1-15-2019

Wavepy - Python package for x-ray grating interferometry with applications in imaging and wavefront characterization

Walan Grizolli
The Advanced Photon Source

Xianbo Shi
The Advanced Photon Source

Lahsen Assoufid
The Advanced Photon Source

Leslie G. Butler
Louisiana State University

Recommended Citation

Grizolli, W., Shi, X., Assoufid, L., & Butler, L. (2019). Wavepy - Python package for x-ray grating interferometry with applications in imaging and wavefront characterization. *AIP Conference Proceedings, 2054* https://doi.org/10.1063/1.5084648

Wavepy - python package for x-ray grating interferometry with applications in imaging and wavefront characterization

Cite as: AIP Conference Proceedings **2054**, 060017 (2019); https://doi.org/10.1063/1.5084648
Published Online: 16 January 2019

Walan Grizolli, Xianbo Shi, Lahsen Assoufid, et al.

wavepy - Python package for x-ray grating interferometry with applications in imaging and wavefront characterization

Walan Grizolli[1,a)], Xianbo Shi[1], Lahsen Assoufid[1] and Leslie G. Butler[2]

[1]*Advanced Photon Source, Argonne National Laboratory, 9700 S. Cass Ave., Argonne, Illinois 60439, USA*
[2]*Department of Chemistry, Louisiana State University, Baton Rouge, Louisiana 70803, USA*

[a)]Corresponding author: Walan.Grizolli@anl.gov

Abstract. X-ray grating interferometry has been employed for a wide range of imaging applications by using the Talbot self-images of a grating. At synchrotron facilities, it has also been exploited to characterize the coherence and wavefront of the x-ray beam. The method is relatively straightforward, easy to implement, and provides accurate quantitative data. However, to exploit its full potential, it requires highly specialized, robust data analysis. It this work, we have set the framework for the development and dissemination of a robust open-source code that is tailored to the data analysis for imaging and beam characterization using grating interferometry.

INTRODUCTION

X-ray Talbot interferometers [1–3] are a versatile tool that provides different information of a sample being studied. It relies on the Talbot self imaging effect [4], which is the image formation of a periodic structure at certain distances. By measuring the distortions to the Talbot self image caused by an sample (compared to a reference image without the sample), one obtains the differential phase image [2, 5] and the dark-field image [5, 6] of the sample. By measuring the propagation of the distortions as the beam propagates, one can retrieve the wavefront profile of the beam[7, 8].

Talbot effect is a wave optics effect [4] and thus it is moderated by the limited coherence of the beam [9, 10]. On the other hand, the effects of partially coherent beams are well understood and therefore the Talbot interferometer can be used to obtain the coherence function of a partially coherent beam [11–13].

The experimental setup is reasonably simple and variations of the experimental setup have been developed over the years with different strengths and drawbacks [14], namely single grating [8, 15] and stepped grating[2]. Different gratings can be used and optimized to different experiments, which may require phase gratings or absorption gratings or both (stepped geometry). The gratings can be 1-dimensional [10] or 2-dimensional gratings[16, 17]. Circular gratings can be used for coherence measurements [18].

To process the data and obtain the actual physical results, advanced and dedicated data analysis are required. In this work we present *wavepy*, a Python [19] library for data analysis of x-ray grating interferometry. The library takes advantage of the Python programming language capabilities and flexibility, and aims to be easy to collaborate, distribute, and integrate with other imaging packages. We discuss all the main features of the code and provide examples of results obtained using the library.

SINGLE GRATING INTERFEROMETRY

Experimentally, the single grating experiment starts by placing the detector at the desired Talbot distance [4]. The Talbot distances can be either calculated [17] or measured experimentally [12]. Longer distances improve the sensitivity of the experiment, but the finite coherence length limits the maximum usable distance [13]. Therefore, it is good practice to measure the Talbot distance to verify that there is contrast at that distance. Next is the alignment of the grating and of the sample, with the requirement that the Talbot image features must be aligned parallel or at 45 degrees to the detector pixel array [12]. Two images are then acquired, one with sample and the other without the sample (the

