Protein Reconstruction Methods in Cryogenic Electron Microscopy and the Contrast Transfer Function

Bidit Acharya, Tracy Lou, Josh Nixon, Alex Pearson

Contents

1 Biological Overview

1.1 Proteins

Proteins are our bodies' workhorses that carry out the cellular process we need to function everyday. By scouring deep in the back of our mind, we may recall some foreign notion of a protein from an introductory biology class. To fathom the importance of understanding protein structure and function, we need to explore its composition.

Everything begins with *DNA (deoxyribonucleic acid)*. DNA is the what many call the "blueprint" or the molecule that holds all the genetic information necessary for carrying out life. Consequently, this means DNA contains all the instructions needed to create functional proteins. To do this step, however, is not so straightforward. Before proteins are assembled, DNA has to be transcribed to *RNA (ribonucleic acid)* by an enzyme called *RNA polymerase*. RNA is less chemically stable which allows it be a catalyze reactions and reconfigure itself better for the next step: translation. Translation is a process that occurs in the ribosome, a site in the cell where proteins are made. The just transcribed RNA find its way to the ribosome where it is read and decoded into amino acids, producing a specific chain of amino acids or polypeptides. Proteins are, in essence, just a chain of amino acids. Very briefly, that is how the primary structure of these cellular machines are constructed. We can then loosely conclude that DNA codes for protein which basically perform all of our necessary cellular duties; this order is called the *central dogma*. Lastly, we also realize that each segment of DNA codes for specific protein that perform specific tasks.

1.2 Protein Function and Folding

Functional proteins need to go through four different steps or four different structures. The primary structure is only a unique sequence of a chain of amino acids. This is the first step in determining the specificity of the its function. The next step involves how the protein is folded, which is what mainly dictates its uniqueness and in turn its function. How the is configured affects how it interacts with other proteins and cellular structures.

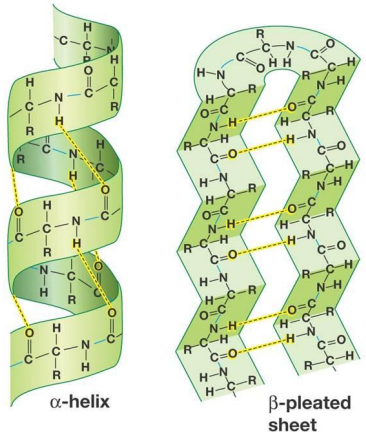


Figure 1.1: α helices and β sheets

Proteins fold into two general shapes, called their secondary struture: the *alpha helix*−which looks like a helix−and the *beta sheet* (which look like a pleated sheet). These irregular folds are held together by atomic interactions governed by an electrostatic Coulombic force. In addition, there are helper proteins called, *chaperonins*, that assist in folding. The individual atoms form hydrogen bonds with each other.

Hydrogen bonds are non-covalent bonds−bonds between atoms that do not share valence electrons−formed between a hydrogen atom and another electronegative atom. Electronegative atoms are commonly known as halogens which are found in the second to last column on the right of the periodic table. Because these halogens are so electroneg-

ative, they induce a partial charge on itself and the hydrogen atom to which it is bonded. This attraction between the partial charges are the weakest type of atomic bonding, but they are the glue that holds this complex structure together. There are other molecular interactions that support the protein's structure such as disulfide bonding. It is important to note that not the entire chain of amino acids gets folded into a single alpha helix or a single beta sheet; rather, only a segment of the amino acid sequence is configured into one of the two shapes. One of the motivations behind why proteins fold is explained by thermodynamics.

Naturally, a protein wants to fold into an energetically favorable stable state. A protein will continue to move and adjust until it is in its most stable state. In addition, misfolding of a protein can lead denaturation−the loss of protein structure and function. A malfunctioning protein in a body can lead to severe diseases such as Alzheimer's and cystic fibrosis. Because of the delicacy of this process, understanding how proteins fold is a very active area of research in biophysics and biochemistry.

Once the alpha helices and beta sheets are configured, these structures are assembled in a geometric shape that further contributes to the protein's identity. This tertiary structure mainly involves the folding of the protein's side chains which can vary for each amino acid; there are twenty amino acids identified by a special group of different molecules for each. To differentiate, the secondary structure only includes folding of the protein's backbone, which is the same for every amino acid. The folding taking place in the tertiary for the most part is dominated by the polar and non-polar interactions between the amino acids. The protein would fold in a way such that non-polar amino acids would curl inwards to avoid interaction with the water molecules that would be surrounding it. By the end of this step, the protein has a globular shape arranged such that hydrophilic amino acids are placed on the outside and the hydrophobic ones are hidden within. Biology is complex; hence, a protein's structure cannot be so simple.

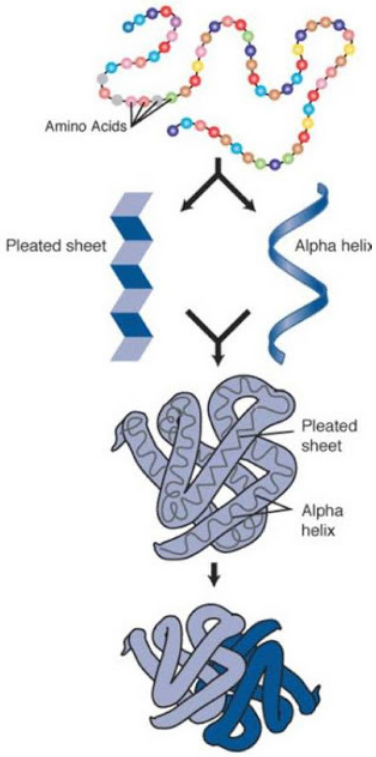
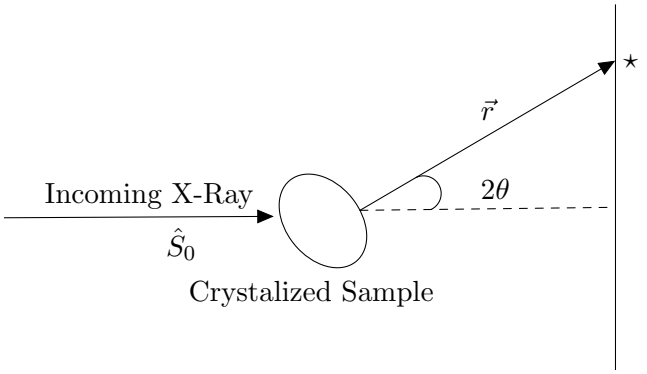


Figure 1.2: Primary, secondary, tertiary and quaternary structures of proteins

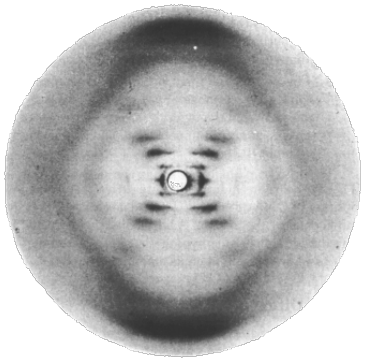
Most often, one chain of amino acids is not enough to form a functional protein. The last step in this process is the formation of the quaternary structure. Simply put, the quaternary structure consists of multiple amino acid chains and thus multiple protein subunits that are uniquely arranged together. The organization in this step is arguably the most important in determining the types of interactions the protein can form and how it can be used. Important proteins who have a quaternary structure include hemoglobin, which transports oxygen throughout the human body, and ion channels which have copious amounts of bodily responsibilities.

1.3 A Broad Brush Over of X-ray Crystallography

Determining the structure of proteins has been a constant continuing field that has since made major leaps in its achievements. Traditionally and for a very long time, biochemists would−and some still−toil away in their laboratories to get a protein in a crystalline form, which is extremely difficult thermodynamically and chemically. Crystallized protein were desired upon the realization that crystals cancel noise and amplify the amount of X-ray scattering. Today, it is still considered an art to be able to crystallize protein. Crystals are just a regular array atoms, but they have special properties such as symmetry and being able to diffract beams in a distinctive way that can be described physically by *Bragg's Law*. Bragg's Law enables biochemists to understand and navigate the diffraction pattern retrieved from the interaction between the x-rays and atoms. In addition, we cannot detect scattering from a molecule, so being in crystal form is the solution. This method of shooting a beam, particularly an X-ray beam, at a crystallized structure is called *X-ray crystallography*. X-rays are used instead of say, light from light microscopy, because the X-ray's wavelength is much shorter and coincidentally matches the distance between atoms. Thus, we can resolve detail at an atomic scale.



(a) Scattering



(b) X-ray diffraction pattern of DNA (Franklin, 1953)

X-ray crystallography is a nontrivial technique that is still widely used and is the one of the oldest and precise methods for determining the atomic structure of a protein. Its main drawback is the process of crystallizing the protein and acquiring an abundance of it use. Once all of this is achieved, the crystal is then placed in front of an intense beam of X-rays. By doing so, the beams will interact with the crystal at an atomic level and scatter at different angles. When the X-rays scatter, they also interfere constructively and destructively, therefore producing a diffraction pattern on a piece of film. In addition, the crystal is rotated in all directions to get a variety of similar or missing reflections. The data this give us are actually the electromagnetic intensities of every orientation of the crystal. Lastly, all of these data sets are compiled with laborious and arduous amounts of biochemistry and to attain the structure of the protein.

1.4 Limitations of X-ray Crystallography

For instance, it took roughly 30 years for biochemists, particularly Ada Yonath, to uncover the structure of the ribosome. That much time was necessary because of technological limitations. The ribosome is large and has many subunits. Furthermore, crystallization of the ribosome seemed impossible, because it was already difficult to crystallize, and then they to get enough of it. As a consequence these biochemists were too impatient and decided use cryo-EM; cryo-EM, however, was only able to give a rough shape at 20Å; it was not exactly of the highest resolution; an Angstrom, Å, is a unit of length between any atom. Therefore, many of the questions regarding the ribosome had to be answered by X-ray crystallography.

The Yonath group were steadily able to successfully crystallize the ribosome and reach its maximum resolution of 3Å! This alone consumed eleven years. The slow unfolding of the ribosome structure continued once many realized ribosome crystallization was feasible. Though, this was not the end Yonath's work—alternatively her graduate students' work—since there are actually various types of ribosomes, each with dissimilar amounts of large and small subunits.

