Set 16: Discrete Fourier Transform

Kyle A. Gallivan

Department of Mathematics

Florida State University

Foundations of Computational Math 2
Spring 2013

Trigonometric Interpolation and Approximation

Recall the trigonometric Fourier Polynomials and the series

- complex and periodic with period 2π
- $[a, b] = (0, 2\pi)$ and $\omega(x) = 1$
- $\phi_k(x) = e^{ikx}$ where $i = \sqrt{-1}$ for $k = 0, \pm 1, \pm 2, \ldots$
- $e^{ikx} = \cos kx + i \sin kx$
- $(f, g) = \int_0^{2\pi} f(x)\overline{g(x)}dx$
- orthogonality: $(\phi_n, \phi_m) = 0$ for $m \neq n$ and $(\phi_n, \phi_n) = 2\pi$

Fourier Approximation

For any $f(x) \in \mathcal{L}^2_\omega[0, 2\pi]$

$$f(x) = \sum_{k=-\infty}^{\infty} \gamma_k \phi_k(x)$$

$$\gamma_k = \int_0^{2\pi} f(x) e^{-ikx} dx = \frac{1}{2\pi}(f, \phi_k)$$

$$f(x) = \alpha(x) + i\beta(x)$$

$$\gamma_k = a_k + ib_k$$

$$a_k = \frac{1}{2\pi}\int_0^{2\pi} (\alpha(x)\cos(kx) + \beta(x)\sin(kx))dx$$

$$b_k = \frac{1}{2\pi}\int_0^{2\pi} (-\alpha(x)\sin(kx) + \beta(x)\cos(kx))dx$$

if $f(x)$ is real then $\gamma_{-k} = \overline{\gamma_k}$

Truncated Fourier Series

An approximation is achieved by truncation (n is assumed even)

$$f_n(x) = \sum_{\tilde{k}=-n/2}^{n/2} \gamma_{\tilde{k}} \phi_{\tilde{k}}(x)$$

$f_n(x)$ is the optimal least squares approximation on the finite dimensional subspace defined by the $\phi_{\tilde{k}}(x)$, for $-(n/2) \leq \tilde{k} \leq n/2$ by the earlier discussion of Hilbert spaces.

There is still the use of a symbolic, i.e., not computational, $\gamma_{\tilde{k}}$. We consider interpolation and quadrature to get a discrete version.

Fourier Interpolation

- Let $\theta = \frac{2\pi}{n}$ and consider angularly equally spaced points $x_j = j\theta, \;\; 0 \leq j \leq n.$
- $p_n(x)$ is a trigonometric interpolating polynomial such that

$$p_n(x) = \sum_{\tilde{k}=-n/2}^{n/2} \tilde{f}^I_{\tilde{k}} \phi_{\tilde{k}}(x)$$

$$p_n(x_j) = f(x_j)$$

Fourier Interpolation

We enforce the linear constraints to get the coefficients via $n+1$ equations:

$$f_n(x_j) = p_n(x_j) = \sum_{\tilde{k}=-n/2}^{n/2} \tilde{f}^I_{\tilde{k}} \phi_{\tilde{k}}(x_j), \;\; 0 \le j \le n$$

$$\tilde{M}\tilde{f}^I = f, \;\; \tilde{M} \in \mathbb{C}^{n+1 \times n+1}$$

$$\text{But } x_0 = 0, \;\; x_n = 2\pi \to \phi_{\tilde{k}}(x_0) = \phi_{\tilde{k}}(x_n) = e^{i\tilde{k}2\pi} = 1$$

$$\therefore e_1^T \tilde{M} = e_{n+1}^T \tilde{M} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix} \to \tilde{M} \text{ is singular}$$

Removing x_n and its equation from consideration results in removing the $n+1$-st row/column of $\tilde{M}$ to get n equations in n unknowns, $\tilde{f}^I_{\tilde{k}}, \;\; 0 \le j \le n-1, -n/2 \le \tilde{k} \le (n/2) - 1.$

