SECTION 10
Fourier Transforms

Outline of Section

- Fourier Transforms – recap
- Discrete FTs (DFTs)
- Sampling & Aliasing
- Spectral methods – Solving the Poisson Equation as an example
- Fast Fourier Transform (FFT) algorithm (non-examinable)

This chapter focuses on how to numerically calculate the Fourier transform of regular, discrete, samples of a function. The **Discrete Fourier Transform** (DFT) is used for this, and is the discrete equivalent of the Fourier transform. The sampled function could be the function found by solving an ODE or PDE using finite difference methods. Or it might be data sampled from, e.g., an oscilloscope or a microphone. Considerations/limitations due to the sampling will be discussed. How to use DFTs to solve differential equations - spectral methods – will be touched on.

The **Fast Fourier Transform** (FFT) – an efficient implementation on the DFT – is used extensively, e.g.,

- Noise suppression – Data analysis
- Image compression – JPEGS
- Music compression – MPEGS
- Medical imaging
- Interferometric imaging
- Modelling of optical systems
- Solution of periodic boundary value problems

to name a few examples.

10.1. Fourier Transforms – Recap

The continuous Fourier Transform (FT), both forward and backward, of a function $f(t)$ are defined as

$$\tilde{f}(\omega) = \int_{-\infty}^{+\infty} e^{i\,\omega\,t} f(t)dt = \mathcal{F}(f(t)) \ , \tag{10.1}$$

$$\text{and} \qquad f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-i\,\omega\,t}\tilde{f}(\omega)d\omega = \mathcal{F}^{-1}(\tilde{f}(\omega)) \ , \tag{10.2}$$

for the case of time t and angular frequency $\omega = 2\pi\nu$ (where ν is frequency). Time t and ω are reciprocal 'coordinates' or variables. $\tilde{f}(\omega)$ is the (angular) frequency spectrum of the function $f(t)$. The FT is an expansion on plane waves $\exp(-i\,\omega\,t)$ which constitute an orthonormal basis, and $\tilde{f}(\omega)$ can be thought of as the "coefficients" of each component wave of angular frequency ω. (Notation note: don't confuse with the tilde '$\sim$' notation used earlier in the course to denote numerical approximation!) In general $\tilde{f} \in \mathbb{C}$, even for a real function $f(t)$, with $|\tilde{f}(\omega)|$ being the amplitude and $\arg(\tilde{f}(\omega))$ the phase of the component wave $\exp(-i\,\omega\,t)$. For $f(t) \in \mathbb{R}$ the **reality condition** applies in reciprocal space (also referred to as 'Fourier space')

$$\tilde{f}(-\omega) = \tilde{f}^{*}(\omega) \ , \tag{10.3}$$

so that the negative frequencies are degenerate; there is no extra information in them. However, for an arbitrary $f(t) \in \mathbb{C}$, $\tilde{f}(-\omega)$ is unrelated to $\tilde{f}^{*}(\omega)$.

The "FT pairs" $f(t)$ and $\tilde{f}(\omega)$ are different representations of the same thing in the time and frequency domain, respectively, and the relationship between such FT pairs is often written as

$$f(t) \rightleftharpoons \tilde{f}(\omega) \ . \tag{10.4}$$

$\mathcal{F}^{-1}(\tilde{f}(\omega))$ is often called the inverse Fourier transform. Note that there are different conventions for defining the FT operations, e.g., which operation gets the $1/2\pi$ factor or whether both share it (i.e. $1/\sqrt{2\pi}$ in front of each), and the sign in the transform kernel (i.e. $\exp(-i\,\omega\,t)$ or $\exp(i\,\omega\,t)$) , etc.