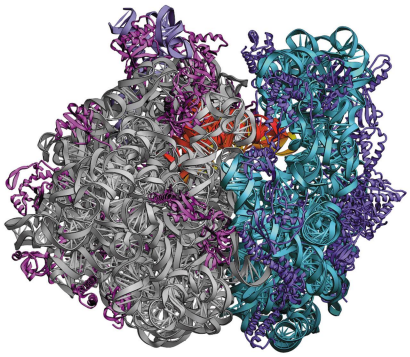


Figure 1.3: Tertiary structure of 70S Ribosome

Recall that ribosomes carry out protein synthesis and in short, translate the DNA "code" into life. In fact, studying the structure and function won Yonath and two others the Nobel Prize in 2009. Understanding the innermost properties of the ribosome is critical and immediately applicable. For example, scientists were able to use the 3D models generated to develop antibiotics that block the function of bacterial ribosomes. The 3D models depict how the antibiotics bind to the ribosome. If the bacteria is unable to bind to its ribosomes, it cannot produce proteins; and thus it will not be able to survive. Hopefully, this example demonstrates the inherent complications in structural biology in regards to time.

1.5 The Phase Problem

Elucidating a protein's structure is made especially more difficult by the fact that every time a physical measurement is made, information concerning the phase is lost. Each diffraction spot on the film corresponds to a point in the crystal lattice and represents a scattered wave with a specific amplitude and a relative phase; however, only the amplitude can be recovered. Thus, this information needs to be recovered to accurately determine the structure of the protein from the all the different diffraction data. This is a huge setback in that what X-ray crystallography does is only give scattering intensities. Currently, there are many tricks to solve the *phase problem*, but two of them will be described below.

The first method is called *molecular replacement*. This technique requires an existing 3D model from prior experiments and assays that were done. If this existing structure is closely related to the one desired, we can make the assumption that we can use the relative phases of the the first

structure to solve for the missing phases in the desired structure. Note that this only works if we have a good model for a reasonably large portion of the structure. This is a reliable technique when a majority of the structure has already been modeled.

A more direct method is called *anomalous scattering*. Unlike molecular replacement method, this does not require a pre-existing 3D model. Rather, heavy atoms−those with high atomic scattering factors−would be added to the protein. The incoming X-ray would cause the electrons in the heavy atoms to absorb energy and transition to a higher energy level (energy levels are quantized). Particles always want to be in the lowest energy state possible, so absorption of energy is unnatural, meaning that once the electron reaches a higher energy level, it would immediately go back to its ground state and emits a photon as a result. Fortunately, the primary structure and thus the atoms are known, so missing phases can be computed from absorption values. During this process, however, the protein's structure is interfered with by appending heavy atoms. As such, it raises the question on the veracity of the 3D structure conceived.

1.6 A Different Brush Over Cryogenic Electron Microscopy

Evidently, the phase problem significantly hinders the progress of recovering a protein's structure. Another technique to producing 3D models of molecular structures is called *electron microscopy (EM)*. This method does not require the protein to be crystallized, but it is not as feasible since preparing the sample for EM basically destroys it. Hence, the recent development of *cryogenic electron microscopy (cryo-EM)* has revolutionized the field of structural biology for the five or so years. In addition, EM is able to solve the phase problem, since a lens is involved. This also stems from the fact that electrons can be focused by an electric field, whilst photons from the X-rays cannot be focused; X-ray lenses do not exist. Lenses essentially takes all the light that diverged and focuses them to the same point on the detector.

In X-ray crystallography, we only get the scattering intensities in the diffraction pattern; in EM, however, we obtain images from which we can work backwards utilizing the lens specifications to resolve the missing phases. Experimentally, instead of attacking the protein with destructive chemicals in EM, in cryo-EM the protein is simply frozen in liquid nitrogen. Ice reduces the damage the protein will face when beams of electrons are discharged at it. Despite going great lengths to protect the protein, recovering a 3D model continues to be challenging. It is not wise to pound the protein with many electrons; in truth, only doses on the scale of maybe one to twenty electron per pixel in an image. As a result, the images obtained contain lots of noise, making it difficult to discern what is true and what is not. Furthermore, because the protein is not being hit by photons that are massless, the electrons do shift the protein, and this drift has to be corrected.

1.7 Applications of Cryo-EM

In spite of all the hurdles, cryo-EM has transformed an entire field of biology. In the past, such as the period of resolving the structure of the ribosome, many vital proteins−and more generally, macromolecules−were untouched, because gathering the necessary amount of crystals was very nearly hopeless. This technique has accelerated the number of molecular structure visualizations. Structures that would take many decades to solve are now solved within a couple of years. For this project, our goal is to use a data set on the *Zika virus* given to us and reconstitute the structure. It is important to remark that the reconstruction of this virus took less than a decade, mainly

because the World Health Organization declared the Zika virus a global health emergency, since the revelation that it causes microcephaly−a birth defect where a the baby's head is smaller than expected−in fetuses. Moreover, critical structures such as *GroEL*, a chaperonin−a protein that assists in protein folding−, was swiftly modeled by cryo-EM techniques.

Not only is cryo-EM able to resolve static structures, it can be used to visualize how other proteins interact with each other. For instance, the 3D structure of cytoplasmic *dynein*, the one that binds to microtubules, was recently revealed in the journal,*Proceedings of the National Academy of Sciences of the United States of America* or *PNAS* for short, by the Kikkawa lab. This is another example of using cryo-EM to determine the structure and function of significant complexes. Dynein is an essential motor protein that moves along microtubules (intracellular structures that are basically "freeways" involved in transportation and movement) and contribute greatly to meiosis (the process of cell division). Because dynein is a very "active" protein, cryo-EM has the potential to reveal which specific atoms are interacting or "walking" with each other on the dynein complex on the microtubules.

Additionally, cryo-EM is able to report back structures that were previously notoriously difficult to see using X-ray crystallography. In 2013, Yifang Cheng and his lab published a paper in the journal, *Nature*, revealing they were able to elicit of the *TRPV1 ion channel*; in fact, this was the first *de novo*−constructing the tertiary structure form the primary structure−structure to be enlightened by cryo-EM.

TRP (transient receptor potential) channels are ion channels that are found on plasma membranes and respond to physical and chemical stimuli. Ion channels' function is to control the flow of ions into and out of the cell. TRP channels are found in numerous biological processes and fields. Their study is applicable in the pharmacology; visualizing how these TRP channels react to physical stimuli caused by drugs is relevant to understanding how the drug and infection is affecting major organs in the body; the role of these channels have emerged in the study of several human diseases. They also are known to be important targets for analgesic drugs. TRP channels are more examined within the field of neuroscience, since they are associated with how signals and senses are transferred from one cell to another−this is called *transduction*.

The TRPV1 channel in particular is colloquially called the *capsaicin* receptor. The function of the TRPV1 channel is to detect and regulate body temperature. A branch of this applies in how we sense heat and pain. Coincidentally, capsaicin is a compound found in chili peppers; and there exist receptors for it on our tongues. In the past, single-molecule microscopy and X-ray crystallography were able to reveal a general structure of the structure at large, but elucidating the structure of very small membrane proteins, such as TRP channels, have been a challenge to achieve at atomic, high resolution. For TRP channels specifically, techniques such as X-ray crystallography and *nuclear magnetic resonance (NMR)* limit the information obtained. These channels are fluid and mobile. Proteins, especially ion channels, are conformationally dynamic, in that they carry out their function by changing their arrangement when they interact with something else. X-ray crystallography can really only get a sliver of the structure at a particular state. Common obstacles in crystallizing, purifying, and expressing membrane proteins also hampered the progress of obtaining TRP protein structures. Cryo-EM circumvents this obstacle of coaxing the protein into a well-ordered, homogeneous crystal. To conclude, Cheng's group's ability to showcase the TRPV1 structure *de novo* highlights the power of the novel cryo-EM method. It was a lengthy process but much quicker and feasible than if they had attempt to do it by X-ray crystallography.

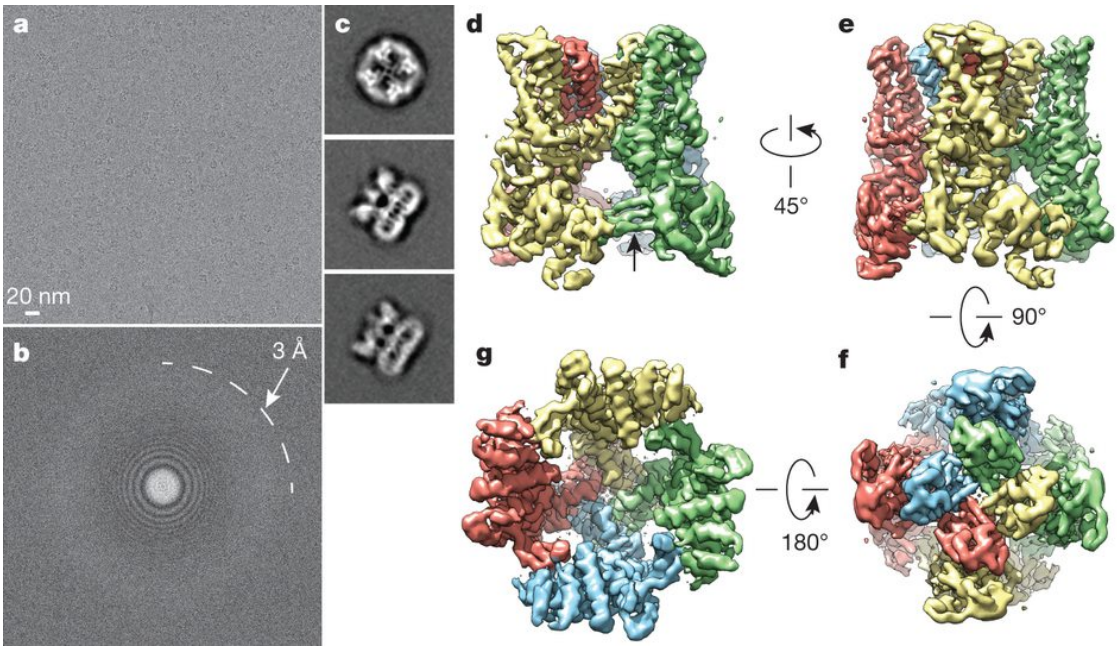


Figure 1.4: **a**: Representative electron micrograph of TRPV1 protein embedded in a thin layer of vitreous ice recorded at a defocus of 1.7 μm. **b**: Fourier transform of micrograph shown in a, with Thon rings extending to nearly 3$\mathring{A}$. **c**: Enlarged views of three representative 2D class averages show fine features of tetrameric channel complex. **d-g**: 3D density map of TRPV1 channel filtered to a resolution of 3.4$\mathring{A}$(scaled to atomic structure) with each subunit color-coded. Four different views of the channel are shown, from side (*d, e*), top (*f*) and bottom. **g**: The arrow in panel *d* indicates β sheet structure in the cytosolic domain of TRPV1. Description courtesy of Cheng[6]

The Cheng group used a variety of electron microscopes (Tecnai TF20, TF30 Polara, and Tecnai T12 microscope), cameras (Gatan−CCD and TemF816−CMOS) , and methods to obtain EM data, including *negative staining* to serve as an initial model for cryo-EM.

