PHASE RECOVERY AND INTERFEROMETRY - 23S

IAN GILL AND HENRY MORRIS
MARCH-JUNE 2023

1. Introduction

[AG: Ask Dylan to help put in a bibliography. I am going to leave the introduction for now and focus on other aspects of this work.] In applications such as synthetic aperture radar (SAR), the underlying signals are complex-valued. Many SAR imaging algorithms are designed to reconstruct the magnitude of the underlying image, mainly because it is more straightforward to do so (phase returns can lack meaningful structure) and the signal magnitude does provide useful information regarding target classification. However coherent imaging, that is, the recovery of both the magnitude and phase of the underlying scene, is useful in downstream processing. For example, interferometric height estimation and coherent change detection both require phase information. We note that while this discussion is inspired by SAR imaging, other coherent imaging modalities, such as ultrasound, are affected by similar issues and the techniques discussed here may apply.

The acquired phase history data in SAR can be modeled as (non-uniform) Fourier samples. The goal of this research is to combine the elements of Ian's work on jump approximation to the current methods for recovering the phase information in interferometry (the recovery of phase from a complex signal given a finite number of its Fourier coefficients). We hope that by better identifying where the edges of height jumps in a SAR image (an issue that may result in errors), we can improve in accuracy the collection of height information of the target image

To recover phase information of a complex signal given a finite number of its Fourier coefficients, we first simplify the problem. Specifically, we reduce the order and model the phase function as linear over the domain of interest. We then describe a more general approach to include other phase functions. As we will show, it is possible to identify using Ian's algorithm the edge detection in linear phase complex valued signals. Given that linear chirps, which are used in SAR, have quadratic phase functions, taking the derivative should allow us to apply Ian's algorithm In the paper, after we detail the phase recovery process for a linear phase function, we demonstrate that this approach is useful in recovering better estimates for jump information (both location and height difference) for a finite number of jumps in the magnitude of the signal.

For these applications, we look to recover information about the underlying signal from the finite number of Fourier samples we are given. For complex signals, we aim to recover both the phase information and jump information for a signal that is piecewise smooth in the magnitude. When the underlying image or signal is real and piecewise smooth, there will be Gibbs phenomenon, where oscillations and the error between the Fourier reconstruction and the underlying signal increase close to the jumps of the signal. This is a well-known phenomenon and there are many ways to mitigate it, as documented in [list papers here].

For many applications, there is often a reason to want to know both jump locations and jump heights, and there are methods for real signals to go directly from Fourier data to edge location and edge heights. The concentration factor edge detection method in [Viswanathan, Gelb, and Cochran 2011] is a band pass filtering method that does this. However, in applications like SAR, the underlying signal is complex, where the magnitude has the piecewise smoothness property but

the phase is typically smooth. In applications like interferometry, which will be discussed later, the phase difference is often a linear function.

Using the concentration factor method unadjusted on a complex signal directly does not work because of the imaginary portion. It is possible to discard the phase by using the concentration factor method on only the magnitude of the reconstruction of the signal. However, it is useful to retain phase information for other methods such as coherent change detection. Also, as will be demonstrated later in these notes, the concentration factor method applied solely to the magnitude of the reconstruction of the signal results in less accurate estimates of the jump location and jump heights than we would like.

While there are ways to retain crude phase estimates, these notes will show that it is possible to modify the concentration factor method so that it recovers edge information in the magnitude while maintaining phase. Indeed, the edge information recovered through this adjusted method is a better estimate than what is recovered through the concentration factor method applied to the magnitude alone. In the paper below, we attempt to show this hypothesis.

Ultimately, the importance of this research stems from the numerous places where complex signals need to be unwrapped in the real world. They are commonly encountered in areas such as telecommunications, radar systems, medical imaging, to name just a few fields. Much has been done in this regard for real signals, but not in the complex case. For our focus, we will be looking at the finite number of Fourier samples taken in SAR, but it should be noted our conclusions can be applied elsewhere.

2. Preliminaries

In this section, we provide notation and definitions that will be used throughout the rest of the notes as well as a background of the concentration factor method defined in [Viswanathan, Gelb, and Cochran 2011].

2.1. Notation and Definitions.

Before introducing our approach to recover edges in a complex signal that is piecewise smooth in the magnitude and smooth in the phase, we provide a list of notations and background definitions that will be used throughout the notes. We use bold typeface and lowercase letters to denote vectors (for example, $\mathbf{x} \in \mathbb{C}^N$) and capital letters in conventional typeface to denote matrices (for example, $M \in \mathbb{C}^{M\times N}$). We use $|\cdot|$ to denote (componentwise) absolute value and / to denote componentwise division.

We now provide some definitions which will be used in the rest of this document.

Definition 1. (*Periodic Function*) A function $f : \mathbb{R} \to \mathbb{C}$ is said to be *periodic with period* τ, or τ-periodic, if

$$f(x+\tau) = f(x), \qquad \forall x \in \mathbb{R}.$$

Since $f \in \mathbb{C}$, it is useful to distinguish between its phase and magnitude, which are defined as:

Definition 2. (*Phase and Magnitude*) The *phase* of $z \in \mathbb{C}$ is the complex exponential $e^{i\theta}$ such that $z = |z|e^{i\theta}$, where $|z|$ is the *magnitude* or *modulus* of z and $\theta \in \mathbb{R}/[0, 2\pi)$. Similarly, for a function $f : \mathbb{R} \to \mathbb{C}$, we write

For a vector $\mathbf{x} \in \mathbb{C}^N$, the *vector of phases* is $\mathbf{x}/|\mathbf{x}| = \begin{bmatrix} e^{i\theta_0} & e^{i\theta_1} & \cdots & e^{i\theta_{N-1}} \end{bmatrix}^T$. [AG: This is not good notation, since $\mathbf{m}x/|\mathbf{m}x|$ typically signifies a normalized vector.

Definition 3. ($L^2(a,b)$) We say $f : \mathbb{R} \to \mathbb{C}$ is in $L^2(a,b)$ if $\int_a^b |f(x)|^2 dx < \infty$.

Definition 4. (Fourier series expansion) If $f \in L^2(-\pi, \pi)$, then f can be written as

$$f = \sum_{k \in \mathbb{Z}} \widehat{f}_k e^{ikx}, \tag{1}$$

where the coefficients $\widehat{f}_k$ are defined as

$$\widehat{f}_k := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx, \qquad k \in Z. \tag{2}$$

Definition 5. (Partial Fourier Sum) We define the Nth *partial Fourier sum* of $f(x)$, $x \in [-\pi, \pi)$ to be

$$S_N f(x) = \sum_{|k| \leq N} \widehat{f}_k e^{ikx}. \tag{3}$$

It is well known that if f is a periodic analytic function then $S_N f(x) \to f(x)$ exponentially fast. need citation here.

In this work we are concerned with functions that are piecewise-smooth. Say something about this means that there is the Gibbs phenomenon, with the ringing artifact near the discontinuities and slow convergence everywhere else in the domain. Thus we need

Definition 6. (*Jump Function, Locations and Values*) Let $f : \mathbb{R} \to \mathbb{C}$ be a piecewise-smooth function. We define the *jump function* of f, denoted as $[f]$, to be

$$[f](x) := f(x^+) - f(x^-), \quad x \in \mathbb{R}, \tag{4}$$

where $f(x^+)$ and $f(x^-)$ denote the right- and left-hand limits of f at x respectively. We define the *jump locations* (or just jumps) in f to be the set $J = \{x \in \mathbb{R} \,|\, [f](x) \neq 0\}$ and the corresponding *jump values* to be $\{[f](x_j)\}_{j \in J}$. We use the terms "jump", "edge", and "discontinuity" interchangeably.

2.2. The concentration factor edge detection method.

Here you can put the traditional edge detection method for real functions, then show why it doesn't work for complex functions.

3. Concentration Factor Method on Complex Function

3.1. Derivation of Fourier Coefficients.

You should write something to the effect of that you are considering a single jump discontinuity for ease of presentation. It is not completely WLOG, when it comes to determining the concentration factor sum, but it is WLOG for the integration by parts. Without loss of generality, we begin by looking at an underlying signal that has a single jump discontinuity in the magnitude for ease of presentation. As will be shown below when we extend this application to an underlying signal with two jump discontinuities in the magnitude, this derivation by integration by parts does not depend on the number of jump discontinuities as long as the jump discontinuities are sufficiently distanced from each other. Let $f(x) = g(x)e^{icx}$ such that $g(x) : \mathbb{R} \to \mathbb{R}$ and $c \notin \mathbb{Z}$ be the underlying function with a single jump in magnitude only at $x = \zeta$. Therefore, $g(\zeta^-) \neq g(\zeta^+)$, but $e^{ic\zeta^-} = e^{ic\zeta^+} = e^{ic\zeta}$. Given the Fourier coefficients in (2), we split the integral and then use integration by parts to obtain

$$\begin{aligned}
\widehat{f}_k &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx \\
&= \frac{1}{2\pi} \int_{-\pi}^{\zeta^-} f(x) e^{-ikx} dx + \frac{1}{2\pi} \int_{\zeta^+}^{\pi} f(x) e^{-ikx} dx \\
&= \frac{1}{2\pi} \int_{-\pi}^{\zeta^-} g(x) e^{-i(k-c)x} dx + \frac{1}{2\pi} \int_{\zeta^+}^{\pi} g(x) e^{-i(k-c)x} dx
\end{aligned}$$

$$= \frac{1}{2\pi}\left(\frac{1}{i(k-c)}\left(g(\zeta^+)-g(\zeta^-)\right)e^{-i(k-c)\zeta} - \frac{1}{i(k-c)}\left(g(\pi)e^{-i(k-c)\pi} - g(-\pi)e^{-i(k-c)(-\pi)}\right)\right.$$
$$\left. + \frac{1}{i(k-c)}\int_{-\pi}^{\zeta^-} g'(x)e^{-.i(k-c)x}dx + \frac{1}{i(k-c)}\int_{\zeta^+}^{\pi} g'(x)e^{-i(k-c)x}dx\right).$$

Due to periodicity, the second term on the right hand side is 0. By continued use of integration by parts we obtain

$$\widehat{f}_k = \frac{e^{-i(k-c)\zeta}}{2\pi}\left(\frac{1}{i(k-c)}[g](\zeta) + \left(\frac{1}{i(k-c)}\right)^2[g'](\zeta) + \left(\frac{1}{i(k-c)}\right)^3[g''](\zeta) + \dots\right)$$
$$= \frac{e^{-i(k-c)\zeta}}{2\pi}\sum_{j=0}^{\infty}\left(\frac{1}{i(k-c)}\right)^{j+1}\left[g^{(j)}\right](\zeta).$$

Then, using $\widehat{f}_k$ as calculated, we get the partial Fourier summation with respect to the jump functions:

$$S_N^\sigma[f](x) = \sum_{|k|\le N}\widehat{f}_k\left(i\mathrm{sgn}(k)\sigma\left(\frac{|k|}{N}\right)\right)e^{ikx}$$
$$= \frac{[g](\zeta)e^{ic\zeta}}{2\pi}\sum_{|k|\le N, k\neq c}\frac{\sigma\left(\frac{|k|}{N}\right)\mathrm{sgn}(k)}{k-c}e^{ik(x-\zeta)}$$
$$+ \frac{[g'](\zeta)e^{ic\zeta}}{2\pi}\sum_{|k|\le N, k\neq c}\frac{\sigma\left(\frac{|k|}{N}\right)\mathrm{sgn}(k)}{i(k-c)^2}e^{ik(x-\zeta)} + \dots$$

Here, we see a complication in the formula, as the constant c is unknown, and thus, if $c \in \mathbb{Z}$, we would not know which element $k = c$ in $|k| \le N$ should be excluded from the summation. This presents a possible area of further exploration, where $c \in \mathbb{Z}$, but for this reason we have defined $c \notin \mathbb{Z}$ for the purpose of this research.