Fourier Interpolation

Adjusting the index in the summation and basis functions yields

$$f_n(x_j) = \sum_{\tilde{k}=-n/2}^{n/2-1} \tilde{f}^I_{\tilde{k}} \phi_{\tilde{k}}(x_j) = \sum_{k=0}^{n-1} \tilde{f}^I_k \phi_k(x_j)$$

$$\phi_k(x) = e^{i(k-n/2)x}$$

$$\therefore M\tilde{f}^I = f,\ M \in \mathbb{C}^{n\times n},\ \ \tilde{f}^I \in \mathbb{C}^n,\ \ f \in \mathbb{C}^n$$

Fourier Interpolation

One final simplification is possible. The $-n/2$ shift can be removed since it is essentially a permutation of $\tilde{f}_k^I$ to $\hat{f}_k^I$.

The interpolation function is therefore defined by

$$\phi_k(x) = e^{ikx}, \quad \theta = \frac{2\pi}{n}, \quad x_j = j\theta$$

$$p_n(x) = \sum_{k=0}^{n-1} \hat{f}_k^I \phi_k(x)$$

$$p_n(x_j) = \sum_{k=0}^{n-1} \hat{f}_k^I \phi_k(x_j), \quad 0 \leq j \leq n-1$$

Fourier Interpolation

$$\theta = \frac{2\pi}{n}, \;\; \omega = e^{i\theta}, \;\; \phi_k(x_j) = e^{ikj\theta} = \omega^{kj}$$

$$\begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \end{pmatrix} = \begin{pmatrix} \phi_0(x_0) & \dots & \phi_{n-1}(x_0) \\ \phi_0(x_1) & \dots & \phi_{n-1}(x_1) \\ \vdots & & \vdots \\ \phi_0(x_{n-1}) & \dots & \phi_{n-1}(x_{n-1}) \end{pmatrix} \begin{pmatrix} \hat{f}^I_0 \\ \hat{f}^I_1 \\ \vdots \\ \hat{f}^I_{n-1} \end{pmatrix}$$

$$f = \Phi \hat{f}^I$$

Quadrature of Fourier Coefficients

- Apply a quadrature method to approximate (f, ϕ_k) to define $\tilde{f}_k^Q$.
- Composite Rectangle Rule (left endpoint)

$$\int_a^b g(z)dz \approx h \sum_{j=0}^{n-1} g(z_j) = h \sum_{j=0}^{n-1} g(z_0 + jh)$$

- Apply to inner product with $h = \theta = 2\pi/n$, $x_j = j\theta$

$$(f, \phi_k) = \int_0^{2\pi} f(x)\bar{\phi}_k(x)dx \approx (f, \phi_k)_n = \frac{2\pi}{n} \sum_{j=0}^{n-1} f(x_j)e^{-i\theta j(k-n/2)}$$

Quadrature of Fourier Coefficients

- Normalize to approximate Fourier coefficient

$$\frac{1}{2\pi}(f,\phi_k) \approx \frac{1}{2\pi}(f,\phi_k)_n = \tilde{f}_k^Q = \frac{1}{n}\sum_{j=0}^{n-1} f(x_j)e^{-i\theta j(k-n/2)}$$

- The permuted form, $\hat{f}_k^Q$, derived as before, satisfies

$$\frac{1}{2\pi}(f,\phi_k) \approx \frac{1}{2\pi}(f,\phi_k)_n = \hat{f}_k^Q = \frac{1}{n}\sum_{j=0}^{n-1} f(x_j)e^{-ikj\theta}$$

Quadrature of Fourier Coefficients

- The task is to show $\hat{f}_k^Q = \hat{f}_k^I$ used in the interpolation equations.
- This can be shown directly from definitions (see the textbook).
- We will show it via an investigation of the matrix defining the linear system for interpolation and the matrix that relates $f(x_j)$ to $\hat{f}_k^Q$ via the quadrature method.
- This will be done up to a convenient scaling via the discrete Fourier transform. (The superscript Q will be supressed to simplify notation.)
- The Fast Fourier transform will be derived via basic polynomial identities.

