Acceleration of phase retrieval algorithms through intermediate phase unwrapping

Oleg Soloviev[a,b]

[a]Flexible Optical BV, Polakweg 10–11, 2288 GG Rijswijk, the Netherlands
[b]DCSC, ME, TU Delft, Mekelweg 2, 2624 CD Delft, the Netherlands

ABSTRACT

We investigate the enhancement of projection-based phase retrieval algorithms through intermediate phase unwrapping steps. In the phase retrieval problem, convergence to local minima presents a serious obstacle for standard iterative algorithms. Wrong solutions are typically characterized by the presence of phase residues, while the ground truth phase is often residue-free. We present our first results in introducing phase unwrapping as an intermediate step in the projection-based algorithms, demonstrated on simulated and experimental data. The method can reduce convergence time or produce a physically meaningful solution where standard algorithms fail. The proposed method can find applications in microscopy, characterization of precise optical instruments, and wavefront sensing.

Keywords: Phase retrieval, phase unwrapping, DRAP algorithm, alternating projections, wavefront sensing, regularization

1. INTRODUCTION

Phase retrieval is a fundamental inverse problem in optics, arising in applications ranging from wavefront sensing and adaptive optics to coherent diffraction imaging and microscopy.[1,2] The problem consists of recovering the phase of a complex field from its intensity measurements in the pupil and image planes. Mathematically, this can be formulated as finding a complex field x such that $|x| = a$ and $|\mathcal{F}x| = A$, where $\mathcal{F}$ denotes the Fourier transform and a, A are the real arrays representing the measured amplitudes.

A large class of phase retrieval algorithms is based on alternating projections between constraint sets. These include the famous Gerchberg-Saxton algorithm[3] which alternates between pupil and image constraints, more advanced methods like Fienup's Hybrid Input-Output (HIO) algorithm,[4] and a number of modern alternatives like RAAR[5] and DRAP,[6] (see Ref. 7 for a review).

Despite their widespread use, projection-based algorithms suffer from convergence to local minima due to the ill-posed nature of the phase retrieval problem. Specifically, the sets defined by the amplitude constraints are non-convex, and in practice they often do not have a common point due to noise and model mismatch. This leads to stagnation or convergence to incorrect solutions. The probability of convergence to a wrong solution is especially high when the phase amplitude is large; in such cases, the algorithm can stagnate even in the ideal noiseless case.

This work was motivated by encountering phase retrieval failure on real experimental data from deformable mirror measurements with large aberrations. Standard DRAP failed across all tested parameter combinations, image preprocessing, and initializations, with visual inspection revealing persistent phase residues correlating with failure.

Given the *a priori* knowledge that the true phase is smooth, we applied least-squares unwrapping algorithms[8] to the retrieved phase. We observed that although the unwrapped phase field satisfied the amplitude constraints less well in the focal plane, the elimination of residues brought the phase closer to the ground truth (in cases where it was known) in some pupil regions. Restarting the algorithm from this unwrapped phase field produced

Further author information:
E-mail: oleg@okotech.com

Quantitative Phase Imaging XII, edited by Yang Liu, YongKeun Park,
Proc. of SPIE Vol. 13861, 138610F · 2026 Published by SPIE
1605-7422 · doi: 10.1117/12.3079354

solutions with fewer residues, and iterating this process led to convergence to the correct solution after several cycles. These observations suggested exploring phase unwrapping as a regularization strategy.

In this work, we demonstrate that intermediate phase unwrapping (after every N iterations) improves the convergence of the DRAP algorithm. Results show iteration reduction of 10–25× on simulated data, with higher impact on poorly initialized cases. More significantly, the enhancement enables convergence on experimental data that completely resisted standard methods.

2. METHODS

We consider here the problem of optical phase retrieval (PR), that is, the recovery of the optical aberration (phase) in the pupil of an optical system from the recorded image of a point source, the point spread function (PSF). Although the methods described here can be applied also to the phase-diverse phase retrieval problem, we focus here on the single-plane phase retrieval only.

We recap below the main ingredients of projection-based phase retrieval algorithms, the DRAP algorithm in particular, and then describe the proposed enhancement through intermediate phase unwrapping steps.