We can similarly look at an underlying function with two jumps. Again, let $f(x) = g(x)e^{icx}$ be periodic on $(-\pi, \pi)$ and let the two jumps occur only in the magnitude, at points $x = \zeta_1$ and $x = \zeta_2$. Without loss of generality, let $\zeta_1 < \zeta_2$. We again are given Fourier data on the domain $(-\pi, \pi)$: $\{\widehat{f}_k\}_{k=-N,\dots,N}$, where $\widehat{f}_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}dx$:

$$\widehat{f}_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}dx$$
$$= \frac{1}{2\pi}\left(\int_{-\pi}^{\zeta_1^-} f(x)e^{-ikx}dx + \int_{\zeta_1^+}^{\zeta_2^-} f(x)e^{-ikx}dx\int_{\zeta_2^+}^{\pi} f(x)e^{-ikx}dx\right)$$
$$= \frac{1}{2\pi}\left(\int_{-\pi}^{\zeta_1^-} g(x)e^{-i(k-c)x}dx + \int_{\zeta_1^+}^{\zeta_2^-} g(x)e^{-i(k-c)x}dx\int_{\zeta_2^+}^{\pi} g(x)e^{-i(k-c)x}dx\right).$$

Using the same approach of integrating by parts infinitely and noting the periodic nature of $f(x)$, we get:

$$\widehat{f}_k = \frac{1}{2\pi}\left(\frac{1}{i(k-c)}\left(e^{-i(k-c)\zeta_1}[g](\zeta_1) + e^{-i(k-c)\zeta_2}[g](\zeta_2)\right)\right.$$

$$+\left(\frac{1}{i(k-c)}\right)^2\left(e^{-i(k-c)\zeta_1}[g'](\zeta_1)+e^{-i(k-c)\zeta_2}[g'](\zeta_2)\right)+\dots\Bigg)$$
$$=\frac{1}{2\pi}\sum_{j=0}^{\infty}\left(\frac{1}{i(k-c)}\right)^{j+1}\left(e^{-i(k-c)\zeta_1}[g^{(j)}](\zeta_1)+e^{-i(k-c)\zeta_2}[g^{(j)}](\zeta_2)\right)$$
$$=\frac{e^{-i(k-c)\zeta_1}}{2\pi}\sum_{j=0}^{\infty}\left(\frac{1}{i(k-c)}\right)^{j+1}[g^{(j)}](\zeta_1)+\frac{e^{-i(k-c)\zeta_2}}{2\pi}\sum_{j=0}^{\infty}\left(\frac{1}{i(k-c)}\right)^{j+1}[g^{(j)}](\zeta_2).$$

Again as before, we use $\widehat{f_k}$ to calculate the partial Fourier summation with respect to the jump functions:

$$S_N^\sigma[g](x)=\sum_{|k|\le N}\widehat{f_k}\left(i\mathrm{sgn}(k)\sigma\left(\frac{|k|}{N}\right)\right)e^{ikx}$$
$$=\frac{[g](\zeta_1)e^{ic\zeta_1}}{2\pi}\sum_{|k|\le N,k\neq c}\frac{\sigma\left(\frac{|k|}{N}\right)\mathrm{sgn}(k)}{k-c}e^{ik(x-\zeta_1)}$$
$$+\frac{[g](\zeta_2)e^{ic\zeta_2}}{2\pi}\sum_{|k|\le N,k\neq c}\frac{\sigma\left(\frac{|k|}{N}\right)\mathrm{sgn}(k)}{k-c}e^{ik(x-\zeta_2)}$$
$$+\frac{[g'](\zeta_1)e^{ic\zeta_1}}{2\pi}\sum_{|k|\le N,k\neq c}\frac{\sigma\left(\frac{|k|}{N}\right)\mathrm{sgn}(k)}{i(k-c)^2}e^{ik(x-\zeta_1)}$$
$$+\frac{[g'](\zeta_2)e^{ic\zeta_2}}{2\pi}\sum_{|k|\le N,k\neq c}\frac{\sigma\left(\frac{|k|}{N}\right)\mathrm{sgn}(k)}{i(k-c)^2}e^{ik(x-\zeta_2)}+\dots$$

Thus, we see that the partial Fourier summation works as well for two jumps in the magnitude that are sufficiently far apart as for one. We note that the global nature of the Fourier data will cause additional interfering oscillations in each jump discontinuity neighborhood, which causes issues if two jump discontinuities are not sufficiently far apart. In this research, we concern ourselves with underlying signals that have a finite number of jumps in the magnitude that are sufficiently distanced from each other. This is not quite true. The global nature of the Fourier data will cause additional interfering oscillations in each jump discontinuity neighborhood. If we assume that the jump discontinuities are located sufficiently far apart then it valid.

3.2. **Current Trigonometric Concentration Factor.** You could say that it doesn't really matter which one of the concentration factors you choose, so as a prototype you will use the trig one. This is where you will start to address (5) in the introduction list, that you can't just apply the standard concentration factor when the function is complex. As noted in 2.2 the trigonometric concentration factor as defined in Viswanathan, Gelb, Cochran (2011) in our partial Fourier summation. The trigonometric concentration factor is defined as:

$$\sigma_G(\eta)=\frac{\pi\sin(\pi\eta)}{\mathrm{Si}(\pi)},\quad\text{where}$$
$$\mathrm{Si}(\pi)=\int_0^\pi\frac{\sin(x)}{x}dx.$$

Thus, using the derivation from above for a single jump in the magnitude in the complex function $f(x) = g(x)e^{icx}$, we get:

$$\begin{aligned} S_N^\sigma[g](x) &= \sum_{|k|\le N} \widehat{f}_k \left(i\text{sgn}(k)\sigma\left(\frac{|k|}{N}\right)\right) e^{ikx} \\ &= \frac{[g](\zeta)e^{ic\zeta}}{2\pi} \sum_{|k|\le N, k\neq c} \frac{\frac{\pi \sin\left(\pi\left(\frac{|k|}{N}\right)\right)}{\text{Si}(\pi)}\text{sgn}(k)}{k-c} e^{ik(x-\zeta)} \\ &\quad + \frac{[g'](\zeta)e^{ic\zeta}}{2\pi} \sum_{|k|\le N, k\neq c} \frac{\frac{\pi \sin\left(\pi\left(\frac{|k|}{N}\right)\right)}{\text{Si}(\pi)}\text{sgn}(k)}{i(k-c)^2} e^{ik(x-\zeta)} + \dots \end{aligned}$$

Since $\sin(x)$ is an odd function, we note that

$$\begin{aligned} \sigma_G\left(\frac{|k|}{N}\right)\text{sgn}(k) &= \frac{\pi \sin(\pi\left(\frac{|k|}{N}\right)}{\text{Si}(\pi)}\text{sgn}(k) \\ &= \frac{\pi \sin(\pi\left(\frac{k}{N}\right)}{\text{Si}(\pi)} \\ &= \sigma_G\left(\frac{k}{N}\right). \end{aligned}$$

We can then simplify the partial Fourier summation:

$$\begin{aligned} S_N^\sigma[g](x) &= \frac{[g](\zeta)e^{ic\zeta}}{2\text{Si}(\pi)} \sum_{|k|\le N, k\neq c} \frac{\sin\left(\pi\left(\frac{k}{N}\right)\right)}{k-c} e^{ik(x-\zeta)} \\ &\quad + \frac{[g'](\zeta)e^{ic\zeta}}{2\text{Si}(\pi)} \sum_{|k|\le N, k\neq c} \frac{\sin\left(\pi\left(\frac{k}{N}\right)\right)}{i(k-c)^2} e^{ik(x-\zeta)} + \dots \end{aligned}$$

So here you can say that while the above works well if g is real (so that $c = 0$), you can easily see what the problem would be for $c \neq 0$. It would be good to put a picture in. Therefore we are motivated to modify the approach for complex signals.

3.3. **New Concentration Factor Derivation for Complex Functions.** We begin by following the approach layed out in Viswanathan, Gelb, Cochran (2011) to derive a new concentration factor for the signals of complex functions. Using the derivation of the partial Fourier coefficients $\hat{f}_k$ above, we then are looking to optimize the concentration factor $\sigma(\eta)$ to apply to the partial Fourier summation

$$S_N^\sigma[g](x) = \sum_{|k|\le N} \hat{f}_k \left(i\text{sgn}(k-c)\sigma\left(\frac{|k-c|}{N}\right)\right) e^{ikx}. \tag{5}$$

The parameters for $\sigma(\eta)$, as defined by Viswanathan, Gelb, Cochran, are:

(1) $\sum_{k=1}^{N} \sigma\left(\frac{k}{N}\right)\sin(kx)$ be odd
(2) $\frac{\sigma(\eta)}{\eta} \in C^2(0,1)$
(3) $\int_\epsilon^1 \frac{\sigma(\eta)}{\eta} \to -\pi, \epsilon = \epsilon(N) > 0$ being small.