Discrete Fourier Transform

Reading:

- Computational Frameworks for the Fast Fourier Transform, C. Van Loan, SIAM
- The DFT: An Owner's Manual for the Disrete Fourier Transform, Briggs and Henson, SIAM
- Golub and VanLoan 96 Chapter 4

The discrete Fourier transform of a vector $x \in \mathbb{C}^n$ can be defined via the application of a matrix $F \in \mathbb{C}^{n \times n}$

$$\hat{f} = Ff$$

Discrete Fourier Transform

- The definition of the discrete Fourier Transform differs somewhat depending on the source. The differences center around scaling and the choice of the basic scalar that defines the operator.

- The quadrature method yields the definition

$$\hat{f}_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j e^{-ikj\theta}, \quad \theta = \frac{2\pi}{n}$$

- It will be shown that given Φ, the interpolation linear system matrix, $\hat{f} = \Phi^{-1} f$, Φ^{-1} is the DFT matrix and Φ is the inverse DFT matrix, where $e_k^T \Phi e_j = e^{ikj\theta} = \omega^{kj}$ and $e_k^T \Phi^{-1} e_j = \frac{1}{n} e^{-ikj\theta} = \overline{\omega}^{kj}$.

- In this form $\hat{f}_k$ contains a scaling by $1/n$.

Discrete Fourier Transform

It is also common to define the transform as

$$\hat{f}_k = \sum_{j=0}^{n-1} f_j e^{-ikj\theta}, \quad \theta = \frac{2\pi}{n}$$

$$\hat{f} = \mathcal{F} f \text{ with } e_k^T \mathcal{F} e_j = e^{-ikj\theta} = \mu^{kj}, \quad \mu = e^{-i\theta}$$

So we must show for this form that

$$\mathcal{F} = n\Phi^{-1} \text{ and } \frac{1}{n}\mathcal{F}^{-1} = \Phi$$

We will use a normalization that allows us to work with a unitary matrix, F, to define

$$\hat{f} = Ff$$

The Discrete Fourier Transform Matrix

- Let $\omega^k \in \mathbb{C}$, $k = 0, \ldots, n-1$ be the nth roots of unity where $\omega = e^{i2\pi/n}$ and $i = \sqrt{-1}$.
- $(\omega^k)^n - 1 = 0$ for $k = 0, \ldots, n-1$
- $\mu = e^{-i\theta} = \omega^{n-1}$

$$F^H = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{pmatrix}$$

Two-dimensional Fourier Transform

- Suppose a matrix A is used to represent an image
- Image processing often makes use of a two-dimensional transform.
- The two-dimensional transform of an

$$FAF^T$$

- FAe_k is the discrete Fourier transform of the k-th column of the image
- $e_j^T AF^T$ is the discrete Fourier transform of the j-th row of the image.
- Note the use of non-conjugate transpose.

Useful Properties

- The matrix F^H is symmetric, i.e., $(F^H)^T = F^H$.
- F is created simply by replacing all ω^k with $\overline{\omega^k} = \overline{\omega}^k$, i.e., $F^H = \overline{F}$.
- F is also symmetric, i.e., $F = F^T$.
- F is unitary and we have $F^{-1} = F^H$ from which the desired relationship to Φ and Φ^{-1} (with the appropriate scaling) hold and therefore $\hat{f}_k^I = \hat{f}_k^Q = \hat{f}_k$.
- The proof that F is unitary helps introduce some of the interactions of the basic properties of the DFT.

F is Unitary

Lemma. *Given* $n > 0$, *let*

- $\theta = 2\pi/n$
- $\omega = e^{i\theta}$,
- $P(\gamma) = \Sigma_{j=0}^{n-1} \gamma^j$.

Note that by definition $(\omega^k)^n - 1 = 0$ *for* $0 \leq k \leq n-1$.

If $0 < k \leq n-1$, *i.e.,* $\omega^k \neq 1$ *then* $P(\omega^k) = 0$.