Proceedings of the 13th International Conference on Synchrotron Radiation Instrumentation – SRI2018
AIP Conf. Proc. 2054, 060017-1–060017-6; https://doi.org/10.1063/1.5084648
Published by AIP Publishing. 978-0-7354-1782-3/$30.00

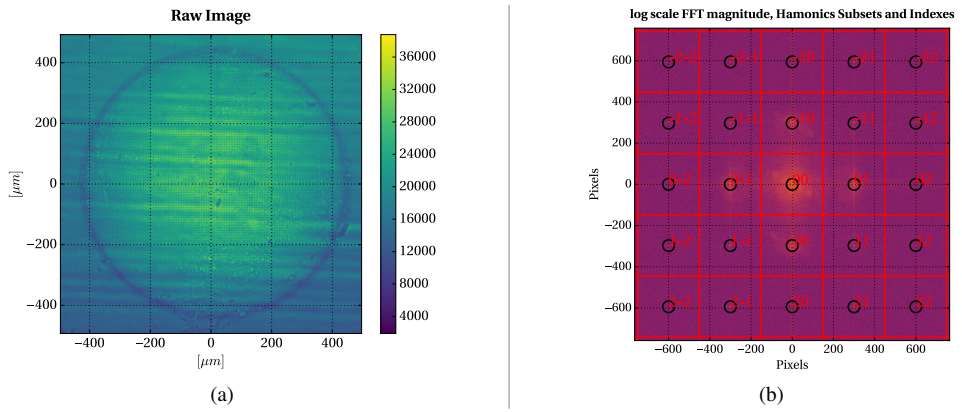

FIGURE 1. (a) A diamond concave lens imaged with a 2D phase grating (the pattern period is 2.4 μm, too small to be seen in the image). (b) The magnitude value of the Fourier transform showing harmonics created by the 2D phase grating. The superimposed grid pattern is added to highlight and identify the harmonics. Color scale is a logarithmic scale.

reference image). A dark image (no beam, same acquisition time) is acquired to account for the dark noise of the detector.

The data analysis of single grating imaging relies on Fourier transform phase retrieval [15]. In practice, this is done by first calculating the Fourier Transform (FT) of the experimental image. In Fig. 1.a is shown the raw image of a parobolic diamond lens measured with the portable interferometer [1] at APS 1-BM-B beamline [20]. The FT image will present a series of harmonics peaks (Fig. 1.b) at the positions

$$h_n = n \cdot \frac{L}{p} + h_0, \tag{1}$$

where L is the lateral image size and p is the pattern period [12, 17], both in meters. h_n and h_0 are the pixel position in the FT image for the n^{th} and zeroth harmonic, and h_0 is at the center of the image. Remembering that the FT are complex numbers of the form $a(f_x, f_y)e^{i\phi(f_x,f_y)}$, at the harmonics we have $f_x = h_{nh}$, $f_y = h_{mv}$ and the value of the magnitude $a(f_x, f_y)$ are local maxima (the indexes $_v$ and $_h$ stand for vertical and horizontal, respectively).

When using the 2D grating, Eq. 1 is used for the vertical and horizontal harmonics, and L, p and h_0 may have different values L_v, L_h, p_v, p_h, h_{0v} and h_{0h} for each direction. The FT image is then cropped around the vertical harmonic h_{1v}. The lateral size of the cropped image is L_v/p_v and L_h/p_h. The cropped images are then shifted in the frequency domain by h_{1v}, which in practice is done by assuming the center of the FT image as the zero frequency, that is, $h_{1v} \to 0$. The same cropping procedure is done around the horizontal frequency h_{1h}, resulting in two cropped images (see Fig. 2).

