Scientific Computing, Utrecht, March 24, 2014

Fourier Transforms
Wavelets
Theory and Applications

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/∼sleij101/

Program

- Computing Fourier Coefficients
- Discrete Fourier Transform
- Discrete Cosine Transform
- Fast Fourier Transform
- Computing Fourier Integrals

Fourier coefficients

Let $f : \mathbb{R} \to \mathbb{C}$ be T-periodic and sufficiently smooth.

$$\gamma_k(f) = \frac{1}{T}\int_0^T f(t)\, e^{-2\pi i t \frac{k}{T}}\, \mathrm{d}t, \qquad f(t) = \sum_{k\in\mathbb{Z}} \gamma_k(f)\, e^{2\pi i t \frac{k}{T}}$$

Suppose f is sampled at t_n with $t_n \equiv n\Delta t$ and $\Delta t \equiv \frac{T}{N}$.

$$\widetilde{\gamma}_k \equiv \frac{\Delta t}{T}\sum_{n=0}^{N-1} f(t_n)\, e^{-2\pi i t_n \frac{k}{T}} = \frac{1}{N}\sum_{n=0}^{N-1} f_n e^{-2\pi i \frac{nk}{N}}$$

$$\widetilde{\gamma}_k = \widetilde{\gamma}_{k+jN} \qquad (k, j \in \mathbb{Z}).$$

$$f_n = \sum_{k\in\mathbb{Z}} \mu_k\, e^{2\pi i \frac{nk}{N}}, \quad \text{where} \quad \mu_k \equiv \sum_{j\in\mathbb{Z}} \gamma_{k+jN}(f).$$

Theorem. $\widetilde{\gamma}_k = \mu_k = \gamma_k(f) + \sum_{j\neq 0} \gamma_{k+jN}(f)$.

Proof. Apply next theorem.

Discrete Fourier Transform

Theorem. Let $(f_0, f_1, \ldots, f_{N-1})$ be a sequence of complex numbers. Define the sequence $(\widetilde{\gamma}_0, \ldots, \widetilde{\gamma}_{N-1})$ by

$$\widetilde{\gamma}_k \equiv \frac{1}{N}\sum_{n=0}^{N-1} f_n\, e^{-2\pi i \frac{nk}{N}} \qquad (k = 0, \ldots, N-1).$$

Then
$$f_n = \sum_{k=0}^{N-1} \widetilde{\gamma}_k\, e^{2\pi i \frac{kn}{N}} \qquad (n = 0, \ldots, N-1).$$

Note. Except for the minus-sign in the exponential and the scaling $\frac{1}{N}$ in the definition of the $\widetilde{\gamma}_k$, the formulae are the same. *Some text books scale both formulae with* $\frac{1}{\sqrt{N}}$.

The sequence $(\widetilde{\gamma}_k)$ is the **Discrete Fourier Transform** of the sequence (f_n). The theorem gives the inverse **DFT**.

Discrete Fourier Transform

Theorem. $\tilde{\gamma}_k \equiv \frac{1}{N}\sum_{n=0}^{N-1} f_n\, e^{-2\pi i \frac{nk}{N}} \;\Rightarrow\; f_n = \sum_{k=0}^{N-1} \tilde{\gamma}_k\, e^{2\pi i \frac{kn}{N}}.$

Proof. Let $\ell(N)$ be the space of sequences $\mathbf{f} \equiv (f_0, \ldots, f_{N-1})$ of N complex numbers with inner product

$$<\mathbf{f},\mathbf{g}> \equiv \frac{1}{N}\sum_{n=0}^{N-1} f_n\, \overline{g_n} \qquad (\mathbf{f},\mathbf{g} \in \ell(N)).$$

For each $k = 0, \ldots, N-1$, consider

$$\phi_k(n) \equiv e^{2\pi i \frac{kn}{N}} \qquad (n = 0, \ldots, N-1).$$

The collection of ϕ_k forms an orthonormal basis of $\ell(N)$.

In particular, $\mathbf{f} = \sum_{k=0}^{N-1} <\mathbf{f}, \phi_k> \phi_k$

The def. of the inner product reveals that $\tilde{\gamma}_k = <\mathbf{f}, \phi_k>$.