2 Simulating Cryo-EM Images

The mathematical model for simulating Cryo-EM images is fairly involved. We begin with a protein and its electron density function $\rho : \mathbb{R}^3 \to \mathbb{R}$. We assume that all copies of the protein are identical. With this assumption, we can say that each image is a picture of a *single* protein from different viewing directions.

Definition. A matrix F is a 3D coordinate frame if $\det F = 1$ and all of its rows has magnitude of 1. Mathematically speaking, for each $i \in \{1, 2, 3\}$, $|F \cdot e_i| = 1$ where e_i is the ith standard basis in $\mathbb{R}^3$. Since F represent rotation in $\mathbb{R}^3$, we shall sometimes refer to coordinate frames as rotational matrices[1].

We model a microscope viewing direction with a frame F. If $F = \begin{bmatrix} | & | & | \\ \vec{a} & \vec{b} & \vec{c} \\ | & | & | \end{bmatrix}$ then $\vec{a}, \vec{b}$ are axes

in the image plane and $\vec{c}$ is the viewing direction.The image obtained from the microscope viewing direction F is a projection onto the image plane $\mathcal{P}_F = \text{span}\{\vec{a}, \vec{b}\}$.
The image of ρ from viewing direction F is modeled by a function $I_F : \mathbb{R}^2 \to \mathbb{R}$ where

$$I_F(x, y) = \int_{\mathbb{R}} \rho(\vec{a}x + \vec{b}y + \vec{c}z)dz \tag{2.1}$$

Remark. I_F is called a *Radon projection*[2] of ρ.

If we wish to simulate an EM image given a ρ, we pick a viewing direction F, and numerically perform I_F to get an image. However, this is $\mathcal{O}\left(n^2 \cdot \zeta\right)$ where n is the image width and ζ is the integration complexity factor. This is computationally expensive. The following theorem however, provides an alternative route; one that is orders of magnitude faster in fact.

Theorem 2.1. (*Projection/Fourier Slice Theorem*) Suppose F is a viewing direction and $P_F = \text{span}\{\vec{a}, \vec{b}, \vec{c}\}$ is the image plane of F, then

$$\underbrace{\mathfrak{F}\{\rho\}|_{P_F}}_{\substack{\text{Fourier transform} \\ \text{restricted to } P_F}} = \underbrace{\mathfrak{F}\{I_F\}}_{\substack{\text{2D Fourier} \\ \text{transform of } I_F}} \tag{2.2}$$

Proof. See Appendix B □

To simulate an image of ρ along F_p therefore

$$\text{Compute } \mathfrak{F}\{\rho\} \implies \overset{\text{sample } \mathfrak{F}\{\rho\} \text{ along } P_F}{\implies} \overset{\text{we have } \mathfrak{F}\{I_F\} \text{ by } FST}{\implies} \implies \mathfrak{F}^{-1}\left\{\mathfrak{F}\{I\}\right\}$$

Obviously, having only image simulating algorithms is not enough. We also need to be able to reconstruct the protein in $\mathbb{R}^3$. While there are many such 3D reconstruction algorithms, the one used by this class, and hence this group is the *weighted back projection algorithm*. The next section is partially devoted to this algorithm. But first, we shall go into the theory of Transfer functions of Imaging systems. This extends to the said algorithm while providing a nice segue into the realm of Contrast Transfer Functions– our extra deliverable.

3 Back Projection and Transfer Functions

3.1 Linear Systems and Convolution

An Imaging System is a time-invariant system that spits out a response given a stimulus. It is often helpful to view it as a black box model. This system can be an optical system like the entirety of a microscope or just an algorithm employed within. Let S such a system. Following the terminology of Goodman [2] and Radermacher [7], we model the system as an operator that maps input functions onto a set of output functions. Thus, if ρ is an input function with output ϕ, we write

$$\phi(x, y, z) = S\left[\rho(x', y', z')\right] \tag{3.1}$$

As usual, we say S is linear if

$$S\left[a\rho_1(\mathbf{x}) + b\rho_2(\mathbf{x})\right] = aS\left[\rho_1(\mathbf{x})\right] + bS\left[\rho_2(\mathbf{x})\right] \tag{3.2}$$

For brevity we will henceforth write $\mathbf{x}$ to denote a vector of variables.
Next, we introduce a crucial idea from Fourier analysis that becomes relavent in the next few sections: convolution of two functions.

Definition. Let $f, g : \mathbb{R}^n \to \mathbb{R}$ with $\left|\int_{\mathbb{R}^n} f(\mathbf{x})d\mathbf{x}\right| < \infty$ and $\left|\int_{\mathbb{R}^3} g(\mathbf{x})d\mathbf{x}\right| < \infty$. The **convolution** of f and g, denoted $f \star g$ is given by

$$(f \star g)(\mathbf{z}) = \int_{-\infty}^{\infty} f(\mathbf{x})g(\mathbf{z} - \mathbf{x})d\mathbf{x}$$

Convolutions are useful because, as the following theorem asserts, they behave very nicely with Fourier transforms.

Theorem 3.1. (*Convolution Theorem*) If $\mathfrak{F}(f)$ is the Fourier Transform of f, then

$$\mathfrak{F}(f \star g) = \mathfrak{F}(f) \cdot \mathfrak{F}(g), \tag{3.3}$$

where $\cdot$ denotes point-wise multiplication of the functions.

Proof. See Appendix B □

Remark. The convolution theorem is important in applications because it allows us to compute convolutions by multiplying their Fourier transforms, which is faster than computing the convolution directly, especially when the Fast Fourier Transform(FFT) is involved. From a computational standpoint, the convolution of f and g on points $\{z_1, \ldots, z_n\}$ is given by

$$(f \star g)(z_i) = \sum_{n=0}^{n=m-1} f(x_n)g(z_i - x_n) \tag{3.4}$$

Clearly, the algorithmic complexity of computing the convolution directly is therefore $\mathcal{O}(n^2)$, while the complexity of computing the convolution using the Fast Fourier Transform is $\mathcal{O}(n \log n)$.

3.2 The Point Spread Function and Transfer Function

Definition. A *point-spread function* of an imaging system, denoted by $h_S(\mathbf{x})$ describes the image of a single point (modeled as a perfect point) when input to the system. Meaning:

$$S\left[\delta(\mathbf{x}')\right] = h_S(\mathbf{x}) \tag{3.5}$$

If the system S is clear from context then we just write $h(\mathbf{x})$

Let S be a linear imaging system with

$$\phi(x, y, z) = S\left[\rho(x', y', z')\right] \tag{3.6}$$

Recall that given the 'niceness[3]' of the input function ρ, it can be expressed as a sum of elementary δ-functions [4] as:

$$\rho(x', y', z') = \int_{\mathbb{R}}\int_{\mathbb{R}}\int_{\mathbb{R}} \rho(\bar{x}, \bar{y}, \bar{z})\delta(x' - \bar{x}, y' - \bar{y}, z' - \bar{z})d\bar{x}d\bar{y}d\bar{z} \tag{3.7}$$

Here, δ is the 3D delta functions. For a more detailed exposition on δ-functions please refer to the first chapter of Griffel[3]. Combining these two equations and using the linearity of S

$$\phi(x,y,z) = \int_{\mathbb{R}}\int_{\mathbb{R}}\int_{\mathbb{R}} \rho(\bar{x},\bar{y},\bar{z}) S\left[\delta(x'-\bar{x},y'-\bar{y},z'-\bar{z})\right] d\bar{x}d\bar{y}d\bar{z} \tag{3.8}$$

Thus what this is telling us is that to know the effect of S on any ϕ, we need to know its response to the delta function. But notice that this is exactly what the *point-spread function* is. For brevity, we denote the point spread function by h. Mathematically,

$$h(x,y,z;\bar{x},\bar{y},\bar{z}) = S\left[\delta(x'-\bar{x},y'-\bar{y},z'-\bar{z})\right] \tag{3.9}$$

or, in other words[5]:

$$h(x-\bar{x},y-\bar{y},z-\bar{z}) = S\left[\delta(x'-\bar{x},y'-\bar{y},z'-\bar{z})\right] \tag{3.10}$$

Combining all of these equations, we finally get

$$\phi(x,y,z) = \int_{\mathbb{R}^3} \rho(\bar{x},\bar{y},\bar{z}) h(x-\bar{x},y-\bar{y},z-\bar{z}) d\bar{x}d\bar{y}d\bar{z} \tag{3.11}$$

This is great news because the right hand side of Eq.3.11 resembles a convolution of ρ and h, in fact, it is exactly that. i.e.

$$\phi(\mathbf{x}) = \rho(\mathbf{x}) \star h(\mathbf{x}) \tag{3.12}$$

Where $\star$ is the convolution operator and $\mathbf{x} = (x,y,z)$, written for brevity. The Convolution Theorem tells us

$$\mathfrak{F}\{\phi\} = \mathfrak{F}\{\rho\} \cdot \mathfrak{F}\{h\} \tag{3.13}$$

By definition, the Fourier transform of the point-spread function is called the **Transfer function** and we write $\mathfrak{F}\{h(\mathbf{x})\} = H(\mathbf{X})$, where $\mathbf{X}$ is the corresponding frequency parameter in the frequency domain. Thus,

$$\rho(\mathbf{x}) = \mathfrak{F}^{-1}\left\{\mathfrak{F}\{\phi\} \cdot \frac{1}{H(\mathbf{X})}\right\} \tag{3.14}$$

As a piece of terminology: Eq.3.14 represents a '**deconvolution**' of $\phi(\mathbf{x})$ by the point spread function $h(\mathbf{x})$

3.3 Unweighted Back Projection

As mentioned before, an Imaging system could model an optical system such as a microscope or just an algorithm used within. As an application to the theory of Transfer Functions, we now look into a widely used 3D reconstruction algorithm: Weighted back projection. The advantage this method has over other iterative or Fourier reconstruction algorithms is that it can be modeled as a Linear Time Invariant (LTI) system; this means that we can use our results from the preceding sections to study its Transfer functions and compute its weighted filter. On top of that, this algorithm is much faster on large data sets than most other reconstruction algorithms out there[6]. But before we get into its $H(\mathbf{X})$, we shall first go over a simple unweighted back projection algorithm to remind us of the specifics.