Substituting the partial Fourier coefficients $\hat{f}_k$ of the signal for a complex function into $S_N^\sigma[g](x)$, we get:

(6)
$$S_N^\sigma[g](x) = \sum_{|k|\leq N} \left(\frac{e^{ic\zeta}}{2\pi} \left(\frac{[g](\zeta)}{i(k-c)} + \frac{[g'](\zeta)}{(i(k-c))^2} + \dots \right) e^{-ik\zeta} \right) i\sigma\left(\frac{|k|}{N}\right) \operatorname{sgn}(k) e^{ikx}$$

(7)
$$= e^{ic\zeta} \left(\frac{[g](\zeta)}{2\pi} \sum_{|k|\leq N} \frac{\sigma\left(\frac{|k-c|}{N}\right)\operatorname{sgn}(k-c)}{(k-c)} e^{ik(x-\zeta)} + \frac{[g'](\zeta)}{2\pi} \sum_{|k|\leq N} \frac{\sigma\left(\frac{|k|}{N}\right)\operatorname{sgn}(k-c)}{i(k-c)^2} e^{ik(x-\zeta)} + \dots \right)$$

(8)
$$= \frac{[g](\zeta)}{2\pi} \sum_{|k|\leq N} \frac{\sigma\left(\frac{|k-c|}{N}\right)\operatorname{sgn}(k-c)}{(k-c)} e^{i(k-c)(x-\zeta)} e^{icx}$$

(9)
$$+ \frac{[g'](\zeta)}{2\pi} \sum_{|k|\leq N} \frac{\sigma\left(\frac{|k-c|}{N}\right)\operatorname{sgn}(k-c)}{i(k-c)^2} e^{i(k-c)(x-\zeta)} e^{icx} + \dots$$

We now use the definition provided in Viswanathan, Gelb, Cochran for the shifted and scaled jump response and alter it slightly to deal with the $k-c$ terms:

(10)
$$\tilde{W}_q^{\sigma,N}(x) := \frac{1}{2\pi} \sum_{|k|\leq N} \frac{\sigma\left(\frac{|\mu|}{N}\right)\operatorname{sgn}(\mu)}{i^q \mu^{q+1}} e^{i\mu x},$$

where $\mu = k - c$ for each k and the fixed c.

Our partial Fourier summation can then be rewritten:

(11)
$$S_N^\sigma[g](x) = e^{icx} \left([g](\zeta)\tilde{W}_0^{\sigma,N}(x-\zeta) + [g'](\zeta)\tilde{W}_1^{\sigma,N}(x-\zeta) + \dots \right).$$

As in the previous paper, we use the regularized equivalent of a jump function to get around the fact that the jump function $[g](\zeta)$ has measure zero and makes analysis difficult. This regularized equivalent is:

(12)
$$[g]_\epsilon(x) = \begin{cases} [g](\zeta), & |x-\zeta| \leq \epsilon, > 0, \epsilon \sim O\left(\frac{2\pi}{2N+1}\right) \\ 0, & \text{else.} \end{cases}$$

Our partial Fourier summation then becomes:

$$\begin{aligned} S_N^\sigma[g](x) &\approx e^{icx} \left([g]_\epsilon(\zeta)\tilde{W}_0^{\sigma,N}(x-\zeta) + [g']_\epsilon(\zeta)\tilde{W}_1^{\sigma,N}(x-\zeta) + \dots \right) \\ &\approx \int_{-\pi}^{\pi} e^{icx}[g]_\epsilon(\zeta)\tilde{W}_0^{\sigma,N}(x-\zeta)d\zeta + \int_{-\pi}^{\pi} e^{icx}[g']_\epsilon(\zeta)\tilde{W}_1^{\sigma,N}(x-\zeta)d\zeta + \dots \\ &= e^{icx} \left(\int_{-\pi}^{\pi} [g]_\epsilon(\zeta)\tilde{W}_0^{\sigma,N}(x-\zeta)d\zeta + \int_{-\pi}^{\pi} [g']_\epsilon(\zeta)\tilde{W}_1^{\sigma,N}(x-\zeta)d\zeta + \dots \right) \end{aligned}$$

This is a convolution integral which can be simply written as:

(13)
$$S_N^\sigma[g](x) = e^{icx} \left(\left([g]_\epsilon * \tilde{W}_0^{\sigma,N} \right)(x) + \left([g]_\epsilon * \tilde{W}_1^{\sigma,N} \right)(x) + \dots \right).$$

Thus, the partial Fourier summation for the complex case appears similar to the partial Fourier summation for a real underlying function, with the distinction being that the convolution (using the altered version of the shifted and scaled jump response) is multiplied by the phase e^{icx}.

Also, for $N >>> c$, we note the following:

$$\begin{aligned}
\tilde{W}_q^{\sigma,N}(x) &= \frac{1}{2\pi} \sum_{|k|\leq N} \frac{\sigma\left(\frac{|k-c|}{N}\right) \operatorname{sgn}(k-c)}{i^q (k-c)^{q+1}} e^{i(k-c)x} \\
&= \frac{1}{2\pi} \sum_{-N-c\leq k'\leq N+c} \frac{\sigma\left(\frac{|k'|}{N}\right) \operatorname{sgn}(k')}{i^q (k')^{q+1}} e^{i(k')x} \\
&\approx \frac{1}{2\pi} \sum_{-N\leq k'\leq N} \frac{\sigma\left(\frac{|k'|}{N}\right) \operatorname{sgn}(k')}{i^q (k')^{q+1}} e^{i(k')x} \\
&= W_q^{\sigma,N}(x).
\end{aligned}$$

With this result, taking into account $N >>> c$, we then get that the partial Fourier summation is approximately that of the real case multiplied by the phase:

$$S_N^{\sigma}[g](x) \approx e^{icx} \left(\left([g]_\epsilon * W_0^{\sigma,N} \right)(x) + \left([g]_\epsilon * W_1^{\sigma,N} \right)(x) + \dots \right). \tag{14}$$

4. Interferometry Background

We describe the process for creating the interferograms between two passes. We assume first that both complex images have been properly registered such that the domain aperture regions are the same. Let f, g be the reconstructed image for Pass a and Pass b respectively. Let the two images be defined for the overlapping pixel x_1, y_1 with:

$$\begin{aligned}
f(x_1, y_1) &= r_A(x, y) \\
g(x_1, y_1) &= r_A(x, y) e^{i(\beta_f - \beta_g) Y_0 h(x,y)}.
\end{aligned}$$

Then the interferogram between Passes a and b is:

$$\begin{aligned}
\Phi_{f,g} = \angle (f * g) &= \angle \left(|r_A(x, y)|^2 e^{i(\beta_f - \beta_g) Y_0 h(x,y)} \right) \\
&= (\beta_f - \beta_g) Y_0 h(x, y),
\end{aligned}$$

where $\beta = \tan \psi$ for the angle of the pass ψ, Y_0 is a common offset, and $h(x, y)$ is the height function for the underlying image.

Thus, the height of the underlying image can be computed once we have the interferogram and values for β_f, β_g, Y_0:

Expanding on the equation above, we can the remove the scaling equation $(\beta_f - \beta_g)Y_0$ and then, taking into account the trigonometry of the layover which relates the physical coordinates of (x_1, y_1) to the physical space coordinates (x, y), we can actually find the true three-space terrain model $(x, y, h(x, y))$. Assuming we have already solved the 2-D unwrapped phase function $/Psi$, we show the terrain height depending only a function of the aperture position angles η and ψ. Substituting the explicit values for the variables Y_0, β_f, β_g in the above equation, we get:

$$\begin{aligned}
\Psi(x_1, y_1) &= \frac{4\pi}{\lambda} cos\psi_g (tan\psi_f - tan\psi_g) h(x, y) \\
&= \frac{4\pi}{\lambda} cos\psi_g (tan(\psi_g + \Delta\psi) - tan\psi_g) h(x, y)
\end{aligned}$$

where $\Delta\psi = \psi_f - \psi_g$ and λ refers to the wavelength of the signal and ψ is the azimuth angle. Since $\Delta\psi << 1$, we can employ trigonometric rule to yield:

$$\begin{aligned}
\Psi(x_1, y_1) &= \frac{4\pi}{\lambda} cos\psi_g \left(\frac{tan\psi_g + tan\Delta\psi}{1 - tan\psi_g tan\Delta\psi} - tan\psi_g\right) h(x, y) \\
&= \frac{4\pi}{\lambda} cos\psi_g tan\Delta\psi \left(\frac{1 + tan^2\psi_g}{1 - tan\psi_g tan\Delta\psi}\right) h(x, y) \\
&\cong \frac{4\pi}{\lambda} cos\psi_g (\Delta\psi) \left(\frac{sec^2\psi_g}{1 - tan\psi_g(\Delta\psi)}\right) h(x, y) \\
&= \frac{4\pi}{\lambda} \frac{\Delta\psi}{cos\psi_g} h(x, y) \left(\frac{1}{1 - tan\psi_g(\Delta\psi)}\right) \\
&\cong \frac{4\pi}{\lambda} \frac{\Delta\psi}{cos\psi_g} h(x, y)
\end{aligned}$$

It then follows that the the terrain height function is:

$$h(x, y) = \frac{\lambda}{4\pi} \frac{cos\psi_g}{\Delta\psi} \Psi(x_1, y_1)$$

We also know the formulas to get from x_1 to x and y_1 to y using the the elevation and azimuth angles of the pass, if we assume no superposition has occurred

$$x_1 = x + tan\eta_g h(x, y)$$

$$y_1 = y + tan\psi_g h(x, y)$$

Solving these three equations simultaneously will give us the true three-space terrain model

We note that unwrapping needs to be done to recover $\Phi_{f,g}$ before trying to recover the height function. This becomes apparent when looking at an interferogram for Passes 1 and 6 using the standard approach as seen in Figure 1.

