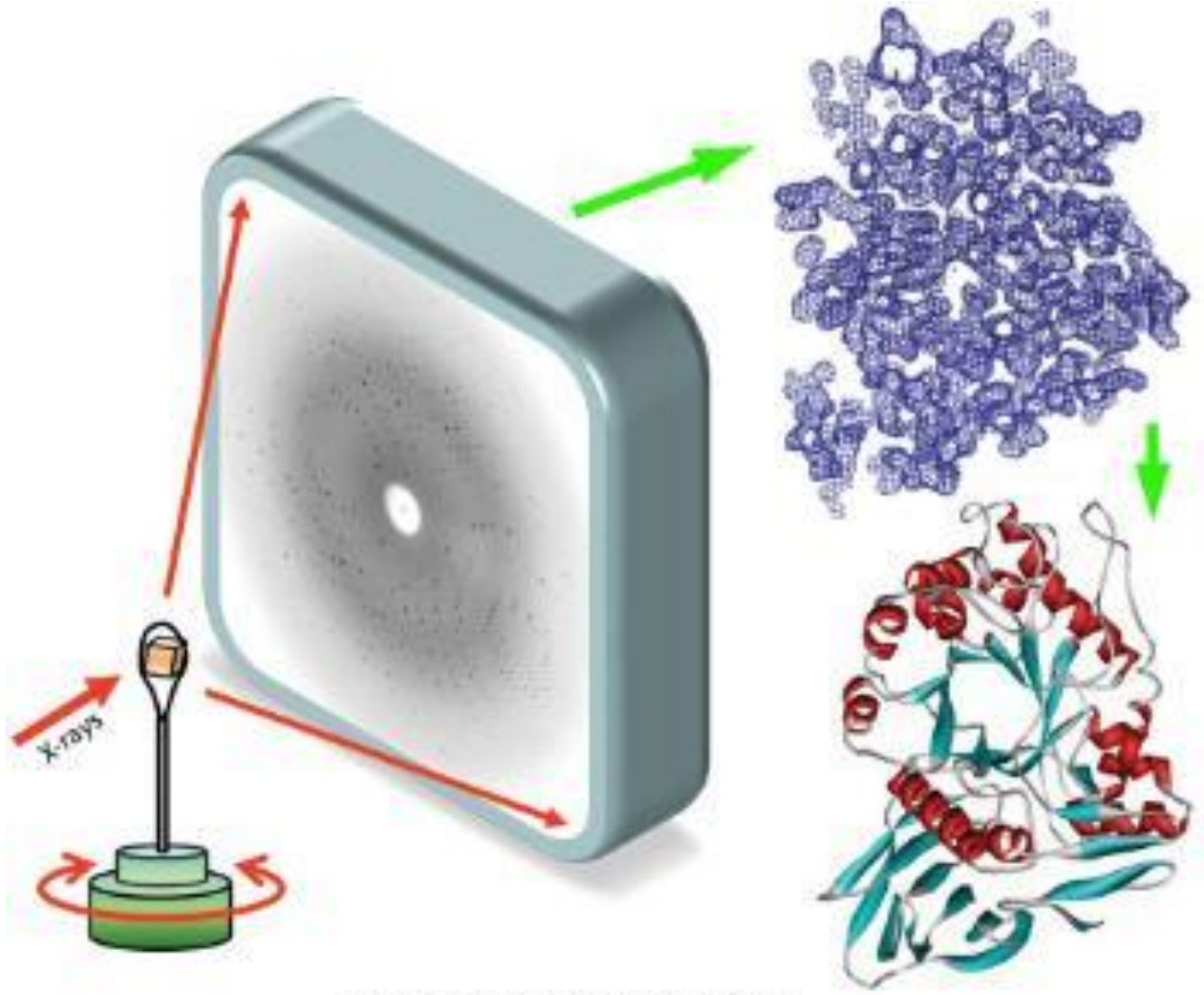
The Fourier transform in X-ray crystallography



Modified from Garland Science 2010

Electron density vs. Structure factor

The electron density can be obtained from the Fourier transform of the structure factors. These are obtained from the diffraction spots observed in the X-ray diffraction experiment. The scattering in the various planes of the crystal is recorded in terms of reciprocal lattice vectors cataloged by the Miller indices h, k and l. The Fourier transform for the density is:

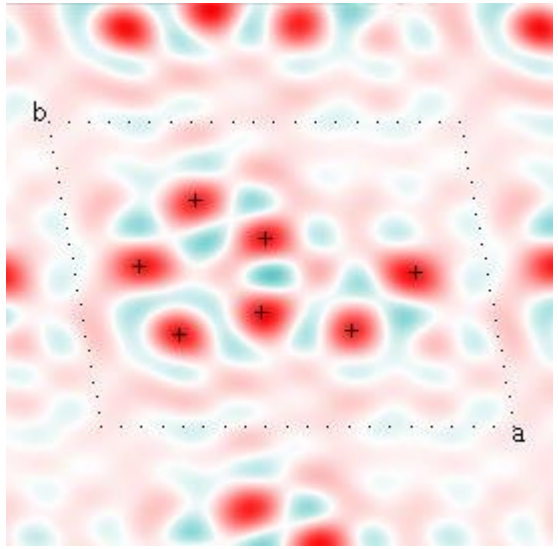
$$\rho(x,y,z) = \sum_{h=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} c_{hkl}\, exp\left\{ i\left(\frac{2\pi h}{a}\right)x + i\left(\frac{2\pi k}{b}\right)y + i\left(\frac{2\pi l}{c}\right)z \right\}$$

The inverse Fourier transform is:

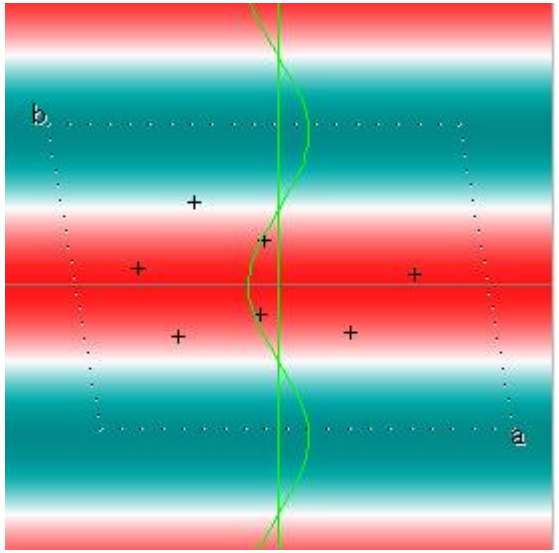
$$c_{hkl} = \frac{1}{abc} \int_0^a \int_0^b \int_0^c \rho(x,y,z)\, exp\left\{ i\left(\frac{2\pi h}{a}\right)x + i\left(\frac{2\pi k}{b}\right)y + i\left(\frac{2\pi l}{c}\right)z \right\} dxdydz$$

A 2-D visual example of structure factors

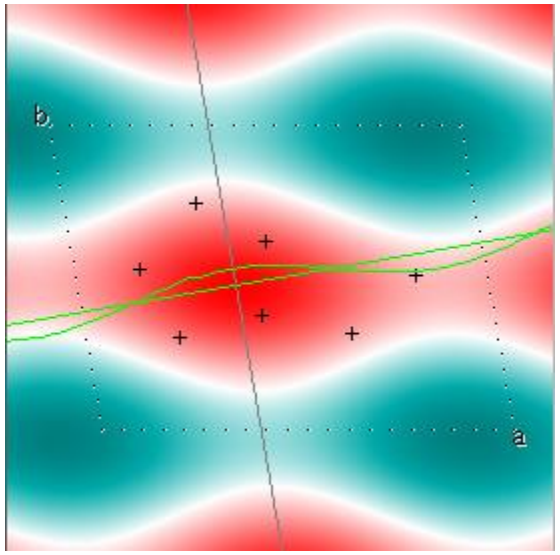
Here is a nice example, from Kevin Cowtan's Interactive Structure Factor Tutorial. The example is 2-dimensional, and shows how rapidly the structure factors, *with their phases*, converge to the target structure. The target electron density function looks like this:



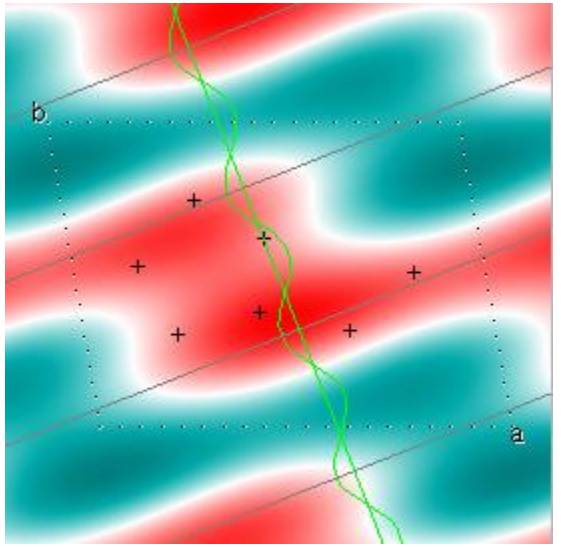
Dr. Kevin Cowtan, York University



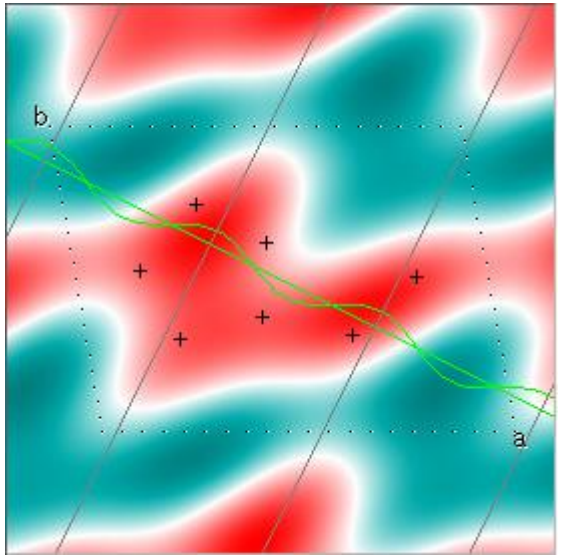
(0,1)



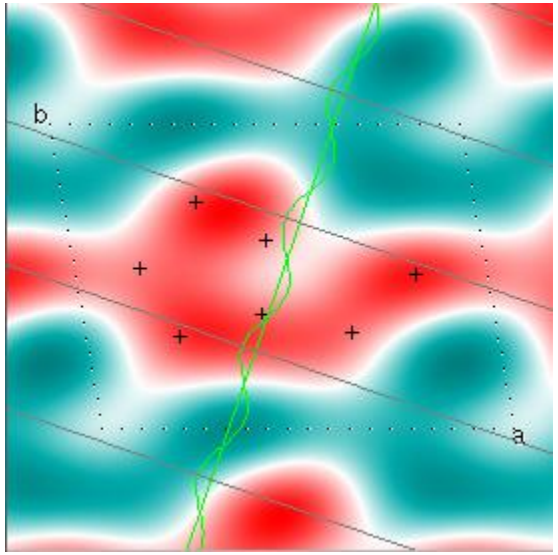
(1,0)



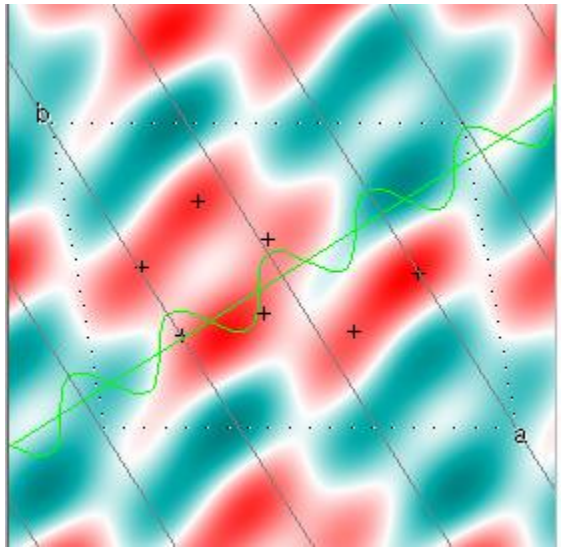
(-1,2)



(-2,1)



(1,2)



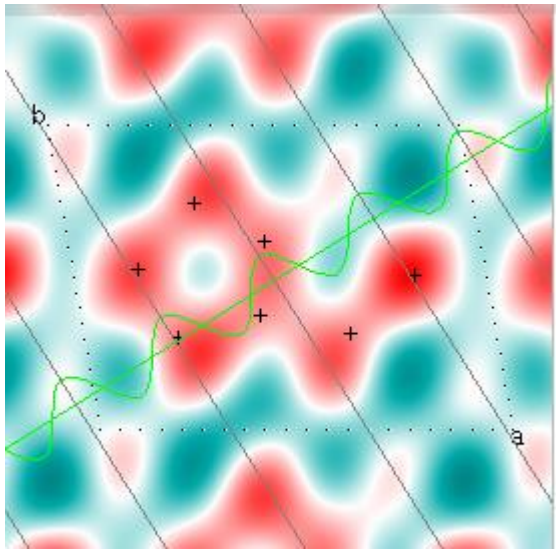
(3,-2)

A 2-D visual example of structure factors

Cowtan's simulation leads to the approximate Fourier synthesis of the target from just the seven largest structure factors: those corresponding to

$$(h,k)=(0,1),(1,0),(-1,2),(-2,1),(1,2),(3,-2),(3,1).$$

Here is how the synthesis proceeds, step by step, each time adding in the next structure factor. The unit cell (not orthorhombic!) is outlined in dots.