For spatial FTs in one dimension (e.g. x) the reciprocal variables become $t \to x$ and $\omega \to k$ where $k = 2\pi/\lambda$ is the wavenumber and λ is wavelength. In multiple spatial dimensions the wavenumber becomes a wave vector

$$\vec{x} \equiv (x,\, y,\, z) \qquad \leftrightarrow \qquad \vec{k} \equiv (k_x,\, k_y,\, k_z) \ , \tag{10.5}$$

and the FTs are modified accordingly

$$\tilde{f}(\vec{k}) = \int_{-\infty}^{+\infty} e^{i\,\vec{k}\cdot\vec{x}} f(\vec{x})\,d\vec{x} = \mathcal{F}(f(\vec{x})) \ , \tag{10.6}$$

$$f(\vec{x}) = \frac{1}{(2\pi)^3}\int_{-\infty}^{+\infty} e^{-i\,\vec{k}\cdot\vec{x}}\tilde{f}(\vec{k})\,d\vec{k} = \mathcal{F}^{-1}(\tilde{f}(\vec{k})) \ . \tag{10.7}$$

Derivatives and FTs

FTs can be very useful in manipulating differential equations. This is because spatial or time derivatives become algebraic operations in Fourier space. As an example consider the following equation in real space

$$\frac{d}{dx}u(x) = v(x) \quad .$$

Taking the FT of this equation yields

$$-\,i\,k\tilde{u}(k) = \tilde{v}(k) \quad . \tag{10.8}$$

This can be shown as follows: replace $u(x)$ and $v(x)$ by their inverse FTs

$$\frac{d}{dx}\left(\frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-i\,k\,x}\tilde{u}(k)dk\right) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-i\,k\,x}\tilde{v}(k)dk \quad . \tag{10.9}$$

The only bits containing x are the transform kernel (i.e. the plane wave part $\exp(-i\,k\,x)$). Therefore using Leibnitz's integral rule yields

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} -i\,k\,e^{-i\,k\,x}\tilde{u}(k)dk = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-i\,k\,x}\tilde{v}(k)dk \quad .$$

Equating the integrands on both sides we have (10.8).

This can be generalised to higher derivatives:

$$\frac{d^n}{dx^n}f(x) \;\rightleftharpoons\; (-i\,k)^n\,\tilde{f}(k) \quad . \tag{10.10}$$

It can be generalised to 3D too:

$$\begin{aligned} \nabla f(\vec{x}) \;&\rightleftharpoons\; -i\,\vec{k}\,\tilde{f}(\vec{k}) \quad , & \nabla^2 f(\vec{x}) \;&\rightleftharpoons\; -|\vec{k}|^2\,\tilde{f}(\vec{k}) \quad , \\ \nabla\cdot\vec{E}(\vec{x}) \;&\rightleftharpoons\; -i\,\vec{k}\cdot\tilde{\vec{E}}(\vec{k}) \quad , & \nabla\times\vec{E}(\vec{x}) \;&\rightleftharpoons\; -i\,\vec{k}\times\tilde{\vec{E}}(\vec{k}) \quad , \end{aligned} \tag{10.11}$$

where $\nabla^2 f = \nabla\cdot(\nabla f)$ has been used.

10.2. Discrete FTs

DFTs are the equivalent of complex Fourier series, which are defined on a finite length domain, but for a **discretely sampled function** rather than a continuous function. We illustrate here with time and angular frequency.

Time domain – We assume the function is sampled by N equally spaced samples (with N even for simplicity, see later) on a time domain of length $T = N\Delta t$ as illustrated in figure 10.1 ;

$$f_n \equiv f(t_n) \qquad \text{with} \qquad n = 0, 1, 2, ..., N-1 \qquad \text{and} \qquad t_n = n\,\Delta t \ . \tag{10.12}$$

The function $f(t)$ is ***assumed to be periodically extended*** beyond the domain $0 \leq t \leq T$ so that $f(t + mT) = f(t)$ for integer m. For the sampled function, periodicity means $f_N = f_0$ hence we choose to discard f_N in our sequence of samples.