F is Unitary

Proof.

$$\forall \rho \in \mathbb{C},\ P(\rho) - 1 = \Sigma_{j=1}^{n-1} \rho^j = \rho(\Sigma_{j=0}^{n-1} \rho^j) - \rho^n = \rho P(\rho) - \rho^n$$

$$\therefore \rho^n - 1 = \rho P(\rho) - P(\rho)$$

$$0 = (\omega^k)^n - 1 = \omega^k P(\omega^k) - P(\omega^k)$$

$$\therefore (\omega^k - 1) P(\omega^k) = 0$$

$$\omega^k \neq 1 \rightarrow P(\omega^k) = 0$$

□

Diagonal Elements

To see that $F^H F = I$ consider first the diagonal elements
$\alpha_{jj} = e_j^H F^H F e_j$

$$\alpha_{jj} = \frac{1}{n} \begin{pmatrix} (\omega^{j-1})^0 & (\omega^{j-1})^1 & \cdots & (\omega^{j-1})^{n-1} \end{pmatrix} \begin{pmatrix} \overline{(\omega^{j-1})}^0 \\ \overline{(\omega^{j-1})}^1 \\ \vdots \\ \overline{(\omega^{j-1})}^{n-1} \end{pmatrix}$$

$$= 1$$

Off-diagonal Elements

Let $\alpha_{kj} = e_k^H F^H F e_j$ with $k > j$

$$\alpha_{kj} = \frac{1}{n} \begin{pmatrix} (\omega^{k-1})^0 & (\omega^{k-1})^1 & \cdots & (\omega^{k-1})^{n-1} \end{pmatrix} \begin{pmatrix} \overline{(\omega^{j-1})}^0 \\ \overline{(\omega^{j-1})}^1 \\ \vdots \\ \overline{(\omega^{j-1})}^{n-1} \end{pmatrix}$$

$$= (\omega^{k-1})^0 \overline{(\omega^{j-1})}^0 + \cdots + (\omega^{k-1})^{n-1} \overline{(\omega^{j-1})}^{n-1}$$

$$= (\omega^{k-1} \overline{(\omega^{j-1})})^0 + \cdots + (\omega^{k-1} \overline{(\omega^{j-1})})^{n-1}$$

$$= (\omega^{k-j} \omega^{j-1} \overline{(\omega^{j-1})})^0 + \cdots + (\omega^{k-j} \omega^{j-1} \overline{(\omega^{j-1})})^{n-1}$$

$$= (\omega^{k-j})^0 + \cdots + (\omega^{k-j})^{n-1} = P(\omega^{k-j}) = 0$$

F and F^H

Lemma 16.1. *There exists a permutation matrix P such that $FP = F^H$ where*

$$P = \begin{pmatrix} e_1 & e_n & e_{n-1} & \cdots & e_2 \end{pmatrix}$$

Proof. To see this note that there is a simple relationship between a root of unity its conjugate, and its powers, $\omega^m = \omega^{m \bmod n}$ and $\overline{(\omega^k)} = \omega^{n-k}$ where $0 \le k \le n-1$. As a result the columns pair up as conjugate pairs, the jth column pairs with column $(-(j-1) \bmod n) + 1$. Note also that for any permutation matrix $P^{-1} = P^H = P^T$. $\square$

F and F^H

Let $n = 4$, $\omega^m = \omega^{m \bmod n}$, $\overline{(\omega^k)} = \omega^{n-k}$, and $\mu = e^{-i\theta} = \omega^{n-1}$.

$$\frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \mu & \mu^2 & \mu^3 \\ 1 & \mu^2 & \mu^4 & \mu^6 \\ 1 & \mu^3 & \mu^6 & \mu^9 \end{pmatrix} \stackrel{P}{\rightarrow} \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \mu^3 & \mu^2 & \mu \\ 1 & \mu^6 & \mu^4 & \mu^2 \\ 1 & \mu^9 & \mu^6 & \mu^3 \end{pmatrix}$$

F and F^H

Let $n = 4$, $\omega^m = \omega^{m \bmod n}$, $\overline{(\omega^k)} = \omega^{n-k}$, and $\mu = e^{-i\theta} = \omega^{n-1}$.