Now, according to the Fourier transform phase retrieval [21], the argument $\phi(f_x, f_y)$ of the cropped images are direct measurements of the displacement of the fringes in the Talbot images, and are related to the DPC image as [15]:

$$DPC_H = -\frac{(\phi^{01}_{\text{sample}} - \phi^{01}_{\text{ref}}) \cdot p_H}{d_{Det} \cdot \lambda} \qquad \text{and} \qquad DPC_V = -\frac{(\phi^{10}_{\text{sample}} - \phi^{10}_{\text{ref}}) \cdot p_V}{d_{Det} \cdot \lambda}. \tag{2}$$

Here, ϕ is in the reciprocal coordinates (f_x, f_y), and the change of coordinates $f_{xy} \to xy$ is obtained by noting that the new coordinates have a grid with step size $\Delta_x = p_H$ and $\Delta_y = p_V$. This also means that the spatial resolution of the technique is defined by the period p_H and p_V of the self image, which we refer as virtual pixel size. The values of p_H and p_V depends on the kind of grating (phase or absorption grating, π or $\pi/2$ phase shift, mesh or checkerboard grating) and on the beam divergence, and are discussed elsewhere [12, 14, 17].

The dark-field images are obtained by [5]

$$DF_H = \frac{I^{01}}{I^{00}} \qquad \text{and} \qquad DF_V = \frac{I^{10}}{I^{00}}, \qquad \text{where} \qquad I^{00} = \frac{a^{00}_{\text{sample}}}{a^{00}_{\text{ref}}}, \qquad I^{01} = \frac{a^{01}_{\text{sample}}}{a^{01}_{\text{ref}}}, \qquad I^{10} = \frac{a^{10}_{\text{sample}}}{a^{10}_{\text{ref}}}. \tag{3}$$

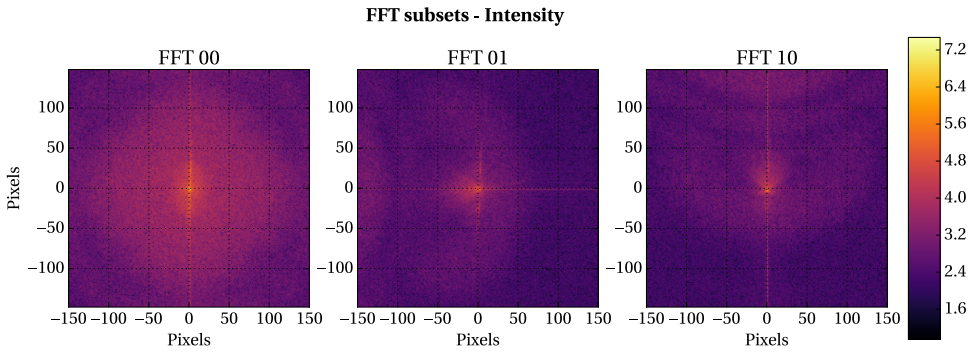

FIGURE 2. Cropped FT images around the central, first horizontal and first vertical harmonics (see harmonic labeling in Fig. 1.b. Note that the hamonic frequencies were shift to zero frequency, *i.e.*, $h_{1v} \rightarrow 0$ and $h_{1v} \rightarrow 0$. Magnitude plots. Color scale is a logarithmic scale.

This work concentrates on the differential phase imaging and wavefront measurements; the dark-field images will not be discussed in detail here.

For the wavefront measurements, the experiment is slightly different; two images of the direct beam (*i.e.* no sample) are measured at two Talbot distances, and the image at the closest distance is used as reference [22]. At this condition, the parameter d_{Det} in Eq. 2 is the difference between the distance of the two measurements.

wavepy - PYTHON LIBRARY

wavepy is a pure Python [19] library for data analysis of imaging and wavefront sensors using Talbot self imaging. The main goal of using Python is twofold: to take advantages of a widely used language in the scientific community, and also to make extensive use of the existing Python libraries for numerical calculation and plotting, namely *numpy* [23], *scipy* [24] and *matplotlib* [25]. Installation is facilitated by using *conda* [26] to solve dependencies. The development makes use of *git* for version control and collaborative development.The code is hosted in the *github* repository [27] `https://github.com/wavepy/wavepy`. One of the reasons for using Python is to be able to integrate *wavepy* with other projects, mainly *DXchange* [28] to seamlessly load images of different formats, and *TomoPy* [29] for tomography. The main code is a collection of functions that can be used in different ways, for instance inside a development environment (DE) or by creating a graphical interface that uses *wavepy* underneath. Examples using *Spyder* (`https://www.spyder-ide.org`) DE are available in the repository.