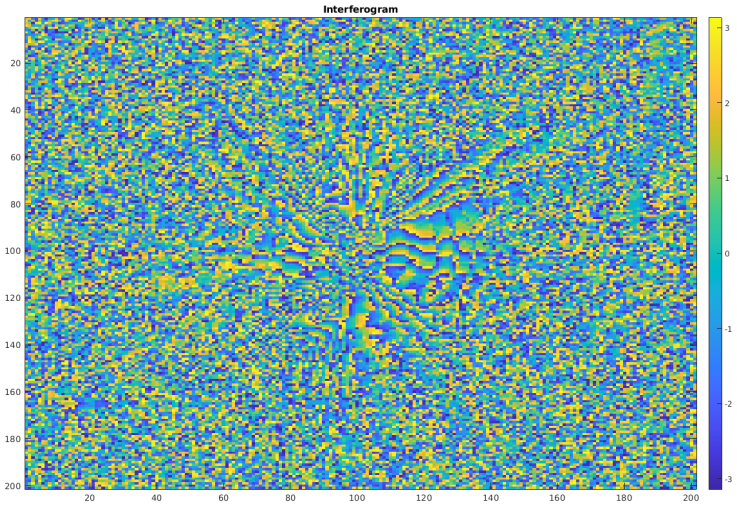

FIGURE 1. Interferogram between Pass 1 and Pass 6

4.1. **Maximum Likelihood Estimator Approach.** Especially in the presence of noise within the image, it can make sense to use an MLE approach. to create an interferogram. This follows a similar formation to the standard interferogram approach. However, for the MLE approach, we compute the MLE estimate for every neighborhood of pixels of size k around a pixel:

$$\Phi_{f,g}^{MLE} = \hat{\phi}_{ML} = \angle \left(\sum_{k=1}^{N} f_k g_k^* \right).$$

The interferogram for Passes 1 and 4 (before we use unwrapping, described below) using the MLE approach can be seen in Figure 2.

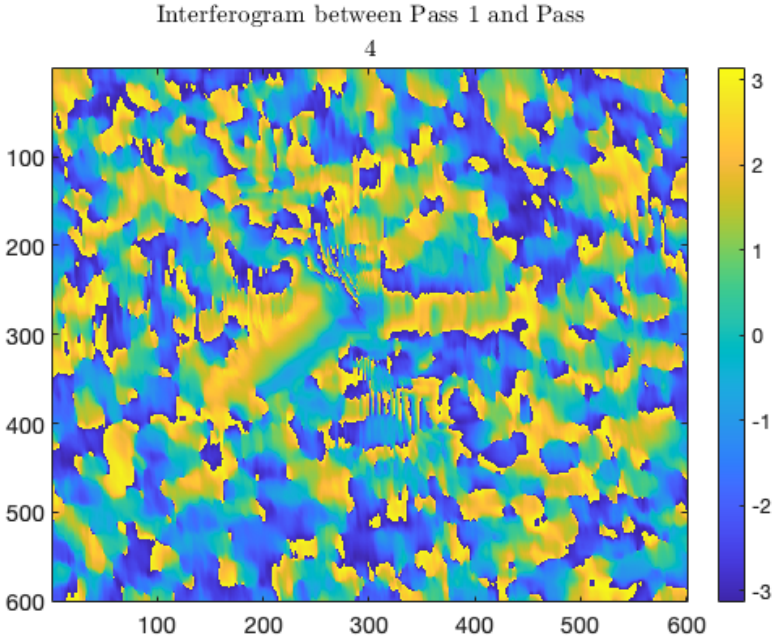

FIGURE 2. Interferogram between Pass 1 and Pass 4 Using MLE Approach, Neighborhood Size 3

5. 2-D Phase Unwrapping

We detail here the algorithm we are currently using to unwrap the phase values for the interferograms. The phase unwrapping process is applied after the interferogram is created, whether using the standard or the MLE approach. 2-D phase unwrapping is an ongoing area of research outside of this project. The phase unwrapping algorithm we utilize is found in [Jakowatz et al. 1996, 314-315]: In the future, we might change the unwrapping function to one that is a weighted least squares optimization problem, but at this moment in our research, we have found that we are reaching an error when we look at other the unwrapped phase data - a problem that will not be fixed by switching to this kind of unwrapping algorithm

In the below interferograms, we demonstrate the outputs we are getting now and where exactly are methods are going wrong

(1) Compute the wrapped first differences, where $W(\Psi_{ij}) = \phi_{ij}$ is the phase value at each pixel:

$$\Delta^x_{i,j} = \begin{cases} W\{\phi_{i+1,j} - \phi_{i,j}\}, & i \in [0, M-2], j \in [0, N-1] \\ 0 & \text{otherwise} \end{cases},$$

$$\Delta^y_{i,j} = \begin{cases} W\{\phi_{i,j+1} - \phi_{i,j}\}, & i \in [0, M-1], j \in [0, N-2] \\ 0 & \text{otherwise} \end{cases}$$

(2) Compute the driving function $d_{i,j}$ for each pixel value:

$$d_{i,j} = \left(\Delta^x_{i,j} - \Delta^x_{i-1,j}\right) + \left(\Delta^y_{i,j} - \Delta^y_{i,j-1}\right).$$

(3) Compute the Discrete Cosine Transform of $d_{i,j}$:

$$\bar{d}_{i,j} = DCT(d_{i,j})$$

(4) Modify $d_{i,j}$ by applying the 2-D discrete cosine transform to the interferogram values as well:

$$\bar{\Psi}_{i,j} = \frac{\bar{d}_{i,j}}{2(\cos \pi i/M + \cos \pi j/N - 2)}$$

(5) Perform the 2-D inverse DCT of $\bar{\Psi}_{i,j}$ to obtain the least-squares unwrapped phase values $\Psi_{i,j}$.

We now look at unwrapped interferograms for Passes 1 and 4 using both the standard interferogram approach and the MLE approach in Figures 3 and 4, respectively. It should be obvious that the massive phase change in the middle of both approaches presents a problem.

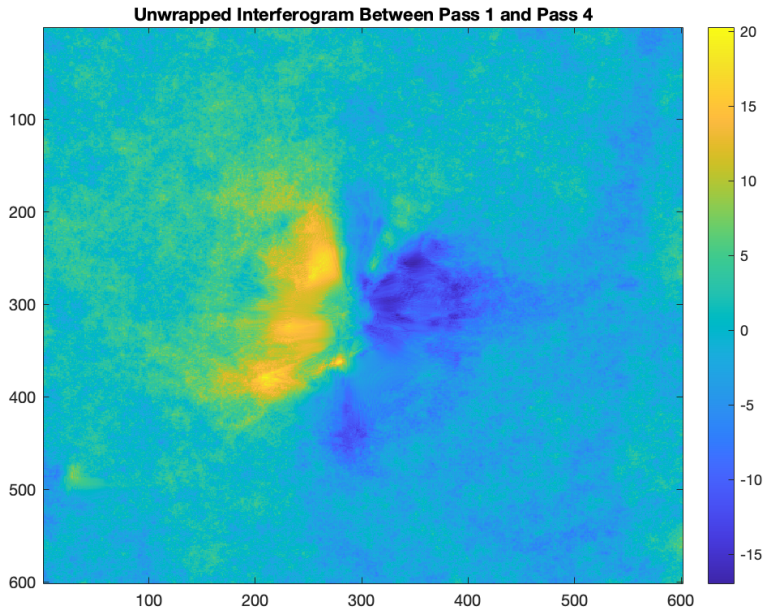

FIGURE 3. Interferogram between Pass 1 and Pass 4 Using 2-D Unwrapping

6. Analyzing the Unwrapped Interferometric Values from Rows

Need to put in an explanation of interferometry and related equations. Later should put in method to interpolate to other (missing) passes, and explain how that will be helpful in overall height approximation.

We next look at the unwrapped phase data from different rows of the interferogram between Pass 1 and Pass 4. As we can see in Figure 5, the values of the unwrapped interferogram between Passes 1 and 4 follow a similar sinusoidal shape if the rows of the image do not contain the tophat, and they follow a different, more heightened shape if the tophat is in that row. Figure 6 shows the difference between Row 100 of the image, that does not contain any of the tophat image, and Row 300, which contains close to the diameter of the tophat.

We next confirm that the same type of behavior is seen in another interferogram, between Passes 1 and 6. As can be seen in Figure 7, we see the same type of behavior. Figure 8 shows a cleaner comparison between Row 100 and Row 300.

Finally, we analyze the interferometric values (unwrapped) across multiple interferograms for the same row. We choose Row 300 because it contains a large amount of the tophat structure where we expect to see the wide oscillatory pattern seen in the previous figures from about pixel 150 to pixel 450. We see in Figure 9 that Interferograms between Pass 1 and Passes 2, 3, 4, 6, 7 (individually) all have a similar wide oscillatory structure between these pixel values, but increasing in their extremes as we increase the distance between passes (Pass 7 is higher and farther away from Pass 1 than Pass 3, for instance).

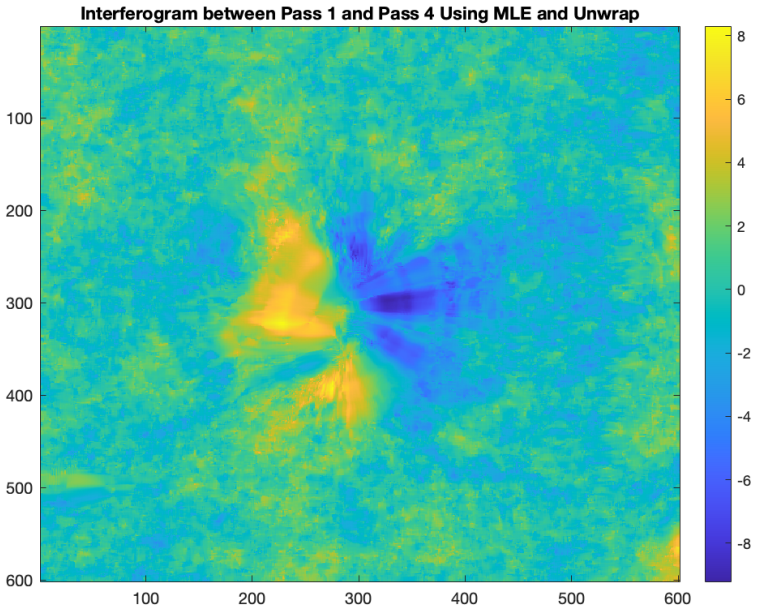

FIGURE 4. Interferogram between Pass 1 and Pass 4 Using MLE Approach and 2-D Phase Unwrap, Neighborhood Size 3