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \mu^3 & \mu^2 & \mu \\ 1 & \mu^6 & \mu^4 & \mu^2 \\ 1 & \mu^9 & \mu^6 & \mu^3 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega^{3n-3} & \omega^{2n-2} & \omega^{n-1} \\ 1 & \omega^{6n-6} & \omega^{4n-4} & \omega^{2n-2} \\ 1 & \omega^{9n-9} & \omega^{6n-6} & \omega^{3n-3} \end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega^9 & \omega^6 & \omega^3 \\ 1 & \omega^{18} & \omega^{12} & \omega^6 \\ 1 & \omega^{27} & \omega^{18} & \omega^9 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega^1 & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix}$$

Important Consequences

- The DFT is usually used to put a science or engineering problem into the "frequency" or "transform" domain.
- This often yields a simpler form of the problem or at least one that allows more intuitive thought (for the engineer or scientist)
- Essentially, it is a change of coordinates
- It can also be viewed as computing the coefficients of the vector x relative to the Fourier basis given by the columns of F.
- **Most importantly there is a Fast Fourier Transform.**
- The FFT can be derived in many ways.
- It requires $O(n \log n)$ computations rather than $O(n^2)$.

Complexity Advantage

We can be more specfic on the complexity (see Van Loan text)

- A length $n = 2^t$ has complexity of $5(n \log n)$ real arithmetic operations. (assuming the weights in all Ω_m are available)
- A conventional matrix vector approach requires $8n^2$ real arithmetic operations.
- Improvement in complexity $\sigma = 8n^2/(5n \log n)$

n	σ
32	≈ 10
1024	≈ 160
32678	≈ 3500
1048576	≈ 84000

Fast Fourier Transform

We have the following given $n > 0$

- $\theta_n = 2\pi/n$
- $\mu_n = e^{-i\theta_n}$
- $\mu_n^m = \mu_n^{m \bmod n}$
- $\mu_n^{n/2} = -1$ when n is even
- $\mu_n^n = 1$
- $\omega_n = e^{i\theta_n}$
- $\mu_n = \omega_n^{n-1}$
- $\mu_n = \mu_{2n}^2$

A Polynomial Viewpoint

The FFT can be derived using an observation on polynomials that is useful for $n = 2k$ and roots of unity.

$$p(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{n-2} x^{n-2} + f_{n-1} x^{n-1}$$

$$p_{even}(x) = f_0 + f_2 x + \cdots + f_{k-1} x^{k-1}$$

$$p_{odd}(x) = f_1 + f_3 x + \cdots + f_{2k-1} x^{2k-1}$$

$$p(x) = p_{even}(x^2) + x p_{odd}(x^2)$$

$$n = 4 \rightarrow p(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3$$

$$p_{even}(x) = f_0 + f_2 x, \;\; p_{odd}(x) = f_1 + f_3 x$$

$$p(x) = (f_0 + f_2 x^2) + x(f_1 + f_3 x^2) = f_0 + f_1 x + f_2 x^2 + f_3 x^3$$

A Polynomial Viewpoint

We have $\mu_{2k}^2 = \mu_k$ so the expression is useful when $n = 2^t$.

$$\mu_2^0 = 1, \;\; \mu_2^1 = -1, \;\; \mu_2^{2k} = 1, \;\; \mu_2^{2k+1} = -1$$

$$\mu_4^0 = 1, \;\; \mu_4^1 = -i, \;\; \mu_4^2 = -1, \;\; \mu_4^3 = i$$

$$\mu_4^2 = -\mu_4^0, \;\; \mu_4^3 = -\mu_4^1$$

A Polynomial Viewpoint

$$p(\mu_4^0) = p_{even}(\mu_4^{0*2}) + \mu_4^0 p_{odd}(\mu_4^{0*2}) = p_{even}(\mu_2^0) + \mu_4^0 p_{odd}(\mu_2^0)$$