An example using one of the core functions is shown below:

```
>>> [int00, int01, int10,
     darkField01, darkField10,
     phi_x, phi_y] = single_2Dgrating_analyses(img, img_ref, [h1v, h1h])
```

In this example, `img` and `img_ref` are the *numpy* arrays containing the raw images for the sample and reference as acquired by the detector, and `h1v` and `h1h` are the first harmonic position in the FT images for the vertical and horizontal direction, respectively, calculated with Eq. 1. The function returns a list of *numpy* arrays containing respectively I^{00}, I^{01}, I^{10}, DF_H, DF_V, $(\phi^{01}_{\text{sample}} - \phi^{01}_{\text{ref}})$ and $(\phi^{10}_{\text{sample}} - \phi^{10}_{\text{ref}})$. Therefore, all the steps of properly cropping the FT images as discussed before are done automatically and the user can concentrate on the physical aspects of the results.

The exhaustive list of functions of *wavepy* is available at the application programming interface (API) reference in `https://wavepy.readthedocs.io` . The example above is the core function for data analysis, and several additional auxiliary functions are available. Some examples of the use of other functions are shown next.

APPLICATION - IMAGING The imaging capabilities of the interferometer has been used for metrology of individual refractive lenses for x-rays [8, 30]. By using grating interferometer the thickness profile is retrieved, providing a more complete picture than surface metrology. Fig. 1.a shows the raw image registered by the detector. By using `phi_x` and `phi_y` obtained with the function `single_2Dgrating_analyses` above, we can then obtain the differential phase contrast (DPC) images in Fig. 3 using

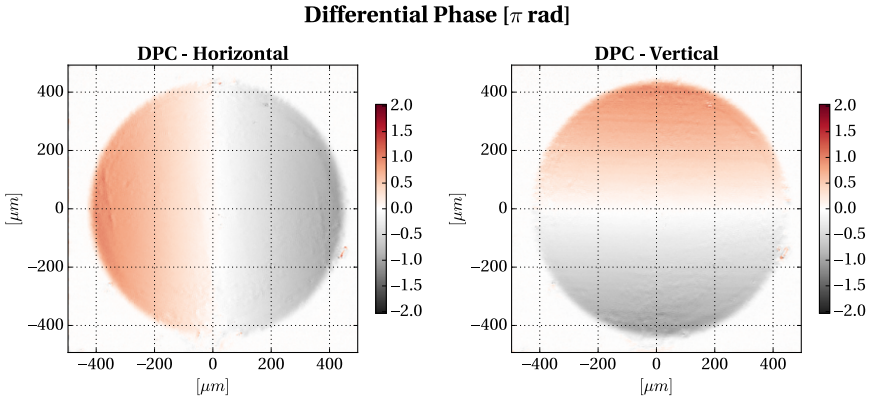

FIGURE 3. Differential phase contrast images of the diamond lens. Since the curved area of the lens has a parabolic profile, the DPC presents a linear change that is clearly visible in the figure.

```
1 >>> diffPhase01 = -phi_x*g_pitch_h/wavelength/d_grat_detector
2 >>> diffPhase10 = -phi_y*g_pitch_h/wavelength/d_grat_detector
3 >>> plot_DPC(diffPhase01, diffPhase10, [g_pitch_v, g_pitch_h])
```

where lines 1 and 2 above are applications of Eq. 2, and line 3 generates Fig. 3.

A common problem encountered in interferometry is phase wrap [31]. Numerical algorithms are available in Python for correcting wrapping (a.k.a *to unwrap*), however some cases can be challenging to correct this problem due to noisy data or when the sample present sharp edges. *wavepy* uses a Python library for unwrapping (`https://pypi.org/project/unwrap/`), but some user intervention may be necessary. In the case of lenses, phase unwrapping methods may add a constant shift to the values of the calculated phases ϕ^{01} and ϕ^{10}. In Fig. 3 this was corrected with the prior knowledge that for the center and the boundary areas $\phi^{01} = \phi^{10} = 0$, and therefore a proper offset could be applied.