As before, let $\rho : \mathbb{R}^3 \to \mathbb{R}$ be a electron density of a protein. Instead of assuming the rotation matrices, we shall construct rotation matrices assuming the angle of projection for each projection. Let ρ be projected at angles θ_k, γ_k (Since we are working in $\mathbb{R}^3$ two angle measures are required.); with the resultant projection being $I_k(x_k, y_k)$. Obviously we need to change the coordinate system for ρ to match with that of I_k. If $\mathbf{x} = (x, y, z)$ be the coordinates with respect to ρ and $\mathbf{x}_k = (x_k, y_k, z_k)$ the coordinates with respect to the projection I_k. The change of coordinates can be easily obtained with two rotation matrices $F_{\theta_k}, F_{\gamma_k}$:

$$\mathbf{x}_k = F_{\theta_k} F_{\gamma_k} \mathbf{x} \tag{3.15}$$

where [7] :

$$F_{\theta_k} = \begin{pmatrix} \cos\theta_k & 0 & -\sin\theta_k \\ 0 & 1 & 0 \\ \sin\theta_k & 0 & \cos\theta_k \end{pmatrix} \text{ and } F_{\gamma_k} = \begin{pmatrix} \cos\gamma_k & \sin\gamma_k & 0 \\ -\sin\gamma_k & \cos\gamma_k & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Just to clarify F_{γ_k} rotates the protein by γ_k around the $z-$axis and then F_{θ_k} rotates it around the y-axis by $-\theta_j$. With the ρ's coordinates rotated to be compatible with that of I_k, we can now perform the Radon Transform of ρ to get I_k

$$I_k = \int \rho(D_{\theta_k} D_{\gamma_k} [x, y, z]^T) dz_k = \int \rho(x_k, y_k, z_k) dz_k \tag{3.16}$$

Recall that the back projection b_k is given by

$$b_k(x_k, y, k, z_k) = I_k \star [\delta(x_k, y_k) \operatorname{rect}(z_i)] \text{ where } \operatorname{rect}(c) = \begin{cases} 1 & -\frac{D}{2} \leq c \leq \frac{D}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

and

$$B(\mathbf{x}) = \sum_{k=1}^{N} b_k(x_k, y_k, z_k) \tag{3.18}$$

3.4 Weighted Back Projection and its Transfer Function $H(\mathbf{X})$

The previous section tells us that back projection algorithm is linear. This means we can use ideas from §3.2 to find the transfer function for the back-projection. This is where the weighted part comes in. The transfer function H provides a sort of weighted filter to each back projection body b_k

Recall that a point spread function for any linear system is the response of the system given an input of a single point in $\mathbb{R}^3$. In essence, we compute the output function of the back projection algorithm given the input function $\delta(\mathbf{x})$.

1. Given viewing angles θ_k, γ_k, projecting $\delta(x, y, z)$ along them gives,

$$I_k = \int_{\mathbb{R}} \delta(D_{\theta_k} D_{\gamma_k} [x, y, z]^T) dz_k = \int_{\mathbb{R}} \delta(x_k, y_k, z_k) dz_k = \delta(x_k, y_k) \tag{3.19}$$

2. From Eq. 3.17, we get that

$$b_k = \delta(x_k, y_k) \star [\delta(x_k, y_k) \operatorname{rect}(z_i)] = \delta(x_k, y_k) \operatorname{rect}(z_i) \tag{3.20}$$

3. Thus by Eq. 3.18 the point spread function for the back projection is

$$B = \sum_{k \in [N]} \delta(x_k, y_k) \operatorname{rect}(z_k) \tag{3.21}$$

By definition, the transfer function is just the Fourier transform of the point spread function B.

$$\begin{aligned}
H(\vec{\omega}) &= \mathfrak{F}\{B\} \\
&= \sum_{k \in [N]} \mathfrak{F}\left\{\delta(x_k, y_k) \operatorname{rect}(z_k)\right\} \\
&= \sum_{k \in [N]} \int_{\mathbb{R}} \int_{\mathbb{R}} \int_{\mathbb{R}} \delta(x_k, y_k) \operatorname{rect}(z_k) e^{-2\pi i \mathbf{x}_k \cdot \vec{\omega}_k} dx_k dy_k dz_k && \vec{\omega}_k = \left(\omega_k^{\{x\}}, \omega_k^{\{y\}}, \omega_k^{\{z\}}\right) \text{ in the spatial frequency domain} \\
&= \sum_{k \in [N]} \int_{\mathbb{R}} \operatorname{rect}(z_k) e^{-2\pi i z_k \omega_k^{\{z\}}} dz_k && \text{Since } \int_{\mathbb{R}^2} \delta(x_k, y_k) dx_k dy_k = 1 \\
&= \sum_{k \in [N]} \int_{-\frac{D}{2}}^{\frac{D}{2}} e^{-2\pi i z_k \omega_k^{\{z\}}} dz_k \\
&= \sum_{k \in [N]} \frac{\sin\left(\pi D \omega_k^{\{z\}}\right)}{\pi \omega_k^{\{z\}}} \\
&= \sum_{k \in [N]} D \operatorname{sinc}\left(\pi D \omega_k^{\{z\}}\right)
\end{aligned}$$

Finally we change the coordinates from $\left(\omega_k^{\{x\}}, \omega_k^{\{y\}}, \omega_k^{\{z\}}\right)$ back to the standard Fourier basis $\left(\omega^{\{x\}}, \omega^{\{y\}}, \omega^{\{z\}}\right)$. Eq. 3.15 tells us that $\omega_k^{\{z\}}$ is the final component of the vector $\vec{\omega}_k = D_{\theta_k} D_{\gamma_k} \vec{\omega}$, which comes out to be:

$$\omega_k^{\{z\}} = \omega^{\{x\}} \sin\theta_k \cos\gamma_k + \omega^{\{y\}} \sin\theta_k \sin\gamma_k + \omega^{\{z\}} \cos\theta_k \tag{3.22}$$

Putting them all together:

$$H\left(\vec{\omega}\right) = \sum_{k \in [N]} D \operatorname{sinc}\left(D\pi(\omega^{\{x\}} \sin\theta_k \cos\gamma_k + \omega^{\{y\}} \sin\theta_k \sin\gamma_k + \omega^{\{z\}} \cos\theta_k)\right) \tag{3.23}$$

4 The Phase Contrast Transfer Function

The contrast transfer function is a type of transfer function. Before the image we get is formed, the scattered and unscattered electron beams are focused by the objective lens. This intervention

impacts the final image we retrieve from the microscope by giving us an impure representation of the object. In turn, this also affects how the object is reconstituted in 3D. And so, we need to take into account the imaging conditions in which the image was taken, specifically the properties of the lens. Coincidentally, that is also what the contrast transfer function is. It is an approximation of the effects the microscope inflicts on the sample. Being aware of how to CTF acts is crucial for a correct interpretation of our data, since it limits how much information can be translated into our image.Specifically, it is taken from the coefficient in front of the sine term for the wave phase shift.

To preface understanding the *contrast transfer function (CTF)* in cryogenic electron microscopy (cryo-EM), we need to realize that cryo-EM is just a type of *transmission electron microscopy (TEM)*. Recall that in cryo-EM, the sample being studied has been put in a very cold?cryogenic, you might say?state. Simply put, cryo-EM is TEM in liquid nitrogen. The cryo-EM technique is popular in the field of structural biology, a branch of molecular biology concerned with studying the composition and structure of macromolecules such as proteins to understand their function.

4.1 Cryo-EM and TEM: What's the difference?

Much of the theory and methods behind processing and obtaining images in TEM is the same as in cryo-EM. The difference between TEM and cryo-EM is that the latter allows us to observe samples that have not been altered chemically. Typically, to study macromolecules in their native state at such high resolution, we would have to damage the sample critically by dousing it in dangerous chemicals (heavy metal salt solutions) to get, say a protein, in a fixed state. The method of fixing the state of the sample commonly used in TEM is called *negative staining*. Negative staining is a commonly used technique in EM to obtain contrast in a thin slice of what is to be studied. It is called negative, because the background is stained rather than the sample; similarly, if the sample were stained, it would be positive staining. Some advantages of negative staining is that it is much easier to do, is resistant to radiation, and provides a high contrast image; however, there is always the possibility of the chemicals damaging the structure enough to cause it collapse. Imaging, too, is imperfect, since movement can cause lots of background distortion. In addition, the electron beam would interact with the atoms within the chemical stain more than in ice producing more noise. When the sample is plunged in vitreous ice, its native conformation is unaltered; and the hydrogen bonds, which are critical in forming the crystal surrounding the structure, are preserved. This also allows little background. Nonetheless, this technique is experimentally more difficult to carry out successfully.

4.2 Images obtained from microscope

Images of the sample are obtained at many different viewpoints from the transmission electron microscope. The significance of the contrast transfer function (CTF) lies within the microscope, specifically the objective lens. In an ideal world, we would not lose any information of our sample during imaging; however, that is not the case, since it turns out electron lenses have large aberrations which limit the resolution and in turn how much information we are able to get from our images. Theoretically, the CTF is a function that encapsulates how the microscope modifies the sample and transfer that data to a recording device. For example, a subwoofer in music describes what the speakers do to sound at lower frequencies. How the CTF does this is in the properties of

the objective lens used. The CTF is better understood by knowing the order of how an image is produced by the electron microscope.