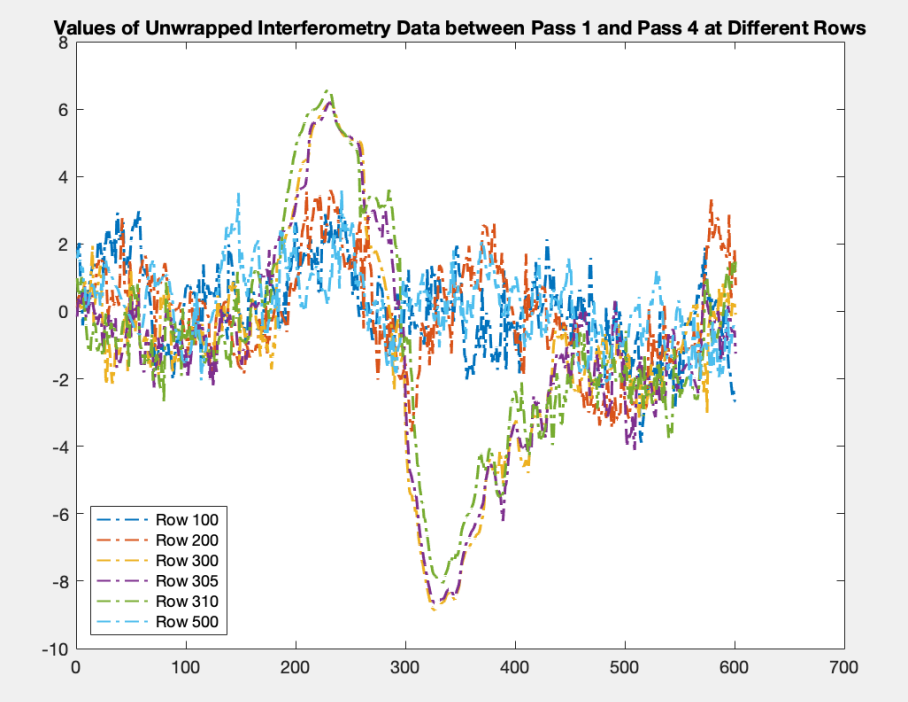

FIGURE 5. Interferometric Values (Unwrapped) for Rows of Image, Passes 1 and 4

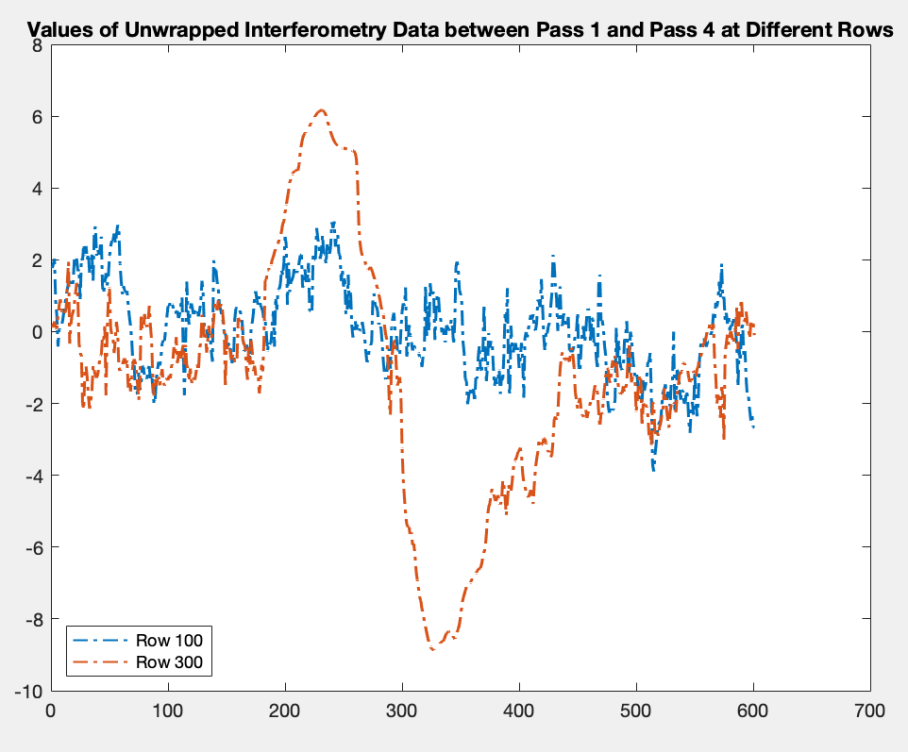

FIGURE 6. Interferometric Values (Unwrapped) for Rows 100, 300 of Image, Passes 1 and 4

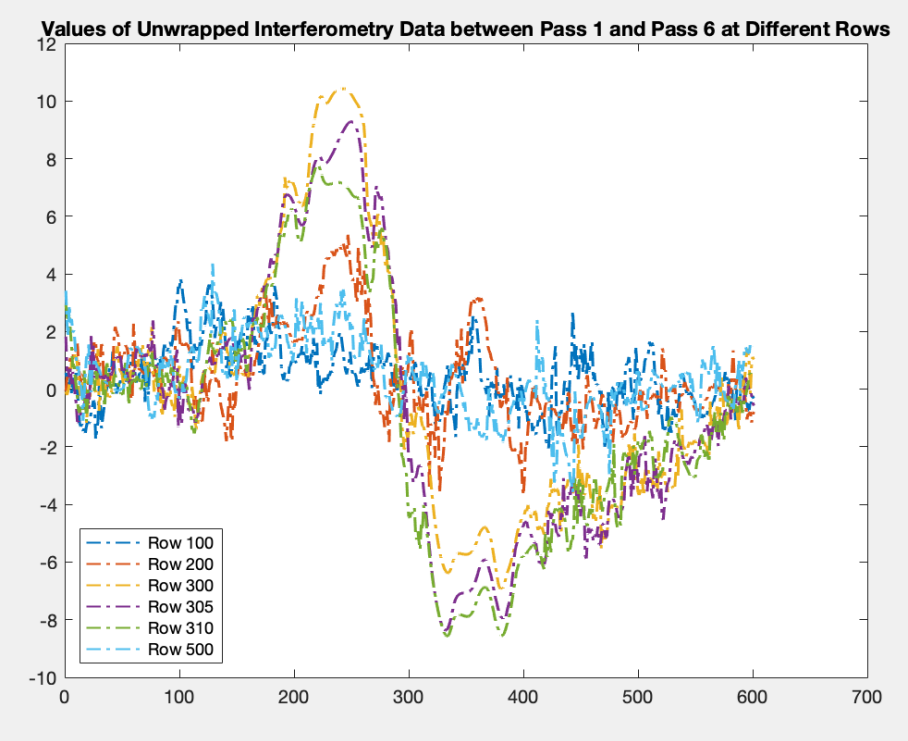

FIGURE 7. Interferometric Values (Unwrapped) for Rows of Image, Passes 1 and 6

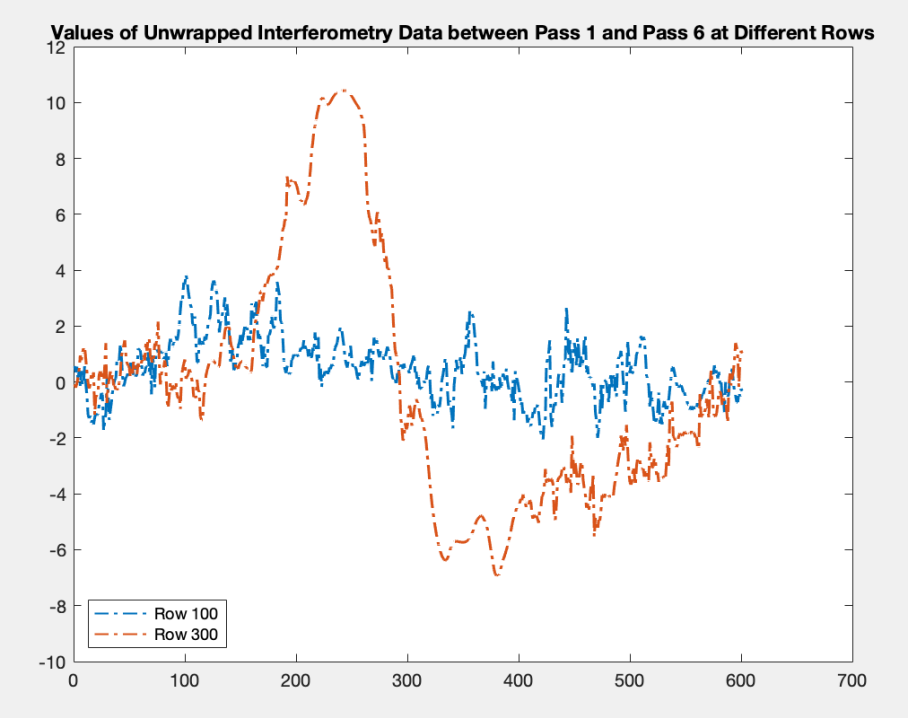

FIGURE 8. Interferometric Values (Unwrapped) for Rows 100, 300 of Image, Passes 1 and 6

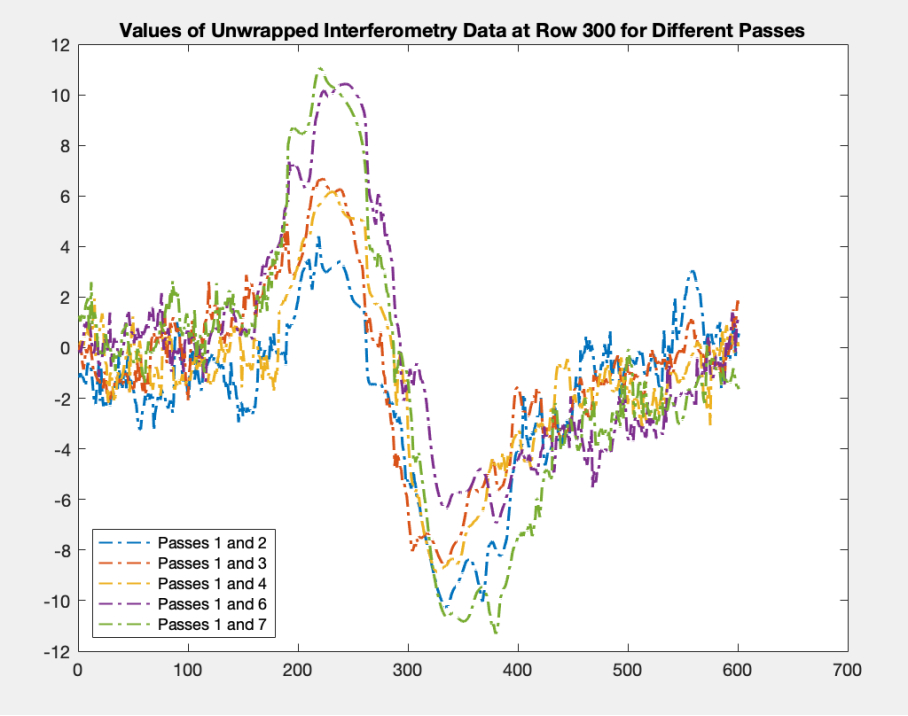

FIGURE 9. Interferometric Values (Unwrapped) for Row 300 of Image, Multiple Interferograms

7. Interpolating Values from Missing Passes

In this section, we discuss the methods we have developed to interpolate phase history

8. Notes from Meeting 4/20

[Transcribed by Ian]

8.1. **Chapter 2. Mathematical Foundation.** SAR data are gathered by some (moving) imaging platform sending and receiving a signal. Using an inverse Fourier transform, we get a projection of the whole underlying scene (in this case 2-D scene) onto a 1-D line. Having our imaging platform moving allows us to use a Radon transform along all the different viewing angles we have. Using a shorter antenna, we collect over a wider angle. On page 43, we have reflectivity for every angle that the signaling platform sends out for. On page 48, we begin with spotlight SAR. In spotlight SAR, we typically fly around the scene of focus, but we use (approximately) the same scene focus for each signal we collect.