$$= p_{even}(\mu_2^0) + 1 * p_{odd}(\mu_2^0)$$

$$p(\mu_4^1) = p_{even}(\mu_4^{1*2}) + \mu_4^1 p_{odd}(\mu_4^{1*2}) = p_{even}(\mu_2^1) + \mu_4^1 p_{odd}(\mu_2^1)$$

$$= p_{even}(\mu_2^1) + \mu_4 * p_{odd}(\mu_2^1)$$

$$p(\mu_4^2) = p_{even}(\mu_4^{2*2}) + \mu_4^2 p_{odd}(\mu_4^{2*2}) = p_{even}(\mu_2^2) + \mu_4^2 p_{odd}(\mu_2^2)$$

$$= p_{even}(\mu_2^2) - 1 * p_{odd}(\mu_2^2)$$

$$= p_{even}(\mu_2^0) - 1 * p_{odd}(\mu_2^0)$$

$$p(\mu_4^3) = p_{even}(\mu_4^{3*2}) + \mu_4^3 p_{odd}(\mu_4^{3*2}) = p_{even}(\mu_2^3) + \mu_4^3 p_{odd}(\mu_2^3)$$

$$= p_{even}(\mu_2^3) - \mu_4 * p_{odd}(\mu_2^3) = p_{even}(\mu_2^1) - \mu_4 * p_{odd}(\mu_2^1)$$

A Polynomial Viewpoint

$$p(\mu_4^0) = p_{even}(\mu_2^0) + 1 * p_{odd}(\mu_2^0)$$

$$p(\mu_4^1) = p_{even}(\mu_2^1) + \mu_4 * p_{odd}(\mu_2^1)$$

$$p(\mu_4^2) = p_{even}(\mu_2^0) - 1 * p_{odd}(\mu_2^0)$$

$$p(\mu_4^3) = p_{even}(\mu_2^1) - \mu_4 * p_{odd}(\mu_2^1)$$

$$\begin{pmatrix} p(\mu_4^0) \\ p(\mu_4^1) \\ p(\mu_4^2) \\ p(\mu_4^3) \end{pmatrix} = \begin{pmatrix} I_2 & \Omega_2 \\ I_2 & -\Omega_2 \end{pmatrix} \begin{pmatrix} p_{even}(\mu_2^0) \\ p_{even}(\mu_2^1) \\ p_{odd}(\mu_2^0) \\ p_{odd}(\mu_2^1) \end{pmatrix}, \quad \Omega_2 = \begin{pmatrix} 1 & 0 \\ 0 & \mu_4 \end{pmatrix}$$

Recursive Form

For $n = 8$

$$\begin{pmatrix} \hat{f}_T^{(8)} \\ \hat{f}_B^{(8)} \end{pmatrix} = \begin{pmatrix} I_4 & \Omega_4 \\ I_4 & -\Omega_4 \end{pmatrix} \begin{pmatrix} F_4 & 0 \\ 0 & F_4 \end{pmatrix} \begin{pmatrix} f_{odd}^{(8)} \\ f_{even}^{(8)} \end{pmatrix}$$

$$\Omega_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \mu_8 & 0 & 0 \\ 0 & 0 & \mu_8^2 & 0 \\ 0 & 0 & 0 & \mu_8^3 \end{pmatrix}$$

A Polynomial Viewpoint

- $n = 8$
- Two FFT's of length 4
- Two diagonal matrices times a vector of length 4
- Two vector additions of length 4
- Generally, $O(n)$ plus cost of two FFT's of length $n/2$
- $n = 2^t \rightarrow O(nt) = O(n \log n)$

DFT and Aliasing

- $f(x) \in \mathcal{L}^2_\omega[0, 2\pi] \rightarrow f(x) = \sum_{k=-\infty}^{\infty} \alpha_k \phi_k(x)$
- The range of α_k that are nonzero and/or have nontrivial magnitudes determines the frequency content of $f(x)$.
- For example, if $\alpha_k = 0$ for $k < -k_0$ and $k > k_1$ $f(x)$ is called band-limited.
- To use the DFT/FFT n must be chosen.
- This must be done with some knowledge of the frequency content of the class of $f(x)$ of interest.
- Aliasing makes one signal look identical to another in the frequency domain, i.e., the reconstructed signal is the same for two different input signals.