Once the microscope and sample are configured into their rightful positions, beams of electrons are passed through the sample, hitting the objective lens,which focuses the electrons back onto the back focal plane−where diffraction of waves occurs−and then recording that information onto the image plane. This image is what we get at the end of this process. It is important to note that unlike photons, electrons have mass; so, when they pass through the object (this is why the sample must be thin), the electron also scatters at an angle when it hits specific atoms. Recall one of the conclusions from the double-slit experiment is that electrons −photons, really− can act as both a particle and a wave. Because so many electrons scatter at different angles, we are able to recover a diffraction pattern like in X-ray crystallography. When an electron is scattered at some angle, θ, it traverses a longer path than that of an unscattered electron to the image plane. Working backwards from the diffraction pattern, we can reconstruct the general molecular structure of our object through fancy computer algorithms. In fact, within this method of reconstruction is where the CTF strikes.

4.3 Phase Contrast Transfer Function Equation

From Eq. 3.12, because taking the convolution of two functions is computationally difficult, we take the Fourier transform of the equation to give[9]

$$\Phi(\mathbf{k}) = P(\mathbf{k}) \cdot H(\mathbf{k}) \tag{4.1}$$

This is equation used to recover the the true structure of the object from the EM images. $\mathbf{k}$ is the spatial frequency vector; in other words, it is the frequency related to distance any structure can be distinctly be distinguished−the resolution−. For high resolution images, there would be many electrons scattered at high angles (Bragg's Law). This can be found from, [1]

$$\mathbf{k} = \left(C_s\lambda^3\right)^{\frac{1}{4}} k \tag{4.2}$$

$\Phi(\mathbf{k})$ is the Fourier Transform of the image represented by ϕ. $P(\mathbf{k})$ is the Fourier Transform of the object, so the atoms in the structure. $H(\mathbf{k})$ is the Fourier Transform of the point spread function. This is the *contrast transfer function* $H(\mathbf{k})$ is given by [9]

$$H(\mathbf{k}) = \sin(\gamma(\mathbf{k})) - W\cos(\gamma(\mathbf{k})) \tag{4.3}$$

$W(\mathbf{k})$ is the amplitude contrast arising from inelastic scattering. Assuming the sample is thin, from the weak phase object approximation, the entire amplitude contrast term can be ignored since there would not be enough inelastic scattering due to the absence of heavy atoms. Though amplitude contrast is present, it is very small and insignificant; it is sometimes taken to be a constant. Hence, only the phase term is taken, i.e., the sine term.

The contrast transfer function harks from the lens aberrations and defocusing effects inherent in the imaging lens that modifies the image. Thus, the phase CTF is $\sin\gamma(\mathbf{k})$. Where [8]

$$\gamma(\mathbf{k}) = \frac{2\pi}{\lambda}\left(C_s\frac{\lambda^3k^4}{4} + \Delta z\frac{\lambda^2k^2}{2}\right) \tag{4.4}$$

where C_s is the spherical aberration constant, Δz is the defocus value, and λ is the wavelength of the electron from the microscope.

Spherical aberration occurs when there is a blurring effect. This arises when the lens is unable to converge the incoming rays at higher angles of incidence to the focus point. Instead, the lens focuses the rays to a point closer to it. In other words, the microscope is nearsighted. The contrast transfer function takes into account other properties of the lens such as the chromatic aberration. Finding the value of C_s is unique to the microscope.

The defocus term measures how much the incoming ray deviates from the focal point. Δz is the distance away from the ideal optical axis. Interestingly, at perfect focus, when the Δz is zero, a good image cannot be obtained, since there is no contrast between atoms and structures. Furthermore, the defocus is at the control of the experimenter. When taking images, the CTF is sinusoidal crossing the **k** axis many times and eve more so at higher **k**. When the CTF crosses that axis, at each zero, information about the image is lost. In addition, this information is recovered by looking at other images and finding complements. These complementary images are found when the experimenter changes the defocus value of the microscope, since the defocus will change the shape of CTF and have it cross the **k** axis at different points. Obviously, the best transfer function is when there are fewest zeros. Otto Scherzer optimized this in 1949 by balancing the spherical aberration with a negative value of the defocus. This is know as the Scherzer defocus.[8]

$$\Delta z = -1.2(C_s\lambda)^{\frac{1}{2}} \tag{4.5}$$

The equation for the phase contrast transfer function above is only the ideal, calculated CTF. There are more physical components that limit the phase contrast transfer function. In fact, without these factors, the CTF would continue forever into higher spatial frequencies, resolving structures at impossible resolutions. The CTF is dampened by envelope functions.

There are two common envelope functions whose effects dominate the CTF the most and determine the information limit of the microscope. The envelope terms dampen strongly at higher spatial frequencies until no phase information can be obtained They are the temporal coherency envelope function, $E_c(\mathbf{k})$ and the spatial coherency envelope function, $E_s(\mathbf{k})$. E_c is accounts for the chromatic aberrations in the microscope, the focal and energy spread, and the voltage and current instabilities. $E_c(\mathbf{k})$ accounts for the finite incident beam convergence, taking into account the size of the sample. These are defined as [8]

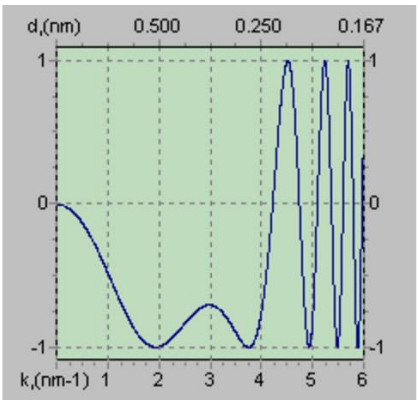
$$E_s(u) = \exp\left[-\left(\frac{\pi\alpha}{\lambda}\right)^2\left(C_s\lambda^3k^3 + \Delta z\lambda k\right)^2\right] \tag{4.6}$$

$$E_c(u) = \exp\left[-\frac{1}{2}(\pi\lambda\delta)^2u^4\right] \tag{4.7}$$

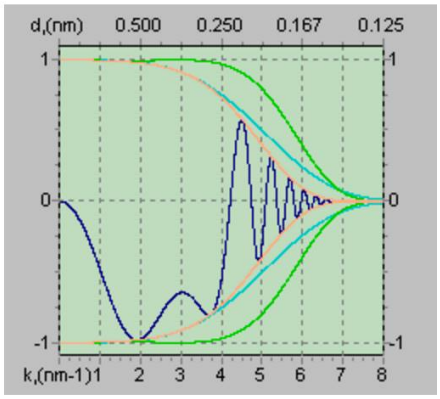
where δ is the defocus spread due to the chromatic aberration.

Thus, the effective phase contrast transfer function is actually [8]

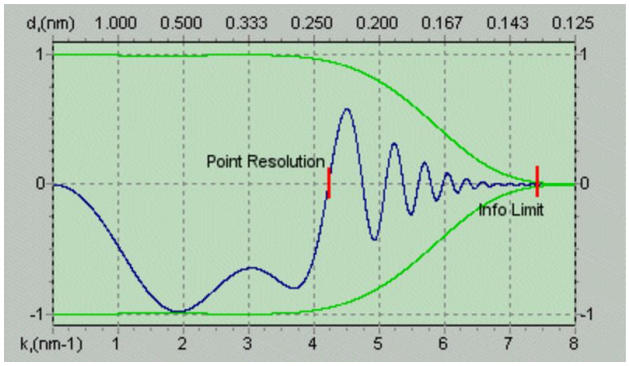
$$H_{\text{eff}} = E_c(\mathbf{k})E_s(\mathbf{k})H(\mathbf{k}) \tag{4.8}$$



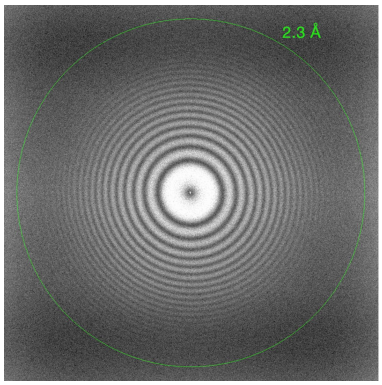
(a)



(b)



(c)



(d)

Figure 4.1: CTF graphs and the Thon ring

4.4 Graphing the Phase CTF

Fig (**a**)-(**c**) courtesy of Maxim V. Sidorov[7] and (**d**) courtesy of *DirectElectron*[8]

(a) This is a graph of a CTF that does not have any damping envelope functions applied. Notice that it starts as 0 and decreases. Notice that there is a lot of contrast at low spatial frequencies, suggesting that objects far away from each other can be resolved well. Once the CTF crosses the frequency axis and obtains a "zero", this is the point resolution that can be observed in figure (c). This is the maximum amount of defocus that can be applied before information about the image is lost.

(b) Unlike the graph of the CTF in figure (a), the CTF is dampened and eventually converges to zero permanently, that suggests the maximum spatial frequency at which the structure can be resolved or observed. The temporal coherence envelope is applied in blue and the spatial coherence envelope is applied in green.

(c) The point resolution or the theoretical maximum resolution is obviously marked as well as the

maximum spatial frequency denoting the information limit. This means that at that spatial frequency is where phase information can no longer be obtained.

(d) Once the Fourier Transform of the image obtained from the microscope is applied, a Thon ring pattern appears. Every time the CTF has a zero, a new Thon ring is added to the image. The last, or outermost Thon ring gives the maximum resolution we can resolve of the structures from the images. In that example, it suggests structures can be resolved at the highest resolution of 2.3Å.

4.5 CTF Correction

Recall that the CTF is a function that accounts for the imperfectness of images due to the microscope lens. Ideally, the goal is to correct the image by dividing by the CTF; however, from the graphs above, there are many points during which the CTF is 0, and a division by zero error would be received. To correct this, it was first thought to just flip the phases. This would get rid of the zero terms, since all the terms would be either all positive or negative, however differentiating between what is considered high and low contrast to be difficult.