On page 56, we have a plot of what data in frequency/Fourier space is for a CAT scan. A SAR data plot looks like an annulus rather than a circle because SAR data is collected on certain frequency bands (determined by hardware or settings of device). Because we are looking at reflectivity rather than going through the image (as happens in CAT scans), we need to have a pause in sending the signal out and receiving it, leading to an annulus rather than a full circle. In 2.3, we have algorithms for CAT scans that are similar to the algorithms people use for SAR imaging. For example, for a section of the annulus, people will use polar reformatting algorithm to turn the annulus into a rectangle of data from which we can use a 2-D inverse Fourier transform to reconstruct the underlying image. People also use filtered back projection for SAR for every single angle we have data for to reconstruct the underlying image. The Projection Slice Theorem says that a Fourier transform of the plane that is a projection of a 3-D scene is the same as a Fourier transform of the entire 3-D scene and then slicing it. This Projection Slice Theorem holds for m and n-dimensional subspaces.

9. Notes from Meeting 4/25

[Transcribed by Ian]

Include the phase unwrapping from the book (which is used in the MATLAB code) and how height in the image is related to the phase (derivation of how to get the height from the phase).

9.1. **Explanation of GOTCHA Dataset.** Interested only in HH within the GOTCHA dataset. Everything is split up by azimuth angle from 1 to 360. Within azimuth 1, for example, we have frequency (some band of frequency). For every pulse that the radar has sent out and been collected back, these are the frequencies we have data available. We also have x, y, z vectors for each azimuth, so we have 117 points and 424 frequencies, giving us 117×424 points for phase history data. Each azimuth has some number of times that radar frequencies were sent out (it's not the same every time, about 117 for each one) and every time a collection was made, we have it for 424 frequencies. Theta gives you what the azimuth angle is, and phi gives you what the elevation angle is.

10. Notes for Meeting 4/8

10.1. **Understanding Unwrapping Algorithm.** Given that the unwrapping algorithm is leading to negative height values on some parts of the tophat, we are motivated to better understand the algorithm we are using and how our implementation might be causing issues. We first consider whether there is a problem in where the tophat is located within the image. We zoom out to reposition the tophat so that it is in the upper left quarter of the image. After creating the same standard interferogram and applying the 2-D phase unwrapping algorithm, we get Figure 10. Thus,

we note that we are still experiencing the same problems as when the tophat was at the center of the image.

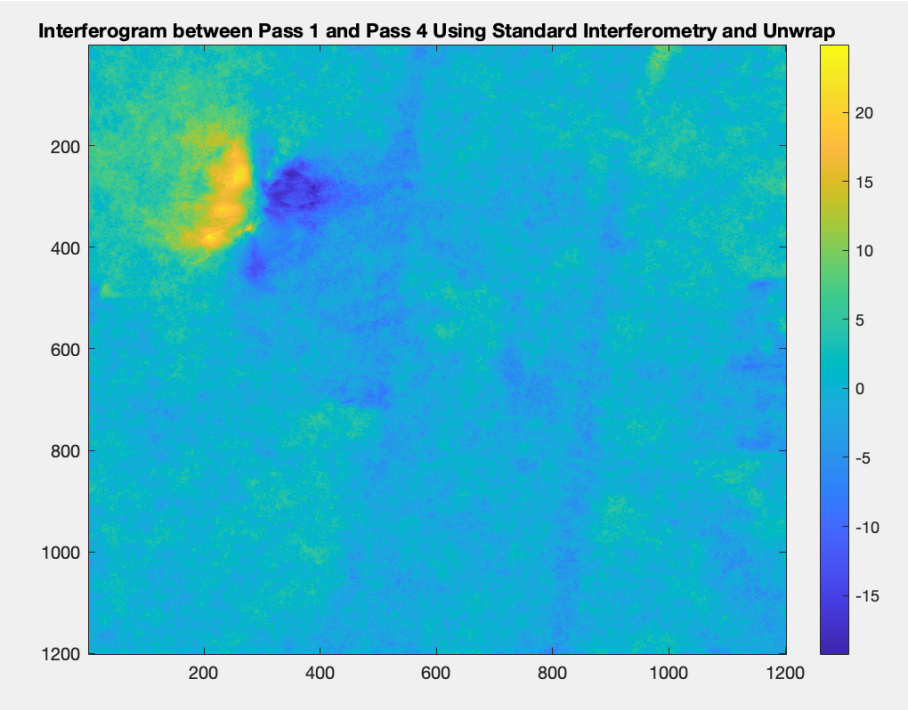

FIGURE 10. Interferogram between Passes 1 and 4 With Tophat in Upper Left Quadrant, Unwrapped

We next consider whether we can reproduce results from the 1-D case, where the unwrap algorithm is standard, using the 2-D algorithm as we are currently implementing it by fixing one direction and then applying our 2-D algorithm. For this trial, we choose a linear phase function dependent only on one variable, $f(x, y) = e^{i\pi x}$. We then try unwrapping using two different approaches: first, we use the 2-D algorithm as it is currently implemented; second, we apply the 1-D algorithm that is built into MATLAB (the unwrap function) onto the phase portion of $f(x) = e^{i\pi x}$ and then turn it into a 2-D matrix by multiplying by a column vector of 1's to ease comparison. The wrapped data can be seen in Figure 11, while the comparison of the two unwrapping methods applied on data fixed in one direction is seen in Figure 12. As can be seen, the 2-D algorithm we are currently using seems to have a problem with scaling at some point, as the 1-D algorithm has a range of $2 \cdot \pi^2$ between its lowest and highest points while the 2-D algorithm has a range of less than $2 \cdot \pi$.

The same problem can be seen when fixing the function in the other direction with $f(x, y) = e^{i\pi y}$, as seen in Figures 13 and 14.

We try further changes to see if we can see differences between the 2-D unwrapping algorithm and the 1-D algorithm. For a piecewise continuous box function fixed in one direction, $f(x, y) = e^{icy}$ for $|y| \leq 2$ and 0 otherwise, we get the results seen in Figures 15 and 16.

Thus, we note that there is likely some implementation problem with the current 2-D unwrapping algorithm since it does not hold in the 1-D case. There seems to be a scaling problem given how the maximum and minimum phase values differ between the 2-D phase unwrap and the 1-D phase unwrap. We also note the slight differences between the 2-D phase unwrap and the 1-D phase unwrap in Figure 16 occurring where there are jumps in the magnitude. This likely also plays a

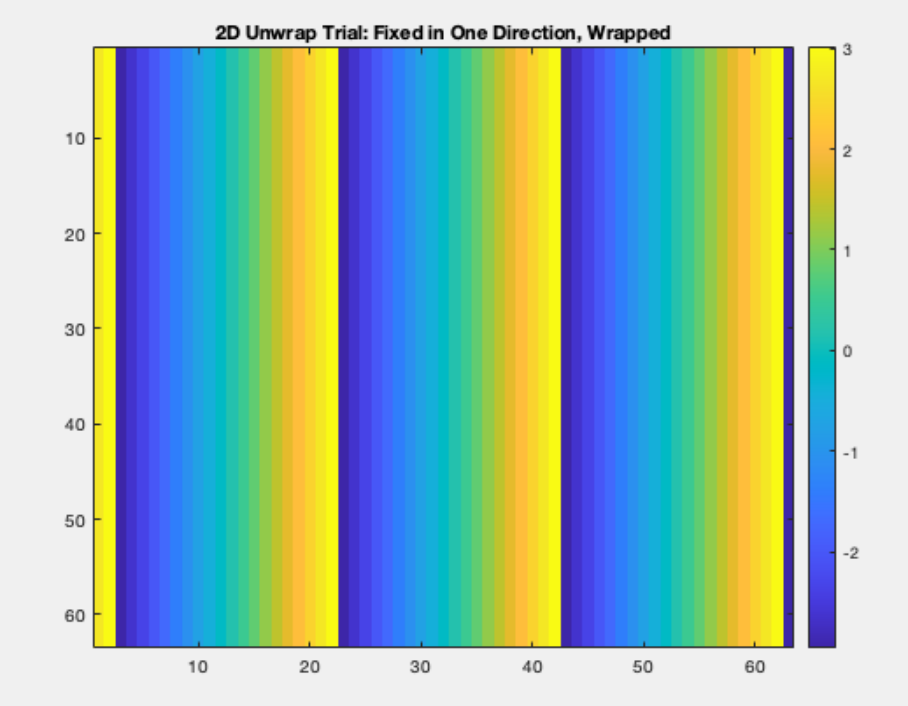

FIGURE 11. Wrapped phase of data of linear phase function fixed in one direction

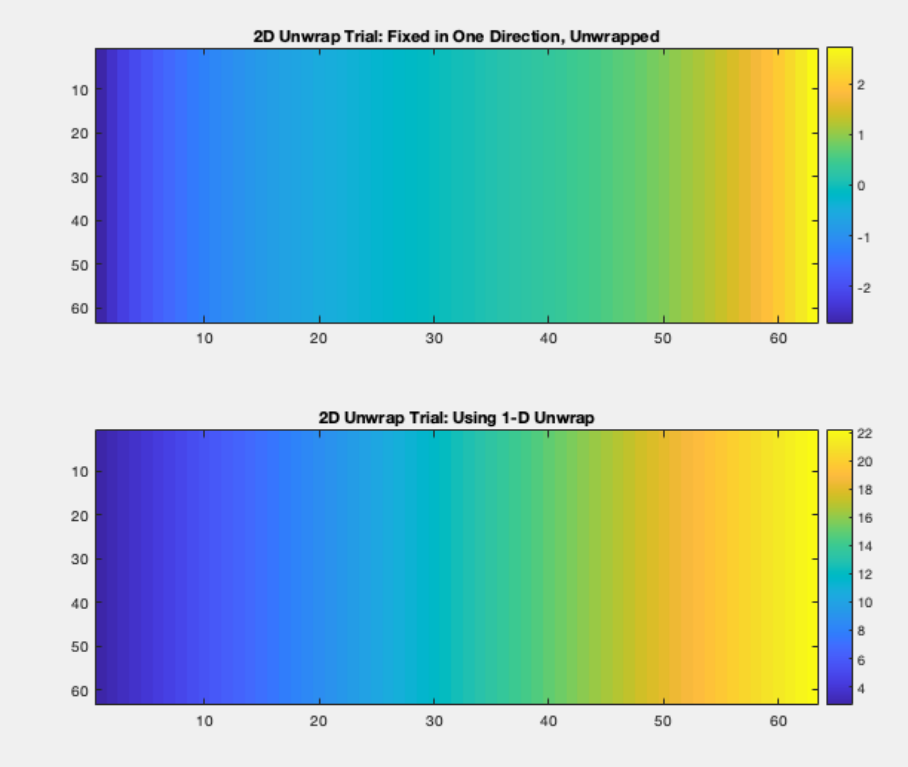

FIGURE 12. Comparison of methods for unwrapped phase of data of linear phase function fixed in one direction