Discrete Cosine Transform

Similarly, if $\mathbf{f} \in \ell(N)$, then complex arithmetic is avoided and at the same time faster decreasing discrete Fourier coefficients $\tilde{\gamma}_k$ are obtained by extending $\mathbf{f}$ first to an even function before extending to a periodic function.

For ease of notation, we put γ_k instead of $\tilde{\gamma}_k$.

Discrete Cosine Transform

Example. Suppose $\mathbf{f} = (f_0, \ldots, f_N) \in \ell(N+1)$.

Extend $\mathbf{f}$ to an function that is even (around $n = N$):

$$\begin{aligned} \mathbf{g} &\equiv (f_0, f_1, \ldots, f_{N-1}, f_N, f_{N-1}, \ldots, f_2, \quad f_1) \\ &= (g_0, g_1, \ldots, g_{N-1}, g_N, g_{N+1}, \ldots, g_{2N-2}, g_{2N-1}) \end{aligned}$$

The **DFT** of $\mathbf{g}$ is

$$\gamma_k \;=\; \frac{1}{2N}[f_0 + (-1)^k f_N] + \frac{1}{N}\sum_{n=1}^{N-1} f_n \cos(2\pi \tfrac{kn}{N})$$

Note that, as $\mathbf{g}$, (γ_k) is even around $k = 0$ and $k = N$.
Therefore, the inverse **DFT**, for $n = 0, \ldots, N$, is

$$g_n = f_n = [\gamma_0 + (-1)^n \gamma_N] + 2\sum_{k=1}^{N-1} \gamma_k \cos(2\pi \tfrac{kn}{N})$$

Discrete Cosine Transform

There are a number of ways to extend a finite sequence to a sequence of length $2N$ that is even.

Example. Suppose $\mathbf{f} = (f_0, \ldots, f_{N-1}) \in \ell(N)$.
Then the extension

$$\mathbf{g} \equiv (\mathbf{f}, \mathbf{f}^{\top}) \quad \text{with} \quad \mathbf{f}^{\top} \equiv (f_{N-1}, f_{N-2}, \ldots, f_1, f_0)$$

leads to an $2N$-periodic function $\mathbf{g}$ that is even around $n = -\frac{1}{2}$ and $n = N - \frac{1}{2}$.
This leads to the so-called **DCT-II** transform:

DCT-II. With $\phi_{n,k} \equiv \cos\left(\pi(n + \frac{1}{2})\frac{k}{N}\right)$,

$$\gamma_k = \frac{1}{N}\sum_{n=0}^{N-1} f_n\, \phi_{n,k}, \qquad f_n = \gamma_0 + 2\sum_{k=1}^{N-1} \gamma_k\, \phi_{n,k}$$

Discrete Cosine Transform

There are a number of ways to extend a finite sequence to a sequence of length $2N$ that is even. The first extension that we considered (even around 0 and N) is called **DCT-I**, the second (even around $-\frac{1}{2}$, $N-\frac{1}{2}$) is **DCT-II**. The **DCT-II** seems to be the most popular one in practice and is often simple called **the DCT**.

Odd extensions lead to sines rather than cosines. However, sinus are cosines up to some phase shift and with some simple manipulation, odd extensions also lead to transforms involving cosines only, to the so called **DCT-III** and **DCT-IV**. **DCT-IV** is the standard **DCT** in Matlab:

DCT-IV. With $\qquad \phi_{n,k} \equiv \cos\left(\frac{\pi}{N}(n+\frac{1}{2})(k+\frac{1}{2})\right),$

$$\gamma_k = \frac{1}{N}\sum_{n=0}^{N-1} f_n\,\phi_{n,k}, \qquad f_n = 2\sum_{k=0}^{N-1}\gamma_k\,\phi_{n,k}.$$

Applications of DCT

• **Image compression.**
Goal. Compression.
2-dimensional (and 3-d) **DCT-II** is used with N low.

JPEG, MJPEG, MPEG use **DCT-II** on 8×8 blocks

• **Audio compression.**
Goal. Compression and **spectral information**: the techniques in audio compression exploit psygological facts on how we hear combinations of harmonic oscillations, that is, compression depends on the distribution of frequencies.