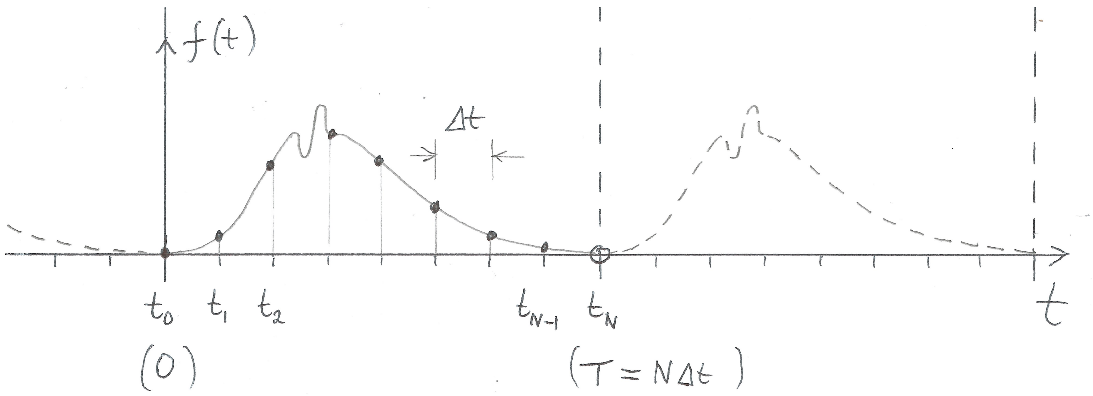

Figure 10.1. Illustration of a signal in the time domain. $N = 8$ here. The signal/function is assumed to be periodically extended beyond $0 \leq t \leq T$.

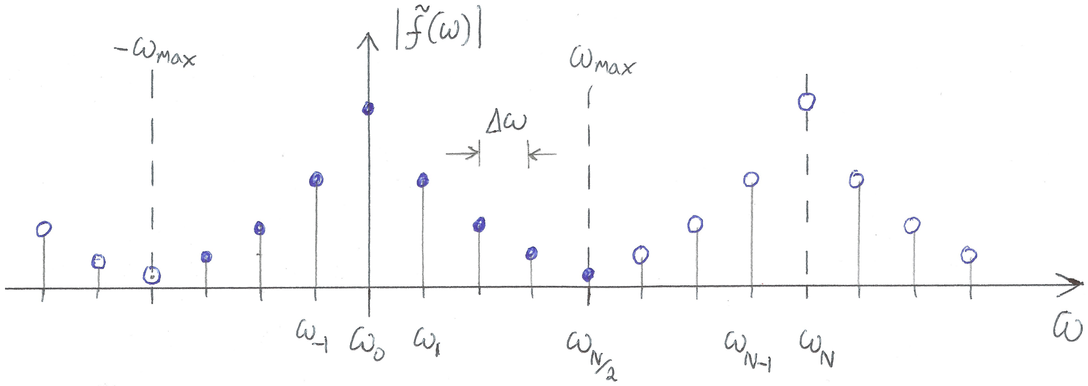

Figure 10.2. Frequency domain and discrete spectrum corresponding to figure 10.1. The discrete spectrum is also periodically extended (beyond $|\omega_{max}|$ in this case). The modulus of $\tilde{f}(\omega)$ is depicted.

Frequency domain – Figure 10.2 illustrates the frequency domain and discrete spectrum of the signal sampled in figure 10.1. The longest wave that can fit exactly into the time domain has an angular frequency $\omega_{min} = 2\pi/T \equiv \Delta\omega$. The shortest wave that can be *recognised* by the grid and fits periodically into the time domain has duration $2\Delta t$ (i.e. one peak and one trough in just two time samples) so that $\omega_{max} = 2\pi/(2\Delta t) = \pi/\Delta t$. This maximum frequency is known as the **Nyquist frequency**

(10.13)
$$\boxed{\omega_{max} = \frac{\pi}{\Delta t} = \Delta\omega\,\frac{N}{2} \quad .}$$

Allowing for negative angular frequencies too, these frequencies define the discrete, finite, angular frequency grid on which $\tilde{f}$ is sampled;
(10.14)
$$\boxed{\tilde{f}_p \approx \frac{1}{\Delta t}\tilde{f}(\omega_p) \quad \text{with} \quad p = -\frac{N}{2}, \ldots, 0, \ldots, \frac{N}{2} \quad \text{and} \qquad \omega_p = p\,\Delta\omega = p\frac{2\pi}{N\,\Delta t} \quad .}$$

Each discrete frequency corresponds to a wave that fits exactly an integer number of times 'p' into the finite domain. Note that the integer index p seems to run over $N+1$ values but in fact the $-N/2$ and $N/2$ samples are degenerate, i.e.,

$$\tilde{f}_{-N/2} \equiv \tilde{f}_{N/2} \quad .$$

so there are only N degrees of freedom. (See section 10.3.)