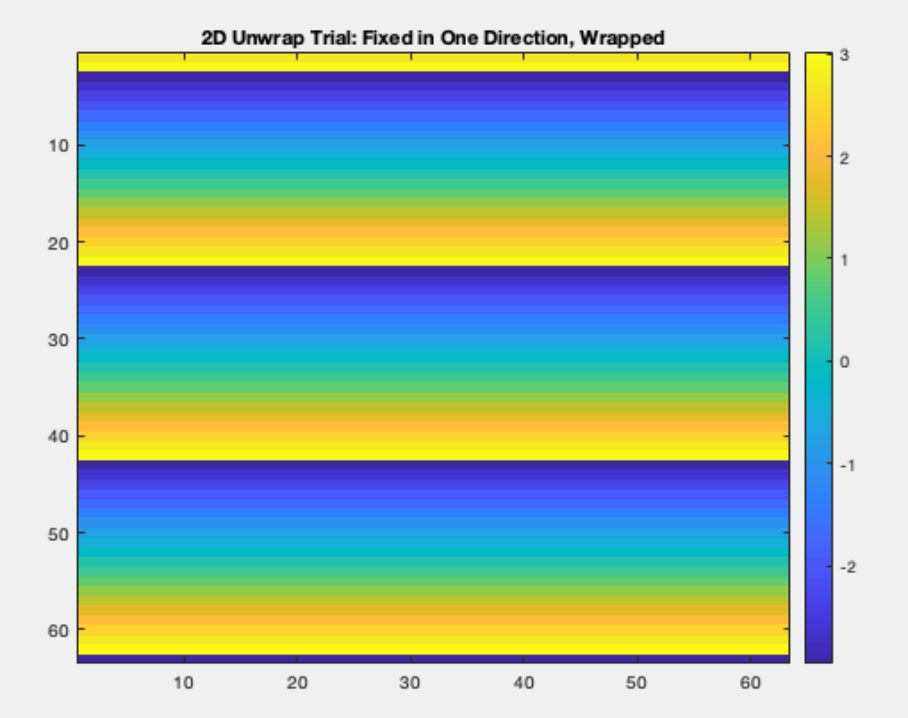

FIGURE 13. Wrapped phase of data of linear phase function fixed in the other direction

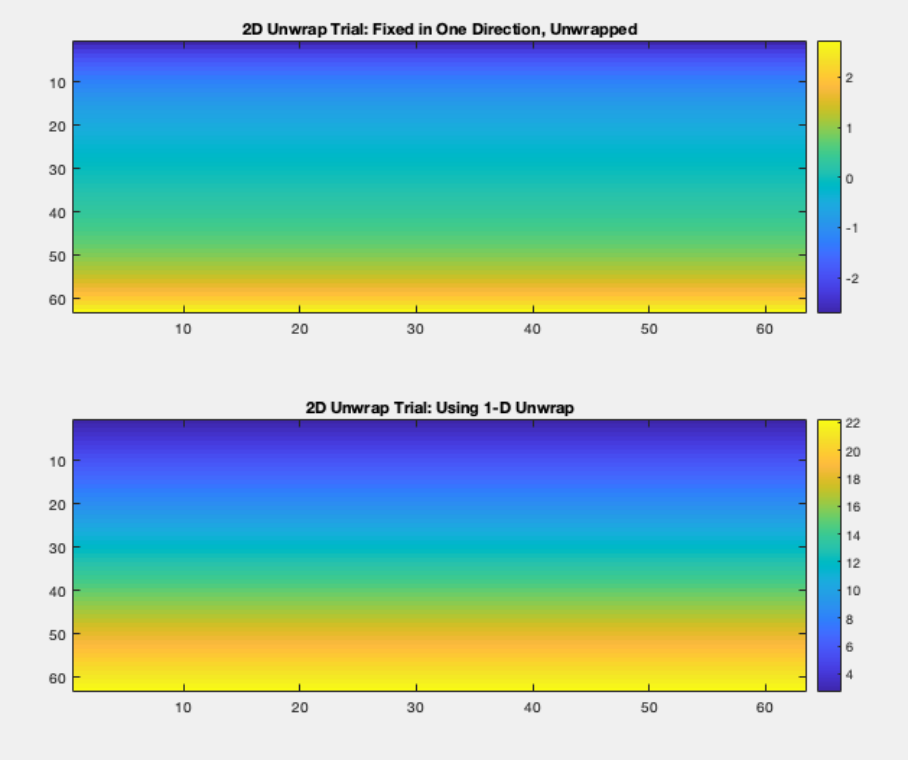

FIGURE 14. Comparison of methods for unwrapped phase of data of linear phase function fixed in the other direction

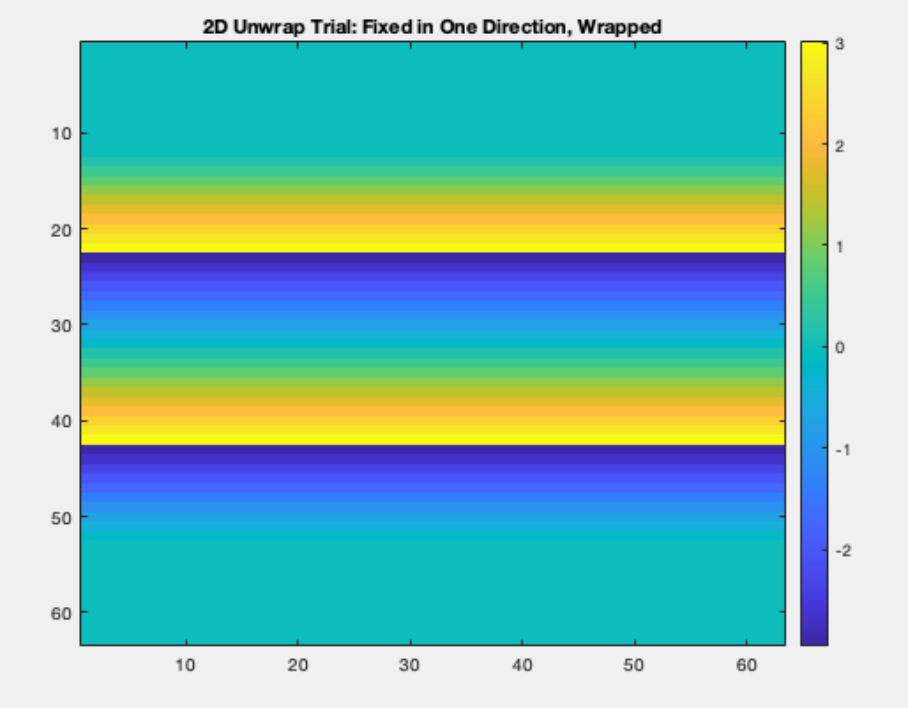

FIGURE 15. Wrapped phase of data of complex box function fixed in the x direction

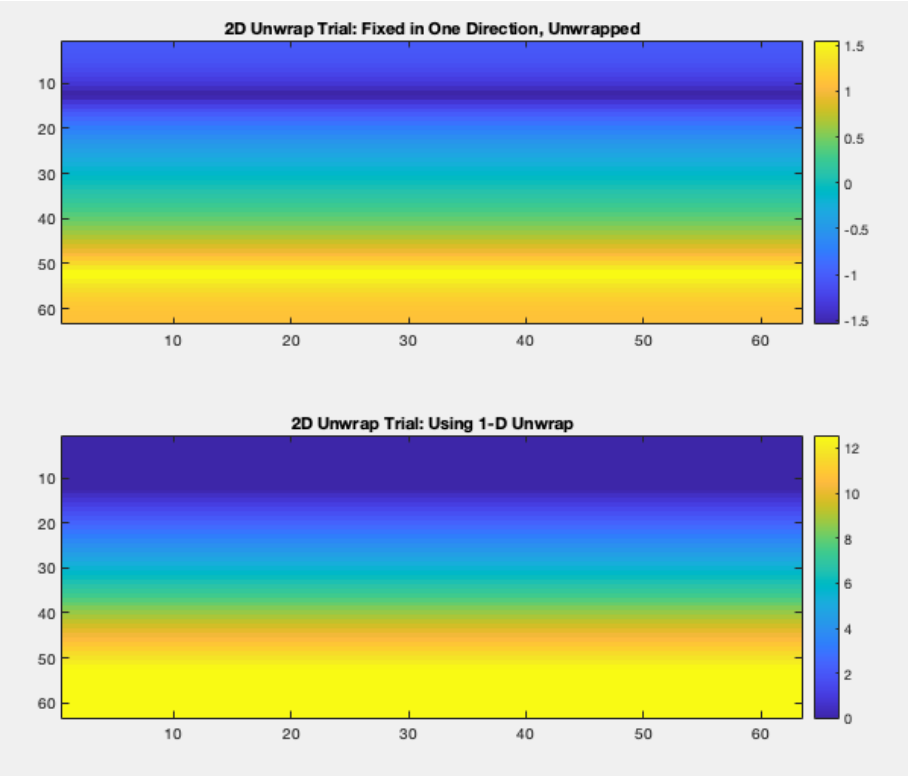

FIGURE 16. Comparison of methods for unwrapped phase of data of complex box function fixed in the x direction

role in the problems we are seeing in the tophat. However, we note that, in this case, the 2-D unwrap and 1-D unwrap both go from negative to positive phase values even as the box function has a negative height difference (going from 1 to 0 in the magnitude) near pixel 50 in the y direction.