A related transform, **Modified DCT**, is used in AAC, Vorbis, MP3.

• **Partial Differential Equations**. **DCT**s are used for solving PDEs, where the variants of **DCT** correspond to (slightly) different boundary conditions.

Fast Fourier Transform

Suppose the sequence $\gamma \equiv (\gamma_0,\ldots,\gamma_{N-1}) \in \ell(N)$ is available. The naive way of computing the **DFT**

$$\mathcal{F}(\gamma)_n \equiv f_n = \sum_{k=0}^{N-1}\gamma_k\,e^{2\pi i\frac{kn}{N}} \qquad (n=0,\ldots,N-1).$$

requires more than $2N^2$ **floating point operations** (additions, multiplications): for each of the N ns, $2N$ **flop**.

In practice N is huge.
N of the order of $10^6 \sim 10^8$ is not exceptional.

Gauss [first half of the 19th century], Runge [1903] and Cooley & Tukey [1965] in the most cited mathematical paper ever, proposed a computational scheme, **FFT**, *that reduces the computational costs to $2N\log_2(N)$* **flop**.
For, e.g., $n = 2^{20} \approx 10^6$, this makes a difference with $2N^2$ of 1 sec versus 3:30 hours.

Fast Fourier Transform

Suppose $N = 2^\ell$ for some $\ell \in \mathbb{N}$. Put $M = 2^{\ell-1} = \frac{1}{2}N$.

$$f_n = \sum_{k=0}^{N-1}\gamma_k\,e^{2\pi i\frac{kn}{N}} \qquad (n=0,\ldots,N-1).$$

With $\; f_{e,n} \equiv \sum_{2k<N}\gamma_{2k}\,e^{2\pi i\frac{kn}{M}}, \quad f_{o,n} \equiv \sum_{2k+1<N}\gamma_{2k+1}\,e^{2\pi i\frac{kn}{M}},$

we have
$$\begin{aligned} f_n &= f_{e,n} + f_{o,n}\,e^{\pi i\frac{n}{M}} \quad (n=0,\ldots,M-1),\\ f_{n+M} &= f_{e,n} - f_{o,n}\,e^{\pi i\frac{n}{M}} \quad (n=0,\ldots,M-1).\end{aligned}$$

Let κ_ℓ be the number of **flop** required to compute the **DFT** of length $M = 2^\ell$. Then, the above implies that

$$\kappa_\ell = 2\kappa_{\ell-1} + 1.5\,N.$$

We need N additions (subtractions), M multiplications;
For now, we neglected the costs for computing $e^{\pi i\frac{n}{M}}$.
We can repeat the partitioning trick to $f_{e,n}$ and $f_{o,n}$.

Fast Fourier Transform

Suppose $N=2^\ell$ for some $\ell\in\mathbb{N}$. Put $M=2^{\ell-1}=\frac{1}{2}N$.

$$f_n=\sum_{k=0}^{N-1}\gamma_k\,e^{2\pi i\frac{kn}{N}}\qquad(n=0,\ldots,N-1).$$

With $f_{e,n}\equiv\sum_{2k<N}\gamma_{2k}\,e^{2\pi i\frac{kn}{M}}$, $\quad f_{o,n}\equiv\sum_{2k+1<N}\gamma_{2k+1}\,e^{2\pi i\frac{kn}{M}}$,

we have
$$\begin{aligned} f_n &= f_{e,n}+f_{o,n}\,e^{\pi i\frac{n}{M}} \quad (n=0,\ldots,M-1),\\ f_{n+M} &= f_{e,n}-f_{o,n}\,e^{\pi i\frac{n}{M}} \quad (n=0,\ldots,M-1).\end{aligned}$$

Repeating this partitioning trick recursively down to level $\ell=0$ is Fast Fourier Transform.

Theorem. **FFT** requires $(1.5\,\ell+0.5)N$ **flop**.

Proof. The $0.5\,N$ comes from the computation of $e^{\pi i\frac{n}{M}}$, which can be computed as $\zeta^n=\zeta^{n-1}\zeta$ with $\zeta=e^{\pi i\frac{1}{M}}$. Note that, $e^{\pi i n 2^{-\ell+j}}=e^{\pi i(2^j n)2^{-\ell}}$.