DFT and Aliasing

Consider the relationship between the discrete Fourier coefficients $\hat{\alpha}_k$, $0 \le k \le n-1$, and the continuous Fourier coefficients α_k, $-\infty \le k \le \infty$. Let $\theta = \frac{2\pi}{n}$ and $x_j = j\theta$.

$$f(x) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikx} \rightarrow f(x_j) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikj\theta}$$

$$\hat{\alpha}_m = \frac{1}{n} \sum_{j=0}^{n-1} f(x_j) e^{-imj\theta} = \frac{1}{n} \sum_{j=0}^{n-1} \Big[\sum_{k=-\infty}^{\infty} \alpha_k e^{ikj\theta} \Big] e^{-imj\theta}$$

$$= \sum_{k=-\infty}^{\infty} \alpha_k \Big[\frac{1}{n} \sum_{j=0}^{n-1} e^{ikj\theta} \Big] e^{-imj\theta} = \sum_{k=-\infty}^{\infty} \alpha_k \Big[\frac{1}{n} \sum_{j=0}^{n-1} \big\{ e^{i\theta(k-m)} \big\}^j \Big]$$

DFT and Aliasing

Let $\omega = e^{i\theta}$

$$\hat{\alpha}_m = \sum_{k=-\infty}^{\infty} \alpha_k \Big[\frac{1}{n} \sum_{j=0}^{n-1} \{e^{i\theta(k-m)}\}^j\Big] = \sum_{k=-\infty}^{\infty} \alpha_k \frac{1}{n} P(\omega^{k-m})$$

$$\text{where } P(\gamma) = \sum_{j=0}^{n-1} \gamma^j, \quad \gamma \in \mathbb{C}$$

We have

$$\begin{cases} P(\omega^{\ell}) = n & \text{if } \ell \mod n = 0 \\ \qquad\quad\; = 0 & \text{if } \ell \mod n \neq 0 \end{cases}$$

DFT and Aliasing

Therefore, for each $0 \le m \le n-1$ an α_k is present in the sum when $(k-m) \mod n = 0$, i.e., $k \mod n = m$.

This yields the simple identity

$$\hat{\alpha}_m = \alpha_m + \alpha_{m \pm n} + \alpha_{m \pm 2n} + \dots$$

DFT and Aliasing

Consider a simple example:

- $n > 0$ given
- $f(x) = e^{4inx}$, i.e., a single frequency
- All $\alpha_k = 0$ except $\alpha_{4n} = 1$.
- Consider the DFT coefficients $\hat{\alpha}_m$ for $0 \leq m \leq n-1$ and their IDFT $f_n(x)$.

DFT and Aliasing

We have for $f(x) = e^{4inx}$

$$\alpha_k = 0, \quad k = 0, \pm 1, \pm 2, \ldots \text{ and } k \neq 4n$$

$$\alpha_{4n} = 1$$

$$\therefore \hat{\alpha}_m = 0, \quad 1 \leq m \leq n-1$$

$$\hat{\alpha}_0 = \alpha_{4n} = 1$$

$$f_n(x) = \sum_{m=0}^{n-1} \hat{\alpha}_m e^{imx} = \hat{\alpha}_0 e^{i0x} = 1$$

$$f_n(x) \neq f(x)$$

Aliasing. Note $f(x_j) = 1$ for $0 \leq j \leq n-1$ so for the given n and its associated meshpoints $f(x)$ and 1 are indistinguishable.

Summary

So for the Fourier Transform we have

- Fast matrix-vector and matrix-matrix products Fv and FA
- Fast solutions to $Fv = b$ via $v = F^H b$
- Fast projections to the "frequency" domain and simple truncation for approximation. (Band pass filtering)
- A fast component to more complicated signal and image processing algorithms
- The DFT and related transforms such as the DCT (Chebyshev or Cosine) can be related to Gauss-Lobatto quadrature applied to evaluating the integrals in the Generalized Fourier series coefficients.