The most current, common correction is called the *Wiener filter*. This is given as[9],

$$\frac{H(\mathbf{k})}{H(\mathbf{k})^2 + N^2} \tag{4.9}$$

Instead of dividing by the plain old contrast transfer function, the image is corrected by being divided by the contrast transfer function $H(\mathbf{k})^2$ and the noise intensity N^2. Now, when the division when $H(\mathbf{k})$ is zero is no longer a problem since N^2 is *never* zero. Therefore, to retrieve the accurate information of the atoms of the sample, the image needs to be *deconvoluted* which is gone over in Eq. 3.14.

4.6 Applications

Knowledge of the contrast transfer function is widely used by any lab that deals with high resolution electron microscopy or cryogenic electron microscopy. All the factors that may have disturb achieving the most accurate representation of the sample is embedded in the CTF. Hence, naturally, investigators would desire to study the CTF and use it to account for any abnormalities in an image. One significant of almost any structure that has use the corrected with the CTF would be the structure of the 70S *Escherichia coli* Ribosome from the Frank lab in the *Journal of Structural Biology*.

5 The Code

The first function that has to be written is `project_fst(mol, R)}`. What `project_fst(mol, R)` does is it produces an image of the virus, ρ, from each frame, F. This function takes in the virus, represented as a 3D array (mol), and a frame, represented as a 3×3 rotation matrix (R), and simulates an image of the virus from each viewing direction as a 2D array. This function is implemented by appealing to the Fourier Slice Theorem.
Recall that there were two ways to resolve the 3D structure of ρ. This function back project takes in n 2D images $\{I_i\}$ and their associated microscope orientations (R), and returns 3D array that should represent a rough approximation of ρ, the virus. This method is called the weighted, filtered back projection.

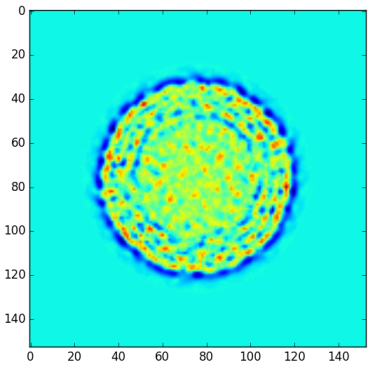
5.1 project fst

```
import numpy as np
import matplotlib as plt
from scipy.interpolate import RegularGridInterpolator


def project_fst(mol, R):
    """
    :param mol: an NxNxN array that contains rho
    :param R: the rotation matrix to use for the image projection
    :return: an image projected using the rotation matrix R and the data in mol
    """
    N = mol.shape[0]
    N_range = np.linspace(-1, 1, N)
    inter = RegularGridInterpolator((N_range, N_range, N_range), mol,
        method='linear', bounds_error=False, fill_value=0)
    pos = np.concatenate((
        N_range.reshape(N, 1, 1, 1) * np.ones(N * N).reshape(1, N, N, 1),
        N_range.reshape(1, N, 1, 1) * np.ones(N * N).reshape(N, 1, N, 1),
        N_range.reshape(1, 1, N, 1) * np.ones(N * N).reshape(N, N, 1, 1)),
        axis=3)
    # pos gives a matrix of size N which at [x,y,z] returns a value which we can
        transform into the image plane space
    r_mol = inter(np.dot(pos, R))
    return np.real(np.fft.ifft2(np.fft.fftn(r_mol)[:, :, 0]))
```

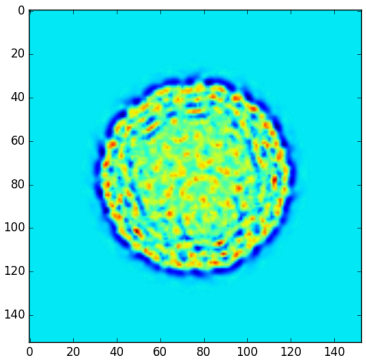
5.2 Testing `project fst`

```
from MRCFile import MRCFile
from sim_image import project_fst
import matplotlib.pyplot as plt
from vanheel import random_rotation_matrix
zika_153 = MRCFile('zika_153.mrc')
zika_153.load_all_slices()
mol = zika_153.slices
# image = project_fst(mol, np.eye(3))
image = project_fst(mol, random_rotation_matrix())
plt.imshow(image)
plt.show()
```

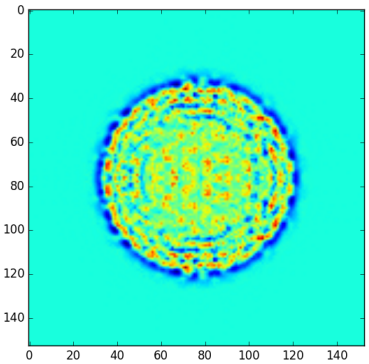
We know the shape of the Zika virus is roughly spherical, so we believe these images are logical output. The MRC file used to test is zika153. Figures (a) and (b) are projections using a random



(a)



(b)



(c)

rotation matrix. Figure (c) is slightly different, because the interpolation is done on exact array indices, using a rotation matrix that is the identity matrix. In other words, no real interpolation of the data is done.

5.3 back projection

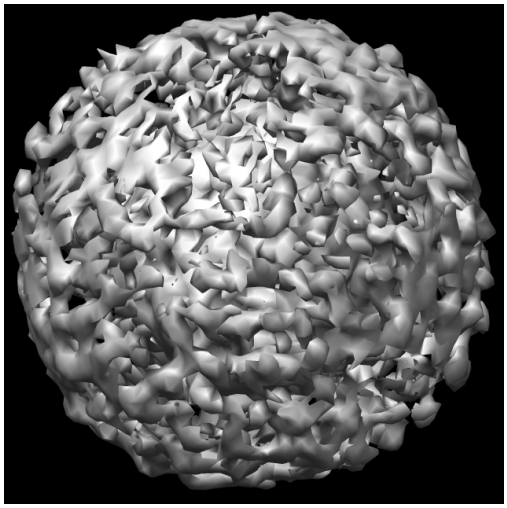
```
import numpy as np
from scipy.interpolate import RegularGridInterpolator


def compute_h(rot, D):
    """
    :param rot: the rotation matrix for a particular image
    :param D: the size of the rho
    :return: the component of h for this image's rotation matrix
    """
    c_vec = rot[:, 2]
    # c_vec is 3rd column of rotation matrix

    pos = np.concatenate((
        np.arange(D).reshape(D, 1, 1, 1)*np.ones(D*D).reshape(1, D, D, 1),
        np.arange(D).reshape(1, D, 1, 1)*np.ones(D*D).reshape(D, 1, D, 1),
        np.arange(D).reshape(1, 1, D, 1)*np.ones(D*D).reshape(D, D, 1, 1)),
        axis=3)
    # pos[x,y,z] = [x,y,z], returns its own index
    return np.sum(D*np.sinc(D * np.dot(pos, c_vec)))


def back_project(data, use_filter=True, D_percent=.5):
    """
    :param data: tuples containing images as np arrays and their rotations
    :param use_filter: whether or not to apply to transfer function filter
    :param D_percent: what percent of the total 'box' does the rho take up
    :return:
    """
    N = data[0][0].shape[0]
    r = np.zeros(N)
    D = int(D_percent*N)
    r[D/2:N-(D/2)] = 1
    s = np.fft.fftshift(np.fft.fft(r))
    # r is the rect function
    s = np.tile(s[np.newaxis, np.newaxis, :], (N, N, 1))
    B = np.zeros((N, N, N))
    if use_filter:
        H = 0
    for image, R in data:
        I_hat = np.fft.fftshift(np.fft.fft2(image))
        I_hat = np.tile(I_hat[..., np.newaxis], (1, 1, N))
        image = np.real(np.fft.ifftn(np.fft.ifftshift(I_hat*s)))
        # rotate and interpolate
        N_range = np.linspace(-1, 1, N)
        inter = RegularGridInterpolator((N_range, N_range, N_range),
            image, method='linear', bounds_error=False, fill_value=0)
```

```
48          pos = np.concatenate((
49              N_range.reshape(N, 1, 1, 1) * np.ones(N * N).reshape(1, N, N, 1),
50              N_range.reshape(1, N, 1, 1) * np.ones(N * N).reshape(N, 1, N, 1),
51              N_range.reshape(1, 1, N, 1) * np.ones(N * N).reshape(N, N, 1, 1)),
52              axis=3)
53          # pos gives a matrix of size N which at [x,y,z] returns a vector which we
                can transform from the image plane space
54          B += inter(np.dot(pos, R.transpose()))
55          # we use R.transpose to transform all the vectors away from image plane
                space to the original
56          if use_filter:
57              H += compute_h(R, D)
58      if use_filter:
59          # H is defined in fourier space
60          B_hat = np.fft.fftshift(np.fft.fftn(B))
61          return np.real(np.fft.ifftn(np.fft.ifftshift(B_hat/H)))
62      else:
63          return B
```

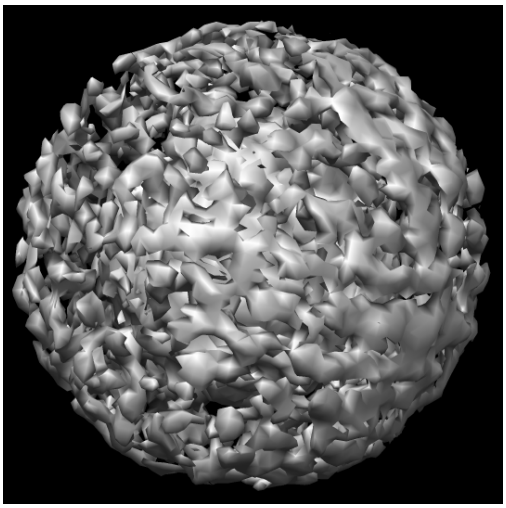
5.4 Testing `back projection`

```
1  from back_projection import back_project
2  from sim_image import project_fst
3  from MRCFile import MRCFile
4  from vanheel import random_rotation_matrix
5  import numpy as np
6  import sys
7
8  if len(sys.argv) != 3:
9      print "please call with an mrc file as argument 1, and number of rotations as
           argument 2"
10 else:
11     f = MRCFile(sys.argv[1])
12     f.load_all_slices()
13     images = []
14     images_noisy = []
15     Rs = [random_rotation_matrix() for i in xrange(int(sys.argv[2]))]
16     # TODO here we can make many images later
17     for R in Rs:
18         image = project_fst(f.slices, R)
19         images.append(image)
20         noise = np.random.normal(0.0, 20*np.mean(image), image.shape)
21         images_noisy.append(image + noise)
22     data = zip(images, Rs)
23     noisy_data = zip(images_noisy, Rs)
24     f.slices = back_project(data, use_filter=False)
25     f.write_file('#nofilter.mrc', overwrite=True)
26     f.slices = back_project(data)
```