Why $\tilde{f}_p \approx \tilde{f}(\omega_p)/\Delta t$ rather than $\tilde{f}_p = \tilde{f}(\omega_p)/\Delta t$ in equation (10.14) above? This is because $\tilde{f}_p$ will be used to represent the DFT which is not always exactly $\tilde{f}(\omega)$, the true spectrum of $f(t)$, simply picked out at discrete frequencies ω_p. We assume that $f(t)$ *is* exactly, discretely sampled. If there is no aliasing (see 10.3) then $\tilde{f}_p = \tilde{f}(\omega_p)/\Delta t$. If there is aliasing, then the DFT is a distorted, discrete, version of the true spectrum. (The $1/\Delta t$ factor is just the convention used in the definition of the DFT, see below.)

The DFT – is the analogue of a continuous FT, but with the continuous integral $\int \ldots dt$ replaced by a finite sum $\sum \ldots \Delta t$;

(10.15)
$$\boxed{\tilde{f}_p = \sum_{n=0}^{N-1} f_n\, e^{i\,\omega_p t_n} = \sum_{n=0}^{N-1} f_n\, e^{i\,2\pi pn/N} \quad ,}$$

where $\omega_p\, t_n = \left(\frac{2\pi\, p}{N\Delta t}\right)\,(n\,\Delta t)$ has been used in obtaining the second form. The backwards transform is similarly defined as

(10.16)
$$f_n = \frac{1}{N} \sum_{p=-N/2+1}^{N/2} \tilde{f}_p\, e^{-i\,2\pi pn/N} \quad .$$

The different normalisation factor of $1/N$ accounts for the discrete sampling. This comes from $dt \to \Delta t$ and $d\omega \to \Delta\omega = 2\pi/(N\Delta t)$. (Note that in going from the FS (10.1) to the DFT (10.15) a factor of Δt has been absorbed into $\tilde{f}_p$, i.e., $\tilde{f}_p \approx \tilde{f}(\omega_p)/\Delta t$.) The backward DFT (10.16) is usually defined as

$$\boxed{f_n = \frac{1}{N}\sum_{p=0}^{N-1} \tilde{f}_p \, e^{-i\,2\pi pn/N} \quad ,} \tag{10.17}$$

which is more symmetrical with the forward DFT. Why this is possible will be seen in section 10.3.

Relation to complex Fourier series – Discrete FTs (DFTs) are intimately related to complex Fourier series (CFS). Complex Fourier series represents the discrete spectrum of a ***continuous function*** $f(t)$, also on a domain of finite length. The allowed angular frequencies have the same spacing $\Delta\omega$, but are now infinitely many.

FFTs – Fast Fourier transforms

The DFTs above, equations (10.15) and (10.16), are easily implemented on a computer, however the naive method of implementing it requires $\mathcal{O}(N^2)$ operations. With typical samples volumes of $N \sim 10^6$ in modern applications this becomes prohibitive even on the fastest computers. The DFT can actually be done in $\boldsymbol{\mathcal{O}(N \log_2 N)}$ **operations**; the algorithm for this is known as the **fast Fourier transform** or just **FFT**. For $N = 10^6$, the speed up is $N/log_2 N \ \approx 50,000$ times! FFTs work best when $N = 2^m$ with integer m, and works by recursively splitting the DFT into separate sums over only the even or odd indices. ***The FFT implementation of the DFT is what is used in practice.*** If $N \neq 2^m$, then it is usual to ***pad out*** the samples with zeros until the size is $N = 2^m$. (Non-examinable – The FFT algorithm is outlined at the end of this section.)