An additional challenge is to combine the two DPC images into a single phase image [32]. Mathematically, this is the same as integrating two orthogonal gradient fields, $\frac{\partial\Phi(x,y)}{\partial x} = \Phi_x$ and $\frac{\partial\Phi(x,y)}{\partial y} = \Phi_y$ to obtain the surface $\Phi(x, y)$. For this to be possible Φ_x and Φ_y must be integrable, where integrability is defined as $\frac{\partial^2\Phi(x,y)}{\partial x\partial y} = \frac{\partial^2\Phi(x,y)}{\partial y\partial x}$ [33]. The challenge arises from the experimental noise in the measured gradients, which is different for the two gradients and therefore the integrability condition is not fulfilled. Several authors have studied the problem of calculating a surface from gradients where noise is present [34, 35], where the simplest method to be implemented herein is the method by Frankot and Chellappa [33]. This method is currently provided in *wavepy*. Works to include other methods are ongoing as a separate project hosted at `https://github.com/wcgrizolli/surface_from_gradient`.

Fig. 4 shows the integrated surface from the data in Fig. 3. In *wavepy* this is done as follows:

```
1 >>> phase = dpc_integration(diffPhase01, diffPhase10, virtual_pixelsize)
2 >>> delta, _ = get_delta(phenergy, material="C")
3 >>> thickness = -(phase - np.min(phase))*2*np.pi/wavelength/delta
4 >>> plot_integration(thickness*1e6, virtual_pixelsize,
                       titleStr=r'Thickness Diamond $[\mu m]$')
```

First, the Frankot-Chellappa algorithm is applied in line 1 to the DPC data. Next, by using the refractive index $n(\lambda) = 1 - \delta(\lambda) + i\beta(\lambda)$ value from a database, the phase shift is converted to sample thickness $T(x, y)$ by

$$T(x,y) = -\frac{\lambda}{2\pi}\frac{\Phi(x,y)}{\delta(\lambda)}, \tag{4}$$

where λ is the wavelength of the radiation, in this case $\lambda = 1.55$Å (photon energy of the radiation 8 KeV). This is done in line 3 using the value of $\delta = 6.503 \cdot 10^{-6}$ for carbon, obtained in line 2 using an *wacepy* function, which in turn gets that value from *xraylib* [36], a library for X-ray-matter interactions. The resulting sample thickness is plotted in line 4 and shown in Fig. 4.

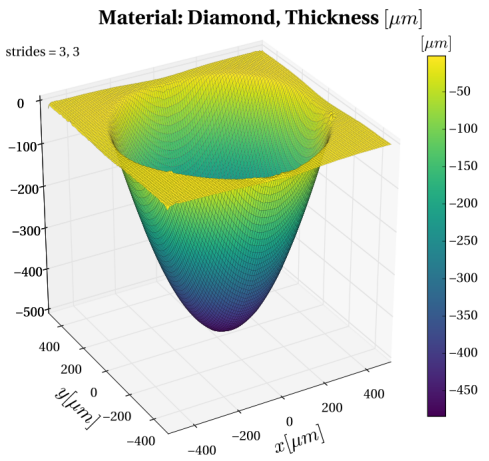

FIGURE 4. Relative thickness of the diamond lens, obtained by integrating the DPC images of Fig. 3.

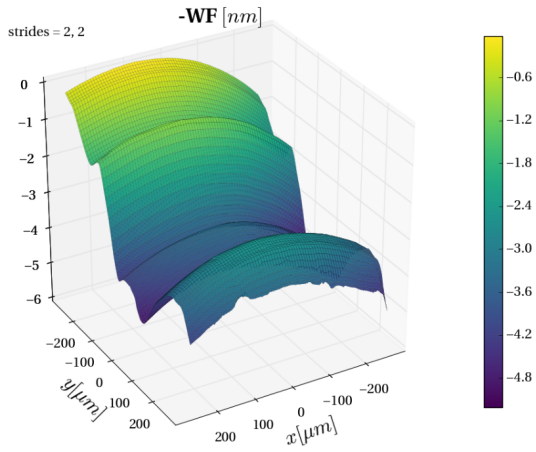

FIGURE 5. Measured phase variation across an X-ray wavefront.

