7.64611 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>0.0480141</td>
<td>$0.00669356$</td>
<td>$5.55865 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0358682</td>
<td>0.0172991</td>
<td>$5.8717 \times 10^{-5}$</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>32</td>
<td>0.148765</td>
<td>0.050177</td>
<td>$0.00142465$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.0900712</td>
<td>0.053724</td>
<td>$0.000907596$</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>0.0595709</td>
<td>0.0476388</td>
<td>$0.0006451$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0358682</td>
<td>0.0922063</td>
<td>$0.000675603$</td>
</tr>
</tbody>
</table>