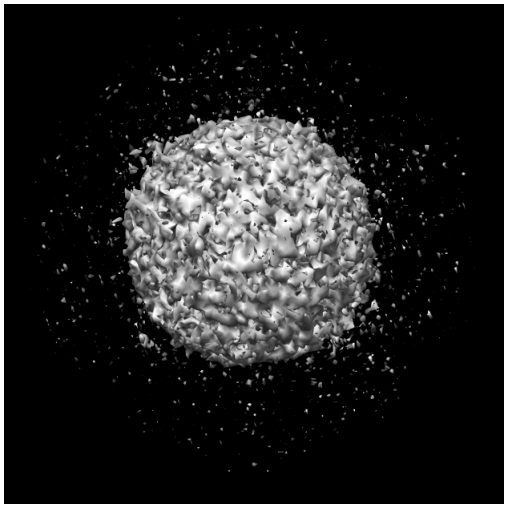
```
27    f.write_file('filter.mrc', overwrite=True)
28    f.slices = back_project(noisy_data, use_filter=False)
29    f.write_file('#nofilter_noise.mrc', overwrite=True)
30    f.slices = back_project(noisy_data)
31    f.write_file('filter_noise.mrc', overwrite=True)
```



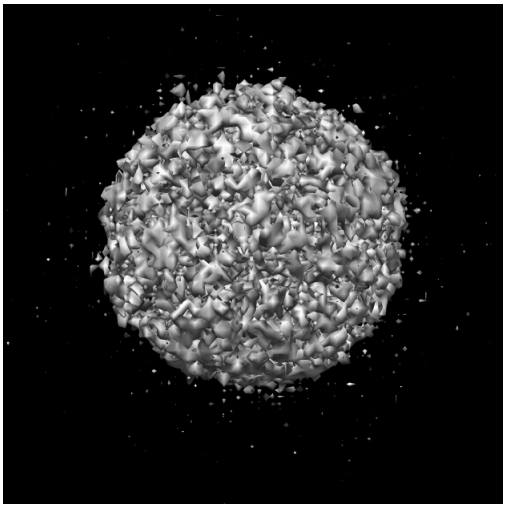
(d) No noise, without filter



(e) No noise, with filter



(f) Noise, without filter



(g) Noise, with filter

Noise is additive Gaussian noise with standard deviation is 20 times the average value of each image. In (g) we can see how the filter reduces noise compared to (f). The filter also doesn't ruin the image when there is no noise. This can be seen in the lack of difference between (d) and (e).

5.5 `common lines`

Unfortunately, we didn't get common lines to work. Our general approach was to make an interpolator object for every image and then for every combination of two images, compute their common lines. We found common lines by looking across many $0 \leq \theta \leq 2\pi$ given by a parameter to the function and using the interpolators to see which θ corresponds to the best common line for each of the two image combinations. We then put the output of the function into the *Van Heel algorithm.*

Notes

[1]In a fancy, albeit concise terms, the set of all rotational matrices in $\mathbb{R}^3$ is denoted by $\mathfrak{so}(3)$. It has the structure of what are called Lie Groups. For more information on Lie Groups see [5].

[2]The Radon projection is an important machinery in Math that has its use in especially in Probability Theory and Harmonic Analysis.

[3]Specifically, the input function ρ would need to be thrice integrable.

[4]This is the well known δ-basis decomposition of a function.

[5]This transformation is only valid if the system is shift invariant, or *isoplantic*. However, we shall brush over that here.

[6]Although the method of Radon Inversion is much faster and elegant, it is not widely used for reasons unknown to the authors. See [7, p. 262] for more detail.

[7]For more information please visit `http://www.maxsidorov.com/ctfexplorer/science/index.htm`

[8]See `http://www.directelectron.com/wp-content/uploads/carbonfilm_thonrings.jpg`

Appendix

A Time and Frequency Domains

One of the most fundamental laws of motion in Newtonian Physics says that the distance covered by an object moving with fixed velocity v, at time t equals $v \cdot t$. In other words, if $s(t)$ is a function that returns distance covered at time t, then

$$s(t) = vt \tag{A.1}$$

A function such as this is said to be expressed in the time domain because it takes in time as its argument. Waves, for instance are also often expressed in the time domain as periodic functions. Take $f(t) = \sin(2\pi 4t)$ as an example. If its graph were to be drawn, the horizontal axis would

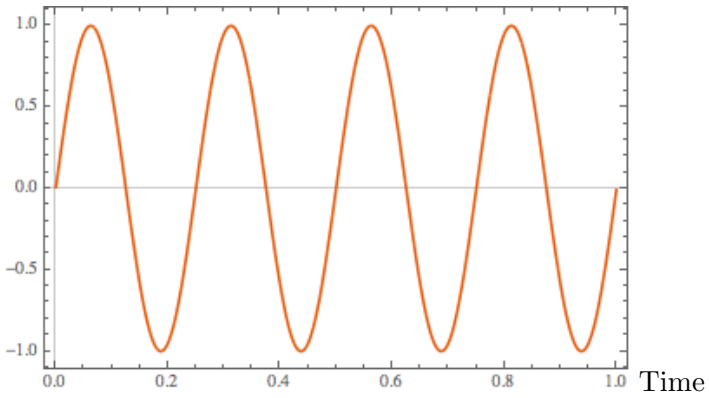


Figure A.1: $f(t) = \sin(2\pi 4t)$

represent time while the vertical $y-$axis could represent energy, or pressure(in case of sound waves) or have any physical meaning depending on the situation (see Fig A.1).

Notice that the sine wave in Fig A.1 oscillates at a frequency 4Hz with amplitude 1. Thus, any

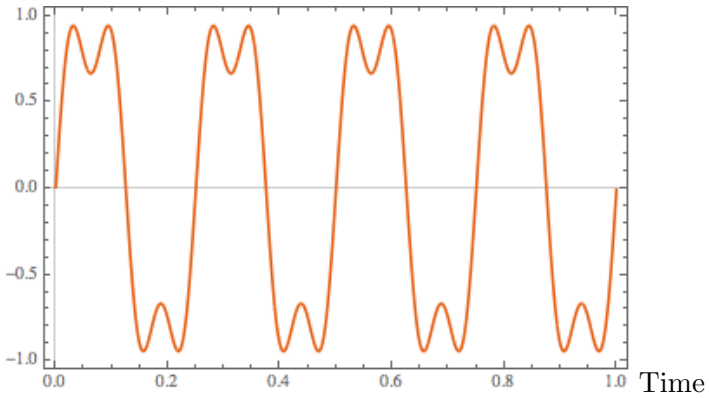


Figure A.2: $g(t) = \sin(2\pi 4t) + \frac{1}{3}\sin(2\pi 12t)$

sinusoidal wave can be described just by its frequency and amplitude. Naturally, we can extend this to linear combinations of sinusoidal waves as well. For instance, let $g(t) = \sin(2\pi 4t) + \frac{1}{3}\sin(2\pi 12t)$. (see Fig A.2).This periodic function is comprised of 2 sinusoidal waves, one with frequency 4Hz and amplitude of 1, while the other with frequency 12Hz and amplitude $\frac{1}{3}$. So instead of working in the time domain like we have been doing so far, why don't we forget about the time data and use only the frequency and amplitude data because as we have seen, they completely describe g. This is what shifting from a time domain to frequency domain means. Instead of representing a wave as a function of time, we represent it as a function of frequency. The graph of g in the frequency domain ($\hat{g}(\nu)$) therefore would look something like: But surely this is not possible for all waves?

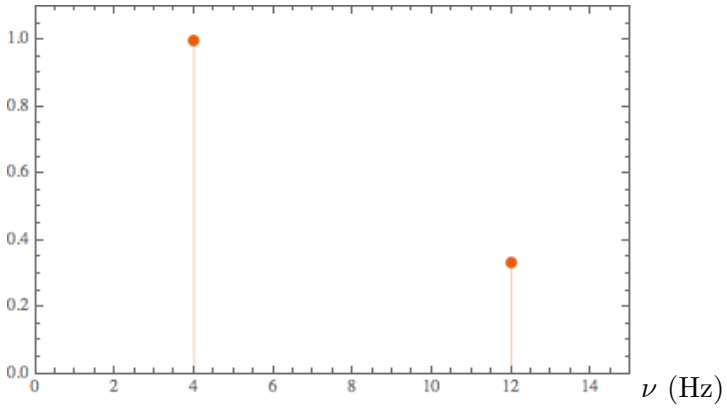


Figure A.3: $\hat{g}(\nu)$

That would imply that all periodic functions are just linear combination of sinusoidal waves. Joseph Fourier and him contemporaries in the 1600s proved exactly this. They showed that any periodic function can be represented as a superposition of sinusoidal waves (possibly infinitely many). Thus shifting from the time to frequency domain would be possible for any periodic function. This was later extended to all integrable functions. We ignore the specifics of the functions involved.
Fourier's method of transforming from the time to the frequency domain is called Fourier Transform and it is given by:

$$\mathfrak{F}\{f\}(\nu) = \int_{-\infty}^{\infty} f(t)\exp(-2\pi i\nu t)dt \tag{A.2}$$

Like the Radon Transform(see Eq. 2.1), it is an integral transform. The Fourier Transform is an invertible integral transform and its inverse Fourier Transform is given by:

$$\mathfrak{F}^{-1}\{f\}(t) = \int_{-\infty}^{\infty} f(\nu)\exp(2\pi it\nu)d\nu \tag{A.3}$$

It shouldn't be too difficult to show that $\mathfrak{F}^{-1}$ is actually the inverse transform of $\mathfrak{F}$.