DFT & FFT as matrix multiplication

Let $\mathbf{f}=(f_0,\ldots,f_{N-1})^T$ and $\gamma=(\gamma_0,\ldots,\gamma_{N-1})^T$ be such that $\quad f_n=\sum_{k=0}^{N-1}\gamma_k\,e^{2\pi i\frac{kn}{N}}\qquad(n=0,\ldots,N-1).$

Let $\mathbf{F}$ be the $N\times N$ matrix with (n,k)-entry $e^{2\pi i\frac{kn}{N}}$. Then

$$\mathbf{f}=\mathbf{F}\gamma$$

FFT. Now suppose $N=2^\ell$. Put $M\equiv2^{\ell-1}$.
Let $\mathbf{F}_\ell\equiv\mathbf{F}$ be the **DFT** for level ℓ, i.e., for $N=2^\ell$.
The first step in **FFT** can be written as

$$\mathbf{f}=\begin{bmatrix}\mathbf{I}_{\ell-1} & +\mathbf{D}_{\ell-1}\\ \mathbf{I}_{\ell-1} & -\mathbf{D}_{\ell-1}\end{bmatrix}\begin{bmatrix}\mathbf{f}_e\\ \mathbf{f}_o\end{bmatrix}=\begin{bmatrix}\mathbf{I}_{\ell-1} & +\mathbf{D}_{\ell-1}\\ \mathbf{I}_{\ell-1} & -\mathbf{D}_{\ell-1}\end{bmatrix}\begin{bmatrix}\mathbf{F}_{\ell-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_{\ell-1}\end{bmatrix}\begin{bmatrix}\gamma_e\\ \gamma_o\end{bmatrix}$$

where $\gamma_e=(\gamma_0,\gamma_2,\ldots)^T$, $\gamma_o=(\gamma_1,\gamma_3,\ldots)^T$.

FFT for sequences of any length?

Suppose $N=q^\ell$ for some $q\in\mathbb{N},q>2$.

Then we can design a **FFT** algorithm similar to the one for $q=2$. For instance, if $q=3$, and $(\gamma_0,\ldots,\gamma_{N-1})$ is a sequence of length N, then we can group the coefficients in three classes (γ_{3k}), (γ_{3k+1}) and (γ_{3k+2}) instead of the two as for $q=2$ (the one with even indices and one with odd indices) and we can decompose the f_n accordingly.

q is the radix of the **FFT**.

The computational costs are in the order of $N\log_3 N$:
Comp. Costs $\approx\mathcal{C}_q\,N\log_3 N$ for some $\mathcal{C}_q>0$.

Property. **FFT** with radix 4 allows the most efficient implementation (i.e., $4=\mathrm{argmin}_q\,\mathcal{C}_q\,N\log_q N$).

FFT for sequences of any length?

We can factorise any $N\in\mathbb{N}$, that is, we can decompose any N into a product of prime factors and we can design a **FFT** for sequences of length N that is a mixture of **FFT**s of radix p_j with p_j the primes that occur in the factors.

However, computationally, this approach is not attractive:

- we have to factorise N
- coding of such a **FFT** with a mixture of **FFT**s with different radixes is messy
- if the primes are large (with the extremal situation where N itself is prime), then the **FFT** is not faster.

FFT for sequences of any length?

If, for instance, we have to compute Fourier coefficients of a T-periodic function $f : \mathbb{R} \to \mathbb{C}$, then we can select the sample frequency $1/\Delta t$ as we like (with the only restriction that it is sufficiently large), for instance,

$$\Delta t = T/N \quad \text{with} \quad N = 2^\ell.$$

Conclusion.
Some application allow to select N to be a power of 2.

FFT for sequences of any length?

Some application allow sequences $(\gamma_0, \ldots, \gamma_{M-1})$ of length M to be extended to sequences of length 2^ℓ (with ℓ such that $2^{\ell-1} < M \leq 2^\ell$) by appending with zeros.