DFT in 2D

For a function $f_{n,m} = f(x_n, y_m)$ defined on a 2D grid $x_n = m\Delta x$ for $0 \leq n \leq N-1$ and $y_m = m\Delta y$ for $0 \leq n \leq M-1$ its DFT is given by

$$\tilde{f}_{p,q} = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f_{n,m} \, e^{i\,2\pi pn/N} \, e^{i\,2\pi qm/M} \quad . \tag{10.18}$$

The backward DFT is

$$f_{n,m} = \frac{1}{N\,M}\sum_{p=0}^{N-1}\sum_{q=0}^{M-1} \tilde{f}_{p,q} \, e^{-i\,2\pi pn/N} \, e^{-i\,2\pi qm/M} \quad . \tag{10.19}$$

Going to 3-dimensions just adds another nested sum (with a new, independent index) and another exponential wave factor (incorporating the new dimension) into the transform kernel.

10.3. Sampling & Aliasing

The discrete sampling of the function to be FT'd, given by equation (10.12), introduces some sampling effects and care has to be taken in allowing for them. There is a minimum sampling rate that needs to be used so that we don't lose information. If the function to be sampled fits into the finite length time (or spatial) domain of length T, and is **bandwidth limited** to below the Nyquist frequency then all frequency content of the function is exactly captured by the discrete sampling process. If the function has $\tilde{f}(\omega) \neq 0$ for $|\omega| > \omega_{max}$ then the frequency content for $|\omega| > \omega_{max}$ will be **aliased** down into the range $|\omega| < \omega_{max}$ and distort the DFT compared to the exact FT of $f(t)$. If the original function is bandwidth limited but is longer than T, then it will be ***clipped*** which will introduce a sharp cutoff at $t = 0$ and/or T. This will effectively increase the bandwidth of the clipped function and aliasing will occur again during sampling of it.

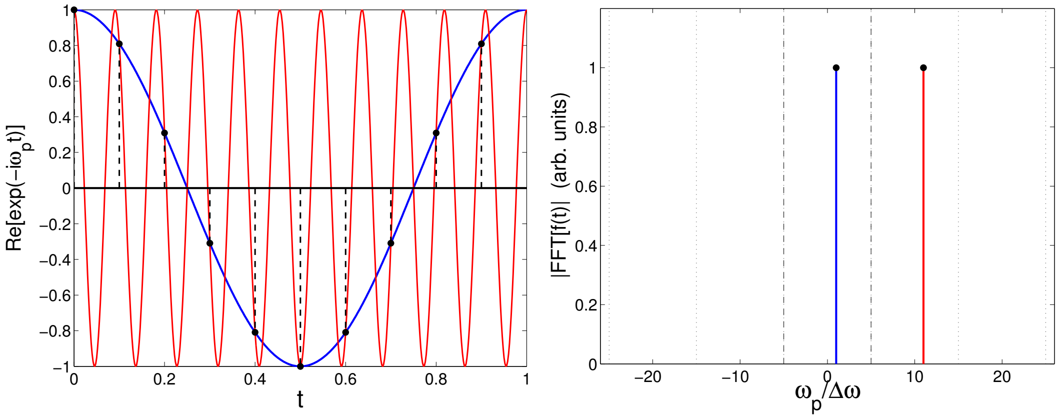

Figure 10.3. (Left) Indistinguishable harmonic waves below and above the Nyquist frequency, for $N = 10$. Blue line: $\omega = \omega_1 = \Delta\omega$. Red line: $\omega = \omega_{N+1} = \Delta\omega + 2\omega_{max}$. (Waves are periodically extended beyond range shown.) (Right) Corresponding (angular) frequency spectrum. The vertical dashed lines lie at $\pm$ the Nyquist frequency.