APPLICATION - WAVEFRONT CHARACTERIZATION The measurement of the wavefront is done by acquiring images at two propagation distances along the beam [22]. The wavefront measurement can also exhibit phase wrapping, but wavefronts are in general smooth and the phase wrapping is unlikely. Fig. 5 shows a wavefront measured at the APS 1-BM-B beamline [20]. For this measurement, a testing mirror was placed in the beam path and the interferometer was placed 2 meters downstream. The high frequency oscillations in $\hat{y}$ directions are due to surface profile errors in the mirror, a first generation bi-morph mirror that was decommissioned after being used for more than 10 years. By using wavefront propagation calculations, one could use this data to retrieve the mirror surface errors and correct it by using the bi-morph actuators. Or, alternatively, one can aim to correct the wavefront, which does not require wavefront propagation calculations and directly improve the beamline performance.

Conclusion

In this work we discuss the development of *wavepy*, a Python library dedicated to data analysis of Talbot grating interferometry. We illustrate the use of the library with two examples, phase imaging and wavefront characterization, and show the use of few of the main library functions. In addition to the examples showed here, several other dedicated functions are provided. Examples are available in the repository and extensive documentation has been prepared. Dedicated functions for plotting meaningful physical quantities have been added, saving time when visualizing the results. The library also covers techniques that have similar experimental setup and data analysis procedures, such as phase-stepping imaging and coherence measurement.

ACKNOWLEDGMENTS

This work was supported by the US Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357. The authors would like to thank Dr. Albert Macrander and Dr. Michael Wojcik for their technical support during beamtime. We also acknowledge Drs. Brian Toby, Zichao Di, Francesco de Carlo and Doga Gursoy for their help with the software development.

REFERENCES

[1] L. Assoufid, X. Shi, S. Marathe, E. Benda, M. J. Wojcik, K. Lang, R. Xu, W. Liu, A. T. Macrander, and J. Z. Tischler, Review of Scientific Instruments **87**, p. 052004 (2016).