Example. The **convolution product** $\alpha \star \beta$ of the sequence $\alpha = (\gamma_0, \ldots, \alpha_{M-1})$ and $\beta = (\beta_0, \ldots, \beta_{M-1})$ is defined by

$$(\alpha \star \beta)_k \equiv \sum_j \alpha_j \, \beta_{k-j} \qquad (k = 0, \ldots, 2M-2).$$

Assume that the length of α and β is $M = 2^\ell$. Append α and β with zeros to sequences of length $N \equiv 2^{\ell+1}$. Next, extend α and β periodically (period N) and define $\star_N$:

$$(\alpha \star_N \beta)_k \equiv \sum_{j=0}^{N-1} \alpha_j \, \beta_{k-j} \qquad (k = 0, \ldots, N-1).$$

Note that the definitions of $\alpha \star \beta$ are consistent (lead to the same values for $k < N$).

Discrete Convolution Products

Definition. For $\alpha, \beta \in \ell(N)$, let

$$(\alpha \star_N \beta)_k \equiv \sum_{j=0}^{N-1} \alpha_j \, \beta_{k-j} \quad (k = 0, \ldots, N-1),$$

where $\beta_{k-j} \equiv \beta_{N+k-j}$ if $k - j < 0$ (periodic extension).

Theorem. $\mathcal{F}_N(\alpha *_N \beta) = \mathcal{F}_N(\alpha) \cdot \mathcal{F}_N(\beta)$, where
the $\cdot$-product is coordinate wise (the **Hadamard product**).

Suppose $2^{\ell-1} < N < 2^\ell$. Put $L \equiv 2\,2^\ell$.
Form $\widetilde{\beta} \equiv (\beta, \mathbf{0}, \beta)$ to a sequence of length L.
Form $\widetilde{\alpha} \equiv (\alpha, \mathbf{0}, \mathbf{0})$ to a sequence of length L.

Property. $(\alpha \star_N \beta)_k = (\widetilde{\alpha} \star_L \widetilde{\beta})_k$ for $k = 0, \ldots, N-1$.

Corollary. $(\alpha *_N \beta)_k = (\mathcal{F}_L^{-1}[\mathcal{F}_L(\widetilde{\alpha}) \cdot \mathcal{F}_L(\widetilde{\beta})])_k \qquad (k < N).$

$\alpha \star_L \beta$ can be computed with three **DFT** of radix 2 plus L mult.. Costs: $\leq 24N(\ell+2)$ **flop** rather than $0.5\,N^2$.

FFT for sequences of any length?

Some application allow sequences $(\gamma_0, \ldots, \gamma_{M-1})$ of length M to be extended to sequences of length 2^ℓ (with ℓ such that $2^{\ell-1} < M \leq 2^\ell$) by appending with zeros.

Example. The **convolution product** $\alpha \star \beta$ of the sequence $\alpha = (\gamma_0, \ldots, \alpha_{M-1})$ and $\beta = (\beta_0, \ldots, \beta_{M-1})$ is defined by

$$(\alpha \star \beta)_k \equiv \sum_j \alpha_j \, \beta_{k-j} \qquad (k = 0, \ldots, 2M-2).$$

Conclusion. In these applications the FFT is nothing more than an efficient computational tool. The quantities to be computed are in same domain as the inputs (time-domain rather than in frequency domain).

Appending with zeros

Consider $\gamma = (\gamma_0, \ldots, \gamma_{M-1}) \in \ell(M)$ with $2^{\ell-1} < M < N \equiv 2^\ell$.
Append γ with zeros to a sequence γ^+ of length N:

$$\gamma^+ \equiv (\gamma_0, \ldots, \gamma_{M-1}, 0, \ldots, 0).$$

Observe that $\quad \mathcal{F}_M(\gamma) \neq \mathcal{F}_N(\gamma^+), \quad$ because

$$\sum_{k=0}^{M-1} \gamma_k\, e^{2\pi i \frac{kn}{M}} = \sum_{k=0}^{N-1} \gamma_k^+\, e^{2\pi i \frac{kn}{M}} \neq \sum_{k=0}^{N-1} \gamma_k^+\, e^{2\pi i \frac{kn}{N}}.$$

Conclusion. If the quantities of interest are in the 'dual' domain (frequency rather than time, or time rather than frequency), then appending zeros is not allowed.

FFT for sequences of any length?