Aliasing & indistinguishable frequencies – For any wave with a frequency ω_p lying in the frequency domain captured by the DFT, there are higher frequency waves with $\omega_{p'} = \omega_p + m\Omega$ (where $m \in \mathbb{Z}$ and $\Omega = 2\omega_{max}$ is the width of the frequency domain) that look exactly the same to the discrete time-sampling grid. This is illustrated in figure 10.3. This can be understood by considering the kernel of the DFT, i.e., $\exp(i\,\omega_{p'} t_n)$.

$$\begin{aligned}\exp\left[i(\omega_p + m\Omega)t_n\right] &= \exp\left[i\left(\frac{2\pi\, pn}{N} + m\frac{2\pi}{\Delta t} n\Delta t\right)\right] = \exp\left(i\frac{2\pi\, pn}{N} + i2\pi\, m\, n\right) \\ &= \exp\left(i\frac{2\pi\, pn}{N}\right) = \exp\left(i\omega_p\, t_n\right) \quad .\end{aligned}$$

This has the following implications;

(a) It explains why $\tilde{f}_{-N/2} \equiv \tilde{f}_{N/2}$.

(b) It also explains why equation (10.17) for the backwards DFT 'works'. The "negative" frequency part of the spectrum $p = \left\{-\frac{N}{2}+1, \ldots, -1\right\}$ maps to the positive spectrum at $p' = p + N = \left\{\frac{N}{2}+1, \frac{N}{2}+2, \ldots, N-1\right\}$ lying beyond the Nyquist frequency. In other words, although the $\sum_{p=N/2+1}^{N-1}$ parts of (10.17) are above the Nyquist frequency, they happen to capture the desired negative frequencies within the Nyquist range.

(c) Aliasing:

$$\tilde{f}_p = \sum_m \mathcal{F}(f)|_{\omega_p + m\Omega} \quad , \tag{10.20}$$

i.e., the DFT (the LHS) folds in the Fourier components from the *exact FT of the continuous function* from the indistinguishable set of waves $\omega'_p = p + m\Omega$.

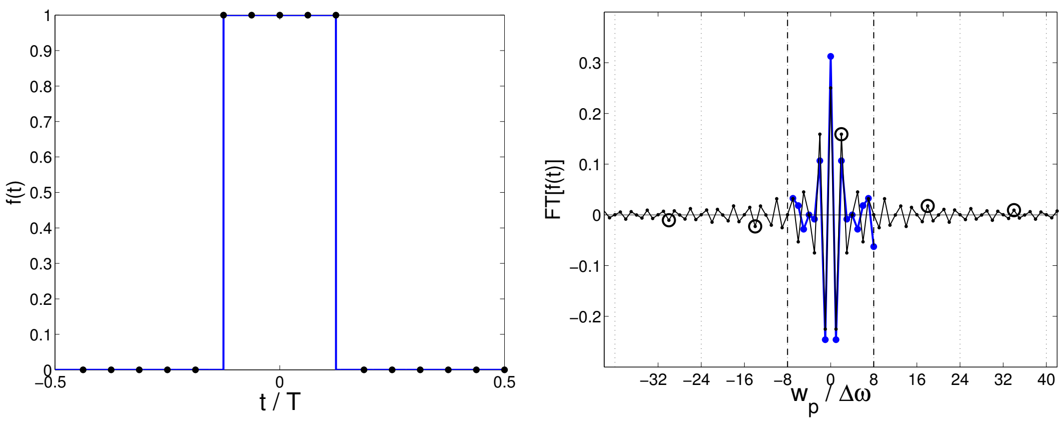

Figure 10.4. Aliasing: (Left) $f(t)$ in the time domain; centred "rect" function of width $T/4$. Sampled with $N = 16$. (Right) The DFT (blue, solid line + markers), and exact FT (black, thin solid line) in the frequency domain. The open circles denote Fourier components outside the range $-\omega_{max} \le \omega \le \omega_{max}$ that are aliased into that range during the DFT. The real part of the DFT & FT are shown. The DFT been divided by Δt.

Figure 10.4 shows the phenomenon of aliasing for a top-hat function

$$f(t) = \begin{cases} 1 & |t| < a/2 \\ 0 & |t| > a/2 \end{cases} \quad .$$

10.4. Spectral methods – Solving the Poisson Equation by FFT

Methods using FTs to solve PDEs are known as **spectral methods**. As an example we consider the general Poisson equation

$$\nabla^2 u(\vec{x}) = \rho(\vec{x}) \quad , \tag{10.21}$$

where ρ is some source density and u is some potential that we are seeking to solve for. For electrostatic problems this specialises to

$$\nabla^2 \phi(\vec{x}) = -\rho_c(\vec{x})/\epsilon_o \quad ,$$

where ϕ and ρ_c are the electrostatic potential and charge-density, respectively, and for (Newtonian) gravitational problems

$$\nabla^2 \Phi(\vec{x}) = 4\pi G\, \rho(\vec{x})$$

where Φ is gravitational potential, G is Newton's constant and ρ is the mass density of matter.