2.1 Phase Retrieval as Feasibility Problem

The PR problem can be formulated as finding a complex field $x \in \mathbb{C}^{M\times N}$ such that

$$|x| = a, \quad |\mathcal{F}x| = A, \tag{1}$$

where $a, A \in \mathbb{R}^{M\times N}$ are measured amplitude distributions in the pupil and image planes, respectively, and $\mathcal{F}$ denotes the discrete Fourier transform. This can be interpreted as a feasibility problem of finding a point in the intersection of two constraint sets defined by equations (1). It should be noted, however, that in practice these sets often do not intersect due to noise and model mismatch, leading to an infeasible problem; thus one looks for a point that is closest in some sense to these sets.[7]

Projection-based algorithms start with some initial guess x_0 and iterate between points from these sets using their projection operators P_a and P_A; a more generic iteration can be written as $x_{k+1} = T(x_k)$, where T is a fixed-point operator constructed from the projections. For example, the Gerchberg-Saxton (GS) algorithm[3] proceeds with $T_{\mathrm{GS}} = P_a P_A$. As it can be written as $y_{k+1} = P_A x_k$, $x_{k+1} = P_a y_{k+1}$, this method is also known as the alternating projections (AP) algorithm. Another example is the Fienup's hybrid input-output (HIO) algorithm,[4] which can be written as $T_{\mathrm{HIO}} = P_a(2P_A - I) + (I - P_A)$, where I is the identity operator. This is also known as the Douglas-Rachford (DR) algorithm in the context of feasibility problems.

Projection-based algorithms iterate until the relative change between iterations falls below a threshold or a maximum number of iterations is reached. In numerical experiments, we assess convergence using the root-mean-square error (RMSE) between the retrieved phase and the ground truth phase when available. For the single-plane phase retrieval problem, the sign of the even part of the phase is not observable, so the algorithm may converge to the twin phase, $\phi_{\mathrm{twin}}(r) = -\phi_{\mathrm{gt}}(-r)$, where ϕ_{gt} is the ground truth phase and r is the position vector in the pupil plane. We compute the RMSE with respect to both the ground truth and the twin phase, and use the minimum of these two values as the convergence metric.

Another metric is the distance from the current approximation to the sets, or the norm of the difference between the current projections onto the two sets, $\|x_k - y_k\|_2$, or its relative version $\|x_k - y_k\|_2/\|x_k\|_2$.

2.2 DRAP Algorithm

The DRAP algorithm[6,9] combines the AP and DR operators through a convex combination with parameter $\beta \in [0, 1]$, with $\beta = 0$ corresponding to the AP algorithm, and $\beta = 1$ to the DR algorithm. DRAP has demonstrated good global convergence properties with a good convergence rate; for details on the DRAP formulation and convergence properties, we refer the reader to Ref. 6.

The parameter β controls the algorithm's behavior: small β produces stable convergence to the nearest local minima (see for example the AP curves starting at iteration #800 in Fig. 4), while large values explore the space for the global minimum and then linearly converge to it (see Fig. 3a), but can be less robust against noise. In practice, one runs DRAP with some high value of β to explore the space, and then switches to a smaller value or to the AP to ensure convergence.

2.3 Initial guess

Being a non-convex problem, phase retrieval algorithms are sensitive to the initial guess. A common choice is to start from a zero phase, that is, $x_0 = a$. Another popular choice is the "spectral" initialisation, which consists of taking the inverse Fourier transform of the square root of the measured PSF, thresholded at some value. This approach provides a better approximation of the true phase. Finally, random initial phases can be used to explore the solution space, especially when no prior information about the phase is available. Often, multiple runs with different random initialisations are performed to increase the chances of finding the global minimum.[10]

2.4 Image Preprocessing

In practice, the measured PSF images often contain noise and background signals that can adversely affect the convergence of phase retrieval algorithms. To mitigate these effects, a preprocessing step can be applied to the input images before feeding them into the algorithm. This preprocessing might include background subtraction, normalization, and thresholding to remove low-intensity noise.

In addition, cropping, zero-padding, and downsampling of the input PSF are often employed. Cropping the PSF to a smaller size around its central peak can help focus the algorithm on the most relevant data, reducing the influence of noise in the outer regions. However, cropping may discard relevant information about the field. Zero-padding the cropped PSF to a larger size can help reduce edge effects in the Fourier transform, improving the accuracy of the amplitude constraints. Downsampling the image can reduce high-frequency noise, but it also reduces the resolution of the data, which may impact the accuracy of the retrieved phase. These three steps affect the sampling of the phase in the pupil plane; as a rule of thumb, the number of informative pixels in the arrays a and A should be approximately equal after these preprocessing steps.