Consider $\gamma = (\gamma_0, \ldots, \gamma_{M-1}) \in \ell(M)$ with $2^{\ell-1} < M < N \equiv 2^\ell$.

Property. With $\beta_k \equiv e^{-\pi i \frac{k^2}{M}}$, we have that

$$\mathcal{F}_M(\gamma) = \mu\, \overline{\beta} \quad \text{with} \quad \mu \equiv (\gamma\, \overline{\beta}) \star_M \beta$$

As we saw before, the convolution product can be computed with three **DFT** of radix 2 (and length $L \equiv 2N$), plus L multiplications. The multiplications $\gamma\overline{\beta}$ and $\mu\overline{\beta}$ require an additional $2M$ multiplications.

Computing Fourier integrals

f sampled at $t_n = t_0 + n\,\Delta t$. $1/\Delta t$ *sample frequency*.
For ease of notation, take $t_0 = 0$ (otherwise shift by t_0).

$$\widehat{f}(\omega) \approx \int_{t_0}^{t_0+T} f(t)\, e^{-2\pi i t \omega}\, \mathrm{d}t \approx \Delta t \sum_{n=0}^{N-1} f_n\, e^{-2\pi i\, n\, \Delta t\, \omega}$$

Here, $T = N\Delta t$ and $f_n = f(t_n)$.
Of interest for $\omega = \frac{k}{T}$ $(k = 0, \ldots, N-1)$.

$\widehat{f}(\omega)$ to be computed by **DFT**.

Two 'discretizations'! How accurate is this?

Windowing

$$\widehat{f}(\omega) \approx \int_{t_0}^{t_0+T} f(t)\, e^{-2\pi i t \omega}\, \mathrm{d}t$$

Actually, we are computing the Fourier transform of

$f W_{t_0}$, where $W(t) = 1$ if $1 \leq t \leq T$, and $W(t) = 0$ elsewhere

and $W_{t_0}(t) \equiv W(t - t_0)$.

W is a **time-window**.
Of interest: the difference between $\widehat{f}(\omega)$ and $(\widehat{f W_{t_0}})(\omega)$.

$\Phi(t, \omega) \equiv (\widehat{f W_t})(\omega)$ is called a **spectogram** of f.

Discretization

$$F(\omega) \equiv \Delta t \sum_{n=-\infty}^{\infty} f_n \, e^{-2\pi i n \Delta t \omega}$$

Relation $\widehat{f}(\omega)$ and $F(\omega)$? Does this depend on ω?

$B_f \equiv \{\omega \in \mathbb{R} \mid |\widehat{f}(\omega)| \neq 0\}$ is the **frequency band** of f.
f is of **bounded bandwidth** if $B_f \subset [-\Omega, +\Omega]$
for some $\Omega > 0$: smallest Ω is the **bandwidth**.

Suppose f is of bandwidth $\leq \Omega$.

$$f(t) = \sum_{k=-\infty}^{\infty} \gamma_k \, e^{2\pi i \frac{t}{T} k} \quad \Leftrightarrow \quad \gamma_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \, e^{-2\pi i \frac{t}{T} k} \, \mathrm{d}t$$

Take $\Delta t = \frac{1}{2\Omega}$, change $-t \leftrightarrow \omega$, $T \leftrightarrow 2\Omega$, $n \leftrightarrow k$ ...

$$F(\omega) \equiv \Delta t \sum_{n=-\infty}^{\infty} f_n \, e^{-2\pi i n \Delta t \omega} \quad \Leftrightarrow \quad f_n = \int_{-\Omega}^{\Omega} F(\omega) \, e^{2\pi i \frac{\omega}{2\Omega} n} \, \mathrm{d}\omega$$

f of bandwidth $\leq \Omega \;\Rightarrow$

$$f(t) = \int_{-\infty}^{\infty} \widehat{f}(\omega) \, e^{2\pi i t \omega} \, \mathrm{d}\omega = \int_{-\Omega}^{\Omega} \widehat{f}(\omega) \, e^{2\pi i t \omega} \, \mathrm{d}\omega$$