Fourier Series and Fourier Transforms

The Fourier transform can also be looked as a change of basis within the space of all functions[9] from the δ basis (as discussed in Eq.3.7) to the exponential basis (which corresponds to a frequency

in the frequency domain)[10].
In a discrete setting(not when the function is over a discrete time domain), the Fourier series works similarly to the Fourier Transform. The Fourier series expands a periodic function f in terms of an infinite sum of sines and cosines. The vector space we are working with is the space of square integrable functions($L_2(-\frac{T}{2}, \frac{T}{2})$) in $[-\frac{T}{2}, \frac{T}{2}]$ and the inner product of two functions in the vector space is given by

$$\langle f, g \rangle = \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)g(x)dx$$

Notice that $\langle \sin x, \cos x \rangle = 0$ with respect to the above inner product. Also notice that the following set if vectors for $L_2(-\frac{T}{2}, \frac{T}{2})$

$$\left\{ \frac{T}{2}, \frac{2}{T}\cos\left(nx\frac{2\pi}{T}\right), \frac{2}{T}\sin\left(nx\frac{2\pi}{T}\right) \middle| n = 1, 2, \dots \right\}$$

Given a function $f(x)$, we can decompose it in the above Fourier series basis to get what is called the 'Fourier Series' of f. More specifically:

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + b_n \sin(nx) \tag{A.4}$$

and linear algebra tells us

$$a_0 = \left\langle f, \frac{T}{2} \right\rangle = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)dx \tag{A.5}$$

$$a_n = \left\langle f, \frac{2}{T}\cos\left(nx\frac{2\pi}{T}\right) \right\rangle = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)\cos\left(\frac{2\pi}{T}nx\right)dx \tag{A.6}$$

$$b_n = \left\langle f, \frac{2}{T}\sin\left(nx\frac{2\pi}{T}\right) \right\rangle = \frac{1}{\pi}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)\sin\left(\frac{2\pi}{T}nx\right)dx \tag{A.7}$$

The following worked out example while being a cool application of the Fourier series, should make things clear:

Example. Compute the Fourier series for $f(x) = x^2$, periodic on $[\pi, \pi]$ and use it to derive the identity

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} = -\frac{\pi^2}{12}$$

Solution. We know that for any given (periodic) $f(x)$,

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + b_n \sin(nx) \tag{A.8}$$

as a limit. In this case,

$$a_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)dx = \frac{1}{2\pi}\int_0^{\pi} x^2 dx = \frac{\pi^2}{6} \tag{A.9}$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)dx = \frac{1}{\pi}\int_0^{\pi} x^2\cos(nx)dx = 2\frac{(-1)^n}{n^2} \tag{A.10}$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)dx = \frac{1}{\pi}\int_0^{\pi} x^2\sin(nx)dx \tag{A.11}$$

Putting them all together,

$$f(x) = \frac{\pi^2}{6} + 2\sum_{n=1}^{\infty}\frac{(-1)^n}{n^2}\cos(nx) + \frac{1}{\pi}\sin(nx)\int_0^{\pi} x^2\sin(nx)dx \tag{A.12}$$

Evaluating at zero yields exactly the equation we want:

$$\cancelto{0}{f(0)} = \frac{\pi^2}{6} + 2\sum_{n=1}^{\infty}\frac{(-1)^n}{n^2} + 0 \tag{A.13}$$

implying

$$\sum_{n=1}^{\infty}\frac{(-1)^n}{n^2} = -\frac{\pi^2}{12} \tag{A.14}$$

B Proof of the Fourier Slice Theorem and Convolution Theorem

As promised, in this appendix we prove Theorem 2.1 (Fourier Slice Theorem) and Theorem 3.1 (Convolution Theorem). We begin with the Fourier Slice Theorem. Recall

Theorem B.1. (Projection/Fourier Slice Theorem) Suppose F is a viewing direction and $P_F = \text{span}\{\vec{a}, \vec{b}, \vec{c}\}$ is the image plane of F, then

$$\underbrace{\mathfrak{F}\{\rho\}|_{P_F}}_{\substack{\text{Fourier transform}\\ \text{restricted to } P_F}} = \underbrace{\mathfrak{F}\{I_F\}}_{\substack{\text{2D Fourier}\\ \text{transform of } I_F}} \tag{B.1}$$

Proof. Assume $F = \text{id} \in \mathfrak{so}(3)$. Then,

$$\mathfrak{F}\{\rho\}(h,k,l) = \int_{\mathbb{R}^3}\rho(x,y,z)\exp\left(-2\pi i(hx+ky+lz)\right)\,dx\,dy\,dz$$

Since $P_F = \text{span}(\vec{a}, \vec{b})$,

$$\begin{aligned}
\mathfrak{F}\{\rho\}(h,k,l)|_{P_F} &= \mathfrak{F}\{\rho\}(h,k,0) \\
&= \int_{\mathbb{R}^3} \rho(x,y,z) \exp\left(-2\pi i(hx+ky)\right)\, dxdydz \\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} \underbrace{\int_{\mathbb{R}} \rho(x,y,z)dz}_{\text{Fubini's Theorem}}\, \exp\left(-2\pi i(hx+ky)\right)\, dxdy \\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} I_F(x,y) \exp\left(-2\pi i(hx+ky)\right)\, dxdy \\
&= \mathfrak{F}\{I_F\}(h,k)
\end{aligned}$$

□

Next up we shall prove the Convolution Theorem (Theorem 3.1). In fact this was one of the homework problems. Again, to recall

Theorem B.2. (*Convolution Theorem*) Let $f, g : \mathbb{R}^n \to \mathbb{R}$ with $\left|\int_{\mathbb{R}^n} f(\mathbf{x})dx\right| < \infty$ and $\left|\int_{\mathbb{R}^n} g(\mathbf{x})dx\right| < \infty$. The convolution of f and g, denoted $f \star g$ is given by

$$(f \star g)(\mathbf{z}) = \int_{\mathbb{R}^n} f(\mathbf{x})g(\mathbf{z}-\mathbf{x})d\mathbf{x}$$

If $\mathfrak{F}(f)$ is the Fourier Transform of f, then

$$\mathfrak{F}(f \star g) = \mathfrak{F}(f) \cdot \mathfrak{F}(g), \tag{B.2}$$

where $\cdot$ denotes point-wise multiplication of the functions.

Proof. We will prove this for $n = 1$ only. Proofs for higher dimensions are similar but likely involve a lot more book-keeping. By definition of convolution

$$\begin{aligned}
f \star g &= \int_{-\infty}^{\infty} f(x)g(z-x)dx \\
&= \int_{-\infty}^{\infty} f(x) \int_{-\infty}^{\infty} \mathfrak{F}\{g\}(\xi) \exp\left(2\pi i\xi(z-x)\right) d\xi dx \\
&= \int_{-\infty}^{\infty} f(x) \int_{-\infty}^{\infty} \mathfrak{F}\{g\}(\xi) \exp\left(2\pi i\xi z\right) \exp\left(-2\pi i\xi x\right) d\xi dx \\
&= \int_{-\infty}^{\infty} \mathfrak{F}\{g\}(\xi) \int_{-\infty}^{\infty} f(x) \exp\left(-2\pi i\xi x\right) dx \cdot \exp\left(2\pi i\xi z\right) d\xi \qquad \text{(Fubini's Theorem)} \\
&= \int_{-\infty}^{\infty} \mathfrak{F}\{g\} \cdot \mathfrak{F}\{f\} \exp\left(2\pi i\xi z\right) dx \\
&= \mathfrak{F}^{-1}\left\{\mathfrak{F}\{g\} \cdot \mathfrak{F}\{f\}\right\} \\
&= \mathfrak{F}^{-1}\left\{\mathfrak{F}\{f\} \cdot \mathfrak{F}\{g\}\right\} \qquad \left(\begin{smallmatrix}\text{pointwise multiplication}\\ \text{is commutative}\end{smallmatrix}\right)
\end{aligned}$$

Applying $\mathfrak{F}$ to both sides gives $\mathfrak{F}(f \star g) = \mathfrak{F}\{f\} \cdot \mathfrak{F}\{g\}$ □

6 References

[1] Frank, J. *Three-dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State.* Oxford: Oxford UP, 2006. Print.

[2] Goodman, J. W. *Introduction to Fourier Optics.* New York: McGraw-Hill, 1968. Print

[3] Griffel, D. H. *Applied Functional Analysis.* Chichester, W. Sussex: E. Horwood, (1981) Print.

[4] Intaraprasonk, Varat, Huolin L. Xin, and David A. Muller. "Analytic Derivation of Optimal Imaging Conditions for Incoherent Imaging in Aberration-corrected Electron Microscopes." *Ultramicroscopy* 108.11 (2008): 1454-466. Web. 28 Apr. 2016.

[5] Lee, John M. "Introduction to Smooth Manifolds. (Graduate Texts in Mathematics).", New York, NY: Springer., (2013) Print.

[6] Liao, M., E. Cao, D. Julius, and Y. Cheng. "Structure of TRPV1 Ion Channel Determined by Single Particle Electron Cryo-microscopy." *Nature* 504 (2013): 107-112. Web.

[7] Rademacher, M. *Weighted back-projection methods.* In: Frank, J. (ed.) *Electron Tomography: Methods for Three Dimensional Visualization of Structures in the Cell,* 2nd Ed, pp. 245274. New York: Springer, 2006. Print

[8] Williams, D. B. and Carter, C. B. "Transmission electron microscopy: A textbook for materials science." New York, NY: Plenum Press., (1996) Print.

[9] Zhu, Jun, Pawel A. Penczek, Rasmus Schrder, and Joachim Frank. "Three-Dimensional Reconstruction with Contrast Transfer Compensation from Defocus Series." *Journal of Structural Biology* 118.3 (1997): 197-219. Web.

Bidit Acharya, University of California, Berkeley
email, B. Acharya: `bidit@berkeley.edu`

Tracy. Lou, University of California, Berkeley
email, T. Lou: `tlou@berkeley.edu`

Joshua P. Nixon, University of California, Berkeley
email, J.P. Nixon: `nixonpjoshua@berkeley.edu`

Alexander Pearson, University of California, Berkeley
email, A. Pearson: `apearson1995@berkeley.edu`

Last updated: Tuesday 10th May, 2016 at 23:36