Preprocessing is a critical step that affects convergence. Its parameters can be considered hyperparameters of the algorithm that need to be tuned for each specific application.

2.5 Phase Unwrapping as Final Step

In optical PR, the retrieved phase is related to the wavefront aberration and needs to be unwrapped before being used in applications such as adaptive optics. If the phase is sampled frequently enough such that the phase jumps between adjacent pixels do not exceed π (Itoh's condition, equivalent to the Nyquist sampling criterion), the phase can be unwrapped straightforwardly using Itoh's algorithm.[11] It proceeds by calculating the wrapped phase gradient:

$$g = w(\nabla\psi), \tag{2}$$

where $\psi = \angle x$ is the wrapped phase of the complex field x, ∇ is the finite difference operator, and $w(\cdot)$ is the phase wrapping operator $w(\phi) = \phi + 2\pi k, k \in \mathbb{Z} : w(\phi) \in (-\pi, \pi]$. From the Itoh condition, it follows that this gradient field is equal to the true gradient of the unwrapped phase ϕ, and the phase can be recovered by integration. Noisy phase measurements can have phase residues—that is, points around which the wrapped phase jumps sum up to $\pm 2\pi$. The residues prevent unwrapping of the phase by making the gradient field g inconsistent, and one has to use more advanced algorithms.[8]

As shown in the next section, PR algorithms can produce phases which contain phase residues even in the noiseless feasible case, for instance when the initial phase error has a large amplitude.

If it is known *a priori* that the phase is residue-free, one can use least-squares phase unwrapping. Here, the gradient field g from Eq. (2) is integrated in the least-squares sense, namely by finding a potential ϕ_u that minimizes

$$\phi_u = \min_{\phi} \|\nabla\phi - g\|_2^2. \tag{3}$$

In our work, we have used an approach based on the Talmi-Ribak algorithm[12] adjusted for a finite aperture as described in Ref. 13.

2.6 Phase Unwrapping as Regularization Step

Traditionally, phase unwrapping terminates the algorithm chain as a post-processing step. However, we propose to use it differently: as an intermediate regularization step within the iterative process.

Once the algorithm has converged to a solution $x^{(1)} = a\mathrm{e}^{\mathrm{i}\phi^{(1)}}$, the unwrapped phase $\phi_u^{(1)}$ can be substituted back into the complex field to form a new starting point for the phase retrieval algorithm:

$$x_0^{(2)} = a\mathrm{e}^{\mathrm{i}\phi_u^{(1)}}. \tag{4}$$

This operation can be interpreted as a "projection" onto the set of fields with residue-free phases. Due to the non-linearity of the wrapping function w, neither the fields with residue-free phases nor residue-free phases themselves form a convex set, which prevents the use of rigorous "projection" terminology. Nevertheless, the operation defined by Eq. (4) wraps the unwrapped phase back, leaving fields with residue-free phases unchanged. For complex fields with phases containing residues, it finds a nearby point with a residue-free phase. Thus, the unwrapping step defined by Eq. (4) acts as a smoother or regularizer.

A plausible explanation for the regularizing effect can be given by the following argument. Consider for simplicity a constant amplitude a in the pupil plane. From a discrete sampling perspective, the projection P_a can produce a complex array with a phase residue, and the amplitude constraint appears satisfied at each pixel. However, physically, a phase residue requires the continuous complex field amplitude to vanish at some point within the 2×2 pixel region surrounding the residue. Thus, the constraint $|x| = a$ cannot be satisfied for the continuous field in that region. By replacing the phase with its unwrapped version, we eliminate the residue and thereby restore physical consistency.

It is not the goal of this work to provide a rigorous mathematical proof of the regularizing property of the unwrapping step. Note that the wrapped gradient projection onto curl-free vector fields *is* rigorous; the phase regularization step is the pragmatic interpretation.

The unwrapping step need not be inserted after the algorithm has converged. Instead, we propose to apply it every N iterations as an intermediate step, effectively adding a physical consistency constraint to the iterative process. This approach aims to prevent the algorithm from stagnating in configurations with phase residues, steering it towards physically plausible solutions.