[2] T. Weitkamp, A. Diaz, C. David, F. Pfeiffer, M. Stampanoni, P. Cloetens, and E. Ziegler, Optics express **13**, 6296–6304 (2005).
[3] A. Momose, H. Kuwabara, and W. Yashiro, Applied Physics Express **4**, 1–3 (2011).
[4] J. Goodman, *Introduction to Fourier Optics*, 2nd ed. (Roberts & Company, 2005).
[5] F. Pfeiffer, M. Bech, O. Bunk, T. Donath, B. Henrich, P. Kraft, and C. David, Journal of Applied Physics **105**, 1–4 (2009).
[6] F. Pfeiffer, M. Bech, O. Bunk, P. Kraft, E. F. Eikenberry, C. Brönnimann, C. Grünzweig, and C. David, Nature Materials **7**, 134–137 (2008).
[7] T. Weitkamp, B. Nöhammer, A. Diaz, C. David, and E. Ziegler, Applied Physics Letters **86**, 1–3 (2005).
[8] W. C. Grizolli, X. Shi, L. Assoufid, T. Kolodziej, and Y. Shvyd'ko, *Advances in Metrology for X-Ray and EUV Optics VII*, p. 1 (2017).
[9] J.-P. Guigay, S. Zabler, P. Cloetens, C. David, R. Mokso, and M. Schlenker, Journal of Synchrotron Radiation **11**, 476–482nov (2004).
[10] F. Pfeiffer, O. Bunk, C. Schulze-Briese, A. Diaz, T. Weitkamp, C. David, J. F. van der Veen, I. Vartanyants, and I. K. Robinson, Physical Review Letters **94**, p. 164801 (2005).
[11] A. Diaz, C. Mocuta, J. Stangl, M. Keplinger, T. Weitkamp, F. Pfeiffer, C. David, T. H. Metzger, and G. Bauer, Journal of Synchrotron Radiation **17**, 299–307 (2010).
[12] S. Marathe, X. Shi, M. J. Wojcik, N. G. Kujala, R. Divan, D. C. Mancini, A. T. Macrander, and L. Assoufid, Optics Express **22**, 14041–14053 (2014).
[13] S. Marathe, X. Shi, M. J. Wojcik, A. T. Macrander, and L. Assoufid, Journal of Visualized Experiments 1–8 (2016).
[14] I. Zanette, "Interférométrie X à réseaux pour l'imagerie et l'analyse de front d'ondes au synchrotron," PhD Thesis, Université de Grenoble 2011.
[15] H. Itoh, K. Nagai, G. Sato, K. Yamaguchi, T. Nakamura, T. Kondoh, C. Ouchi, T. Teshima, Y. Setomoto, and T. Den, Optics Express **19**, p. 3339feb (2011).
[16] I. Zanette, S. Rutishauser, C. David, and T. Weitkamp, AIP Conference Proceedings **1365**, 325–328 (2010).
[17] I. Zanette, C. David, S. Rutishauser, T. Weitkamp, M. Denecke, and C. T. Walker, AIP Conference Proceedings **1221**, 73–79 (2010).
[18] X. Shi, S. Marathe, M. J. Wojcik, N. G. Kujala, A. T. Macrander, and L. Assoufid, Applied Physics Letters **105**, p. 041116 (2014).
[19] Python Software Foundation, Python Language Reference version 3.5, `http://www.python.org`.
[20] A. Macrander, M. Erdmann, N. Kujala, S. Stoupin, S. Marathe, X. Shi, M. Wojcik, D. Nocher, R. Conley, J. Sullivan, K. Goetze, J. Maser, and L. Assoufid, AIP Conference Proceedings **1741**, p. 030030 (2016).
[21] M. Takeda, H. Ina, and S. Kobayashi, Journal of the Optical Society of America **72**, p. 156 (1982).
[22] S. Berujon, E. Ziegler, and P. Cloetens, Journal of Synchrotron Radiation **22**, 886–894 (2015).
[23] S. van der Walt, S. C. Colbert, and G. Varoquaux, Computing in Science & Engineering **13**, 22–30 (2011), `http://www.numpy.org/`.
[24] SciPy: Open source scientific tools for Python, Online: `http://www.scipy.org/`.
[25] J. D. Hunter, Computing In Science & Engineering **9**, 90–95 (2007).
[26] Continuum Analytics, Inc., Conda - Package, dependency and environment management, `https://www.anaconda.com`.
[27] W. Grizolli, wavepy: Open source Python library for grating interferometry, `https://github.com/wavepy/wavepy` (2018).
[28] F. De Carlo, D. Gürsoy, F. Marone, M. Rivers, D. Y. Parkinson, F. Khan, N. Schwarz, D. J. Vine, S. Vogt, S.-C. Gleber, S. Narayanan, M. Newville, T. Lanzirotti, Y. Sun, Y. P. Hong, and C. Jacobsen, Journal of Synchrotron Radiation **21**, 1224–1230 (2014).
[29] D. Gürsoy, F. De Carlo, X. Xiao, and C. Jacobsen, Journal of Synchrotron Radiation **21**, 1188–1193 (2014).
[30] T. Kolodziej, S. Stoupin, W. Grizolli, J. Krzywinski, X. Shi, K.-J. Kim, J. Qian, L. Assoufid, and Y. Shvyd'ko, Journal of Synchrotron Radiation **25**, 354–360 (2018).
[31] D. Ghiglia and M. Pritt, *Two-dimensional phase unwrapping: theory algorithms and software*, Wiley-Interscience publication (Wiley, 1998).
[32] C. Kottler, C. David, F. Pfeiffer, and O. Bunk, Optics Express **15**, p. 1175 (2007).
[33] R. Frankot and R. Chellappa, IEEE Transactions on Pattern Analysis and Machine Intelligence **10**, 439–451 (1988).
[34] A. Agrawal, R. Raskar, and R. Chellappa, in *European Conference on Computer Vision* (2006), pp. 578–591.
[35] M. Harker and P. O'Leary, *IEEE Conference on Computer Vision and Pattern Recognition*, 1–7 (2008).
[36] A. Brunetti, M. S. del Rio, T. Ikonen, T. Schoonjans, M. Wormington, and D. Sagan, xraylib: a library for interactions of x-rays with matter, `https://github.com/tschoonj/xraylib` (2018).