Because of the results seen above, we then try to go dimension by dimension in trying to phase unwrap in 2-D. However, this process did not work, even after trying to scale each row or column by finding the phase unwrap in the other direction for the first column or row, respectively. We thus turn to pre-written code found online for other 2-D phase unwrapping algorithms to see if we can properly unwrap the image we have chosen with a sizeable jump discontinuity in the center (the tophat structure).

Henry has written up MATLAB code to enable the group to recover the height of the image once the phase of the image is properly unwrapped. However, until a proper 2D phase unwrap can be done, we are unable to use the code.

One possible 2-D phase unwrapping algorithm is that proposed by Herráez et al. (2002) which was put into MATLAB code by Kasim (2017). When trying to use this unwrapping algorithm on the interferogram between passes 1 and 4 focused on the tophat structure, however, we again don't see positive values where the tophat is in the unwrap, as can be seen in Figure 17. Here, we still note that there is the possibility that we have a problem with mis-registration.

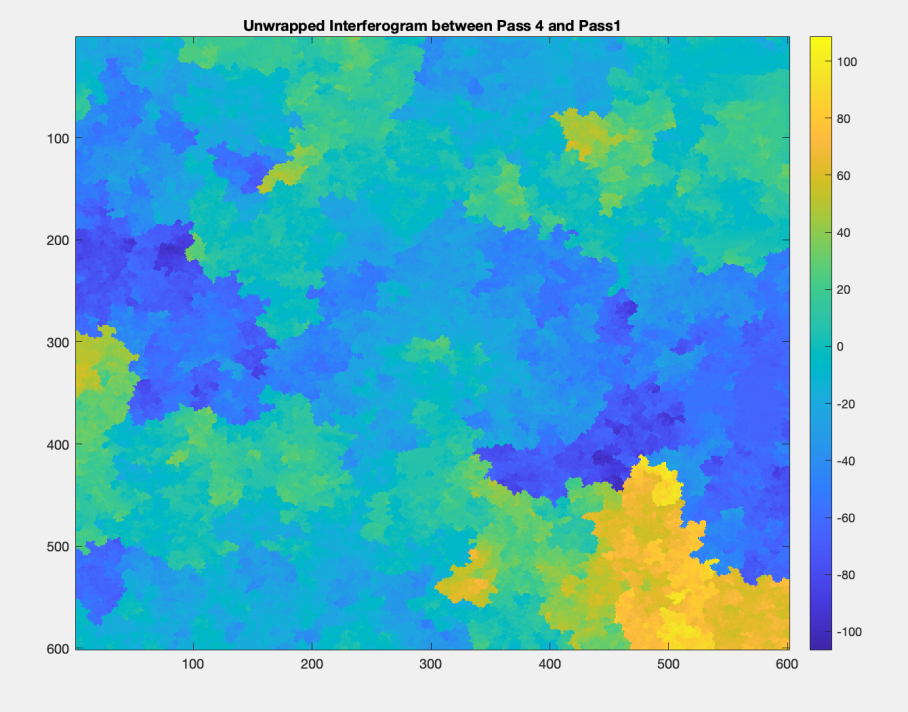

FIGURE 17. Unwrapped Interferogram Using Kasim Algorithm, Passes 1 and 4

We then try to apply this same interferogram unwrapping algorithm on a created function. For this, we use a sample function:

$$f(x, y) = \sin(x + y) \cdot \exp(i\pi(x + y)), \quad x, y \in [-\pi, \pi].$$

We then use both the Kasim algorithm and the algorithm described above from Jakowatz et al. to compare for a 250 x 250 image. The results are seen in Figures 18, 19, 20.

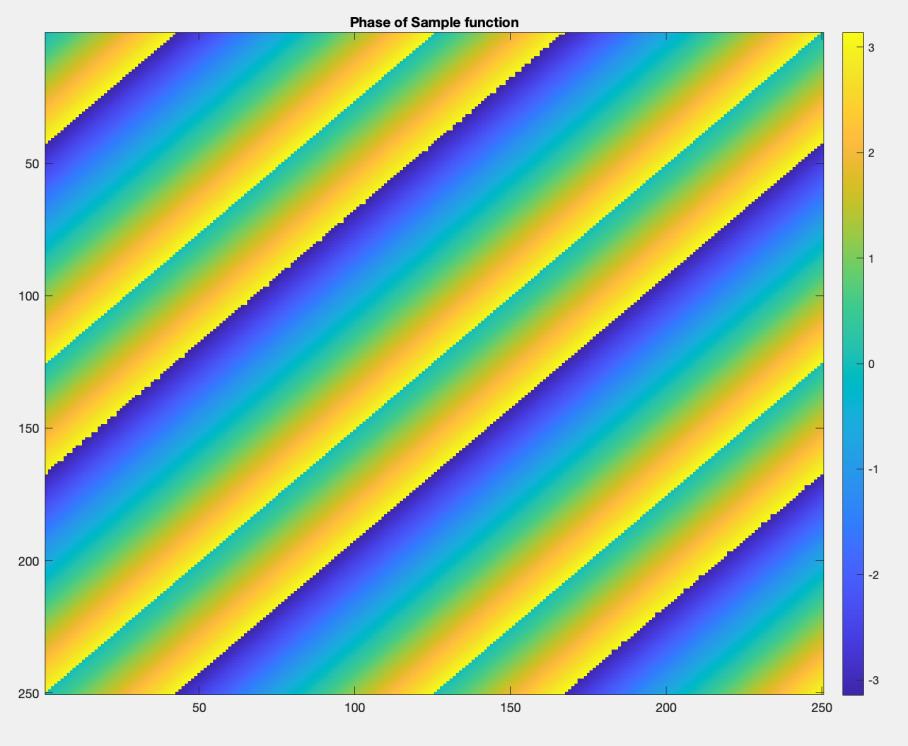

FIGURE 18. Wrapped phase of sample function $f(x, y)$

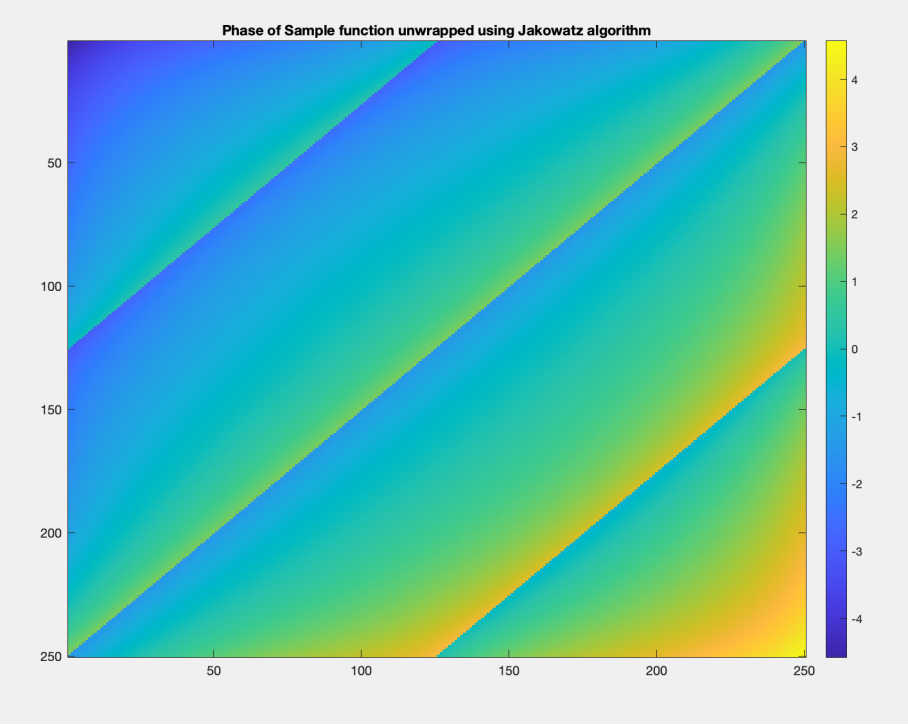

FIGURE 19. Phase of sample function $f(x, y)$, Unwrapped Using Jakowatz Algorithm

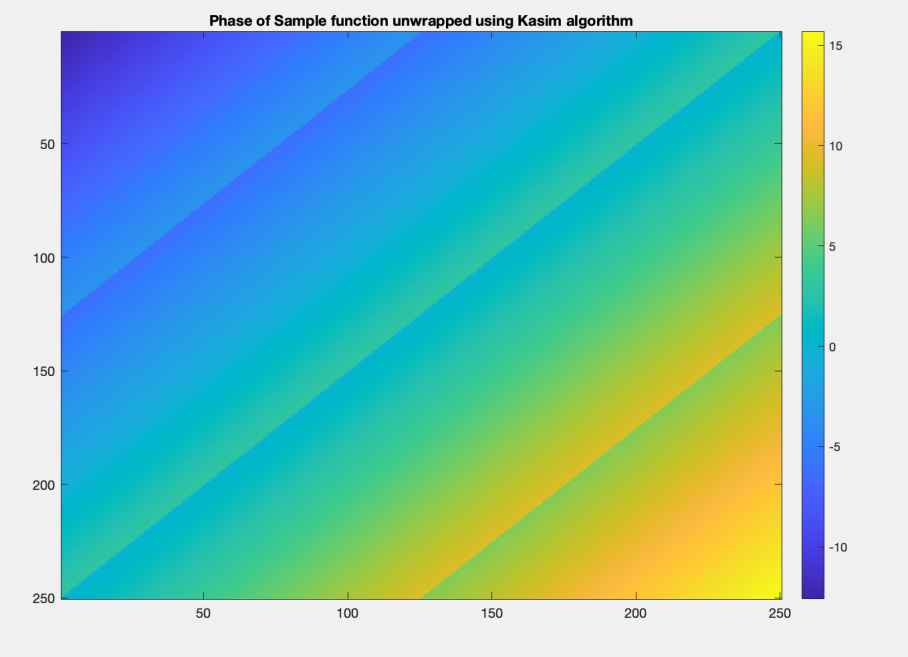

FIGURE 20. Phase of sample function $f(x, y)$, Unwrapped Using Kasim Algorithm

Here, we identify again that there seems to be a scaling problem with the Jakowatz algorithm as currently implemented in our MATLAB code, since the Kasim algorithm is closer to the true phase values of $-2\pi^2$ to $2\pi^2$, whereas the Jakowatz unwrapping algorithm only gives (-4, 4).