In the next section we demonstrate that indeed early and periodic unwrapping accelerates convergence of the DRAP algorithm.

3. RESULTS

We demonstrate the results of the proposed enhancement on both simulated and experimental data. The simulations include both feasible and infeasible cases, while the experimental data consists of historical measurements archived from previous projects, obtained on different deformable mirror measurement setups.

As shown in the previous section, the algorithms for phase retrieval usually comprise a number of steps, which are adjusted to the specific application. It is important to note that with a number of hyperparameters to be tuned, there is no universal recipe that works best in all situations. This makes the comparison of different algorithms difficult, as the performance depends not only on the algorithm itself, but also on the choice of these hyperparameters. Where possible, we have tuned the baseline algorithm hyperparameters and used the same values for both methods to ensure a fair comparison.

3.1 Input Data

For the simulations, we modelled an optical setup consisting of a beam with 10mm diameter footprint focused with a 300 mm focal length lens, with the PSF registered using a UI-1240 camera (5.3 μm pixel size, 1280×1024 pixels, 8-bit depth). The ground truth phase is a smooth aberration composed of low-order Zernike polynomials with a peak-to-valley (PV) value of approximately 5 wavelengths as shown in Fig. 1. These parameters were chosen such that the aperture in the pupil array is large enough to produce numerous phase residues even in the noiseless feasible case, making it an ideal test case for demonstrating the unwrapping enhancement. The forward and inverse problem experiments are performed using the PhaseRetrieval.jl package.[14]

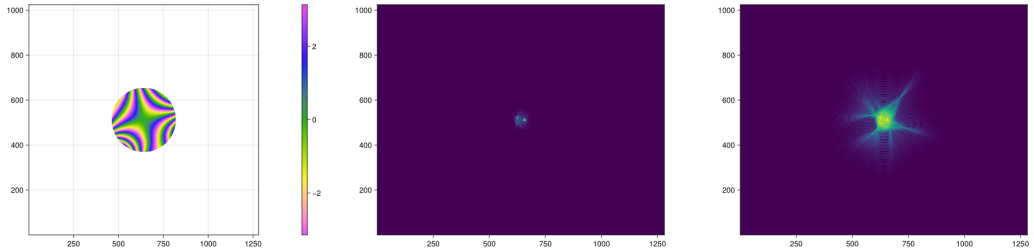


Figure 1: Ground truth phase, noiseless simulated PSF (shown in linear and logarithmic scales).

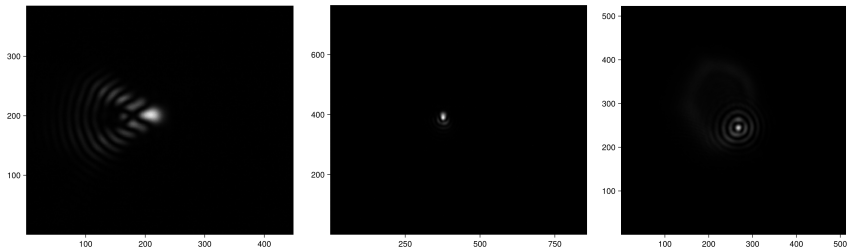


Figure 2: Examples of the experimental raw images used in the tests: obtained with setup 1 (left, middle) and setup 2 (right).

The experimental datasets consist of PSF measurements from two different optical setups, as illustrated in Fig. 2. These measurements were obtained from deformable mirror characterization experiments with different aberration amplitudes and numerical apertures. Setup 1 used a UI-1490 camera with a 300mm focal length lens and a 10mm beam diameter, similar to the simulated setup but with three times finer pixel resolution. Setup 2 used a UI-3860, a 12-bit camera with 2.9μm pixel size, a 750mm focal length lens, and a 33mm beam diameter.