In particular, $\quad f_n = f(t_n) = \int_{-\Omega}^{\Omega} \widehat{f}(\omega) \, e^{2\pi i \frac{n\omega}{2\Omega}} \, \mathrm{d}\omega.$

$$f(t) = \sum_{k=-\infty}^{\infty} \gamma_k \, e^{2\pi i \frac{t}{T} k} \quad \Leftrightarrow \quad \gamma_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \, e^{-2\pi i \frac{t}{T} k} \, \mathrm{d}t$$

Take $\Delta t = \frac{1}{2\Omega}$, change $-t \leftrightarrow \omega$, $T \leftrightarrow 2\Omega$, $n \leftrightarrow k$ ...

$$F(\omega) \equiv \Delta t \sum_{n=-\infty}^{\infty} f_n \, e^{-2\pi i n \Delta t \omega} \quad \Leftrightarrow \quad f_n = \int_{-\Omega}^{\Omega} F(\omega) \, e^{2\pi i \frac{\omega}{2\Omega} n} \, \mathrm{d}\omega$$

f of bandwidth $\leq \Omega \;\Rightarrow$

$$\int_{-\Omega}^{\Omega} [\widehat{f}(\omega) - F(\omega)] e^{2\pi i \frac{n\omega}{2\Omega}} \, \mathrm{d}\omega = 0 \qquad \forall n \in \mathbb{Z}.$$

$$f(t) = \sum_{k=-\infty}^{\infty} \gamma_k \, e^{2\pi i \frac{t}{T} k} \quad \Leftrightarrow \quad \gamma_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \, e^{-2\pi i \frac{t}{T} k} \, \mathrm{d}t$$

Take $\Delta t = \frac{1}{2\Omega}$, change $-t \leftrightarrow \omega$, $T \leftrightarrow 2\Omega$, $n \leftrightarrow k$ ...

$$F(\omega) \equiv \Delta t \sum_{n=-\infty}^{\infty} f_n \, e^{-2\pi i n \Delta t \omega} \quad \Leftrightarrow \quad f_n = \int_{-\Omega}^{\Omega} F(\omega) \, e^{2\pi i \frac{\omega}{2\Omega} n} \, \mathrm{d}\omega$$

f of bandwidth $\leq \Omega \;\Rightarrow$

$$\widehat{f}(\omega) = \Delta t \sum_{n=-\infty}^{\infty} f_n \, e^{-2\pi i n \Delta t \omega} \qquad \forall \omega \in [-\Omega, +\Omega].$$

$\Delta t = \frac{1}{2\Omega}$ is the **Nyquist rate**.

Theorem.
f of bandwidth $\leq \Omega$ & sample frequency $1/\Delta t \geq 2\Omega \;\Rightarrow$

$$\widehat{f}(\omega) = \Delta t \sum_{n=-\infty}^{\infty} f_n \, e^{-2\pi i\, n\, \Delta t\, \omega} \qquad \forall \omega \in [-\Omega, +\Omega]$$

The discretization is exact if the bandwidth $\leq \Omega$ and the sample frequency $\geq 2\Omega$ $(\Delta t \leq 1/(2\Omega))$. Fourier transform of this result leads to

The Shannon–Whittakker Theorem.
f of bandwidth $\leq \Omega$ & sample frequency $1/\Delta t \geq 2\Omega \;\Rightarrow$

$$f(t) = \sum_{n=-\infty}^{\infty} f_n \,\mathrm{sinc}\left(\frac{t - t_n}{\Delta t}\right) \qquad \forall t \in \mathbb{R}.$$

Discussion. The Shannon–Whittakker theorem tells us that f can be reconstructed from its sample values, if f is of bounded bandwidth and the sample frequency is at least twice the maximal frequency of f. However, reconstruction requires values f_n from the (far) future as well as from the (far) past.

Application. Resampling (sampling at another sampling rate) is possible.

If the new sample rate is $\frac{p}{q}$ times the old sample rate Δt, then, in practice, resampling is achieved by

1) upsampling by p
2) filtering to get rid of frequencies $> \Omega$
3) downsampling by q.

(Details later)

Conclusions

- Discretization is fine provided f is of bounded bandwidth and the sample frequency is high enough.

- Perturbations by windowing can not be avoided. Effects include smearing and leakage. Effects can be diminished by a larger time-window. One effect can be diminished at the cost of others (by other time-windows).