Fast Fourier Transform

The need for a fast Fourier Transform

While NMR line shapes can be obtained in theory using an analytical function, try doing this if there are 50 nucleic oscillating in the sample.

There are many examples in science where we need FT Methods. We will use Fourier transform infrared in the class. In that method the signal is composed of the detector response at various different positions of the moving mirror in an interferometer. We must take the FT of the "interferogram" in order to obtain the optical response.

In powder X-ray diffraction the electron density is obtained as the FT of the measured intensities at various h,k,l (Miller) indices, corresponding to Bragg planes in the material.

The discrete Fourier transform

An FFT is the fastest known method to compute the Discrete Fourier Transfer (DFT). Let x_0, ..., x_{N-1} be complex numbers. The DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}} \qquad k = 0, 1, \ldots . . N - 1$$

Evaluating this definition directly requires O(N^2) operations: there are N outputs X_k, and each output requires a sum of N terms. An FFT is any method to compute the same results in O(N log N) operations.

The Cooley-Tukey algorithm

By far the most commonly used FFT is the Cooley-Tukey algorithm. This is a divide and conquer algorithm that recursively breaks down a DFT of any composite size $N = N_1 N_2$ into many smaller DFTs of sizes N_1 and N_2, along with $O(N)$ multiplications by complex roots of unity traditionally called twiddle factors (after Gentleman and Sande, 1966).

This method (and the general idea of an FFT) was published by J.W. Cooley and J.W. Tukey in 1965, but it was later discovered that those two authors had independently re-invented an algorithm known to Gauss around 1805.

Divide and conquer

The best known use of the Cooley–Tukey algorithm is to divide the transform into two pieces of size $N/2$ at each step, and is therefore limited to power-of-two sizes, but any factorization can be used in general (as was known to both Gauss and Cooley/Tukey). These are called the **radix-2** and **mixed-radix** cases, respectively (and other variants such as the split-radix FFT have their own names as well). Although the basic idea is recursive, most traditional implementations rearrange the algorithm to avoid explicit recursion. Also, because the Cooley–Tukey algorithm breaks the DFT into smaller DFTs, it can be combined arbitrarily with any other algorithm for the DFT.

Even and odd terms

The Radix-2 DIT algorithm rearranges the DFT of the function x_n into two parts: a sum over the even-numbered indices n = 2m and a sum over the odd-numbered indices n = 2m + 1.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{i2\pi(2m)k}{N}} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{i2\pi(2m+1)k}{N}}$$

A radix-2 decimation-in-time (DIT) FFT is the simplest and most common form of the Cooley–Tukey algorithm, although highly optimized Cooley–Tukey implementations typically use other forms of the algorithm as described below. Radix-2 DIT divides a DFT of size *N* into two interleaved DFTs (hence the name "radix-2") of size *N*/2 with each recursive stage.

Recursive approach

Radix-2 DIT first computes the DFTs of the even-indexed inputs $x_{2m} = x_0, x_2, \ldots x_{N-2}$ and of the odd-indexed inputs $x_{2m+1} = x_1, x_3, \ldots . x_{N-1}$, and then combines those two results to produce the DFT of the whole sequence. This idea can then be performed recursively to reduce the overall runtime to $O(N \log N)$. This simplified form assumes that N is a power of two. Since the number of sample points N can usually be chosen freely by the application, this is often not an important restriction.

The twiddle factor

One can factor a common multiplier $e^{-\frac{2\pi i}{N}k}$ out of the second sum, as shown in the equation below. It is then clear that the two sums are the DFT of the even-indexed part x_{2m} and the DFT of odd-indexed part x_{2m+1} of the function. Denote the DFT of the ***E***ven-indexed inputs x_{2m} by E_k and the DFT of the ***O***dd-indexed inputs x_{2m+1} by O_k and we obtain:

$$X_k = \underbrace{\sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk}}_{\text{DFT of even-indexed part of } x_m} + e^{-\frac{2\pi i}{N}k} \underbrace{\sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}}_{\text{DFT of odd-indexed part of } x_m}$$