We have seen how to solve this system using an iterative method in Section 9.2. The iterative method works for any BC; periodic, fixed, etc. For periodic BCs, using FFTs is much more efficient.

The exact analytical solution can easily be written in terms of FTs as follow. As we have seen the Laplacian is replaced by the factor $-|\vec{k}|^2$ in Fourier space so that exact solution in Fourier space is

$$\tilde{u}(\vec{k}) = -\frac{\tilde{\rho}(\vec{k})}{|\vec{k}|^2} \quad . \tag{10.22}$$

To obtain the exact solution in real (coordinate) space we just need to inverse transform the above

$$u(\vec{x}) = \mathcal{F}^{-1}\left[-\frac{\tilde{\rho}(\vec{k})}{|\vec{k}|^2}\right] \quad . \tag{10.23}$$

In practice for general $\rho(\vec{x})$, tractable, analytic expressions for the Fourier transform integrals will be difficult (even impossible) to find.

Equipped with a discrete Fourier transform, the obvious way to solve (10.21) numerically (in, e.g., 2D) would seem to be to take a FFT of the gridded source density $\rho_{n,m}$ to get $\tilde{\rho}_{p,q}$, divide by $|\vec{k}_{p,q}|^2$ (where $(k_x)_p = p\pi/\Delta x$ and $(k_y)_q = q\pi/\Delta y$) and then take the backwards DFT to get $u_{n,m}$. However, this does not yield the same thing as solving the discretised PDE, as we did in section 9.2 (i.e. equation (9.24)). The reason is that we have not yet taken into account the FD approximation of the Laplacian.

10.4.1. Application to discrete system – 1-D

We discretise the system as usual with grid spacing h and use a second order finite difference scheme for the Laplacian

$$\frac{u_{j+1} - 2\,u_j + u_{j-1}}{h^2} = \rho_j \, . \tag{10.24}$$

Expanding each term through the backwards DFT we have

$$u_j = \frac{1}{N} \sum_p \tilde{u}_p \, e^{-i\,2\pi\, pj/N} \,, \tag{10.25}$$

and

$$u_{j\pm 1} = \frac{1}{N} \sum_p \tilde{u}_p \, e^{-i\,2\pi\, p(j\pm 1)/N} = \frac{1}{N} \sum_p \tilde{u}_p \, e^{\mp i\,2\pi\, p/N} \, e^{-i\,2\pi\, pj/N} \quad . \tag{10.26}$$

Putting these into (10.24) gives

$$\frac{1}{N} \sum_p \tilde{u}_p \, e^{-i\,2\pi\, pj/N} \left[e^{+i\,2\pi\, p/N} + e^{-i\,2\pi\, p/N} - 2 \right] = \frac{h^2}{N} \sum_p \tilde{\rho}_p \, e^{-i\,2\pi\, pj/N} \quad . \tag{10.27}$$

Equating the bits inside the sum we have

$$\tilde{u}_p = \frac{h^2 \, \tilde{\rho}_p}{\left[e^{+i\,2\pi\, p/N} + e^{-i\,2\pi\, p/N} - 2 \right]} \quad , \tag{10.28}$$

and since

$$e^{i\,\theta} = \cos\theta + i\,\sin\theta \,, \tag{10.29}$$

the denominator simplifies to

$$\left[e^{+i\,2\pi\, p/N} + e^{-i\,2\pi\, p/N} - 2 \right] = \left[e^{+i\,\pi\, p/N} - e^{-i\,\pi\, p/N} \right]^2 = (2i)^2 \, \sin^2(\pi\, p/N) \quad , \tag{10.30}$$

giving

$$\tilde{u}_p = -\frac{h^2 \, \tilde{\rho}_p}{4\sin^2(\pi\, p/N)} \tag{10.31}$$

Notice that the 'monopole' $\tilde{u}_0$ diverges if $\tilde{\rho}_0 \neq 0$. (Note that $\tilde{\rho}_0$ is the average source density on the finite, periodically repeated, domain.) Physically, potential is only defined to within a constant so we can safely set $\tilde{\rho}_0 = 0$ to make the potential in real space zero when for a system where all the 'mass' is uniformly spread out. We then obtain the solution in coordinate space by taking the inverse DFT of (10.31)

$$u_j = -\frac{h^2}{4\,N} \sum_p \frac{\tilde{\rho}_p}{\sin^2(\pi\, p/N)} \, e^{-i\,2\pi\, pkj/N} \quad . \tag{10.32}$$

In general the DFTs are replaced by black-box FFT routines from standard libraries and the algorithm can be summarised as

- Generate source lattice ρ_j.
- FFT source lattice $\rho_j \to \tilde{\rho}_p$.
- Solve for $\tilde{u}_p$ (and remove monopole).
- Inverse FFT $\tilde{u}_p \to u_j$.

The method can be easily extended to higher dimensions, e.g., 2D or 3D lattice.

10.5. Fast Fourier Transforms - FFTs (non-examinable)

The FFT algorithm for doing a discrete Fourier transform in $\mathcal{O}(N \log N)$ operation was described by Daniel & Lanczos in 1942 but rediscovered and popularised in the computer age by Cooley & Tukey in 1965.

The trick is to split the DFT into separate sums involving only even (e) and odd (o) indices

$$\begin{aligned} \tilde{f}_k = \sum_{n=0}^{N-1} f_n \, e^{i\,2\pi kn/N} &= \sum_{n=0}^{N/2-1} f_{2n} \, e^{i\,2\pi k2n/N} + \sum_{n=0}^{N/2-1} f_{2n+1} \, e^{i\,2\pi k(2n+1)/N} \,, \\ &= \sum_{n=0}^{N/2-1} f_{2n} \, e^{i\,2\pi kn/(N/2)} + e^{i\,2\pi k/N} \sum_{n=0}^{N/2-1} f_{2n+1} \, e^{i\,2\pi kn/(N/2)} \,, \quad (10.33) \\ &= \tilde{f}^e_k + e^{i\,2\pi k/N} \tilde{f}^o_k \,. \end{aligned}$$

(10.34)

The two functions $\tilde{f}^e_k$ and $\tilde{f}^o_k$ are then periodic over $N/2$ i.e. half the range of the original function. This splitting requires N operations. We can iterate this process and split both $\tilde{f}^e_k$ and $\tilde{f}^o_k$ into even and odd parts themselves

$$\tilde{f}^e_k = \tilde{f}^{ee}_k + e^{i\,2\pi k/N} \tilde{f}^{eo}_k \,, \tag{10.35}$$

and

$$\tilde{f}^o_k = \tilde{f}^{oe}_k + e^{i\,2\pi k/N} \tilde{f}^{oo}_k \,, \tag{10.36}$$

with the new functions being periodic over the range $N/4$. The second splitting requires $N/2$ operations.

Assuming $N = 2^p$ (i.e. even), after p even/odd splitting we are left with N functions of period 1 which are each trivially FT'd; the FT of each is the same as itself

$$\tilde{f}^{eooeeoeoeo...e}_k = f_n \,, \tag{10.37}$$

for some n.

All we have left to do is identify which combination of $eoeeo...e$ corresponds to each n - this can be done by assigning the binary value $e \equiv 0$ and $o \equiv 1$ to each element in the sequence and then **reversing** the sequence. The series of 1s and 0s obtained is a binary representation of n.

The complete method requires

$$N \times p = \frac{N \log N}{\log 2} \sim \mathcal{O}(N \log N) \,, \tag{10.38}$$

operations. As an example consider a problem with $N \sim 10^4$ samples, in this case the FFT is faster than the naive DFT by a factor of about 750.