3.2 Simulated Images, Feasible Case

For the feasible case, the input images represent exactly the absolute value of the Fourier transform squared of the pupil field:

$$A = |\mathcal{F}(a\mathrm{e}^{\mathrm{i}\phi_{\mathrm{gt}}})|, \tag{5}$$

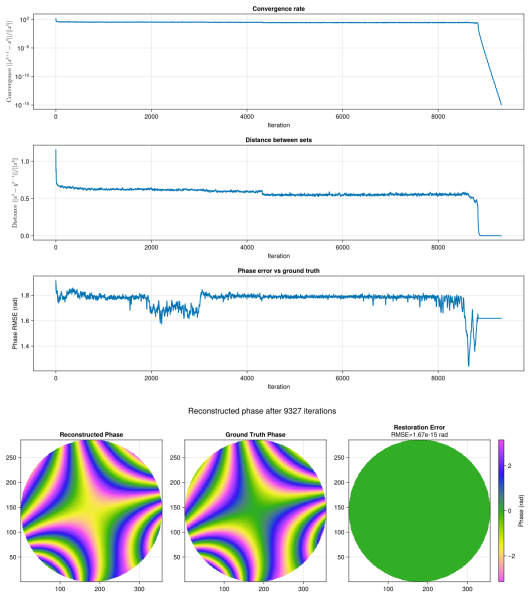
where a is the amplitude and ϕ_{gt} is the ground truth phase, so the PR problem has an exact solution $x = a\mathrm{e}^{\mathrm{i}\phi_{\mathrm{gt}}}$.

Due to nonconvexity, the algorithm may still fail to converge to the exact solution even in this ideal case. Moreover, due to accumulated numerical error, the algorithm may converge to different values on different machines or when different numerical libraries are used (*e.g.,* changing the number of the threads in BLAS library used to perform the FFT can change the convergence results).

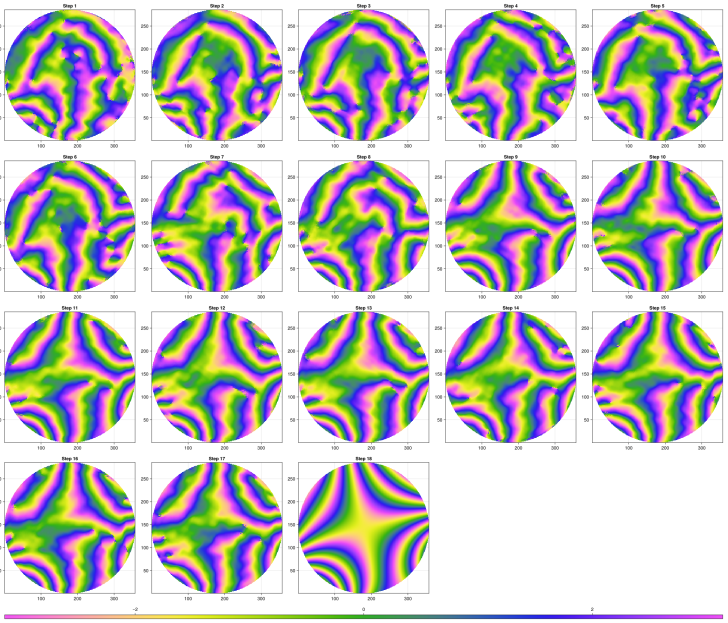
As a baseline, we run the DRAP algorithm with $\beta = 0.9$ on the simulated feasible case, starting from the zero initial phase. As seen from Fig. 3, the algorithm converges to the true solution after 9327 iterations, when the relative update became smaller than 10^{-16}. The retrieved phase matches the ground truth phase perfectly (up to a global piston).

It is interesting to inspect the intermediate stages of the algorithm (Fig. 3b) together with the "Distance between sets" curve. We can see several stages: first, the algorithm quickly converges to a local minimum with many phase residues, which are recognizable on the phase snapshots as the points inside the aperture where the phase fringes originate and terminate. After that, the algorithm slowly eliminates these residues one by one (note a sudden drop in the distance between sets curve around iteration #4000 and how it corresponds to residue removal in the upper half of the phase), and struggles to eliminate the last few residues, finally converging to the true solution. This suggests that eliminating the residues at an earlier stage may accelerate convergence, which we explore next.

Figure 4 shows the results of adding an intermediate unwrapping step after every 200 iterations of the same DRAP algorithm. With this enhancement, starting from the same zero phase initial guess, the algorithm needs only 803 iterations to converge, representing an 11× acceleration compared to the baseline.



(a) Convergence curves and retrieved phase



(b) Intermediate stages of the $\angle x_k$ for every 500 iterations

Figure 3: Baseline results: DRAP with $\beta = 0.9$ on the simulated feasible case (without intermediate unwrapping). The phase is perfectly reconstructed (up to a global piston) after 9327 iterations, when the relative update became smaller than 10^{-16}.

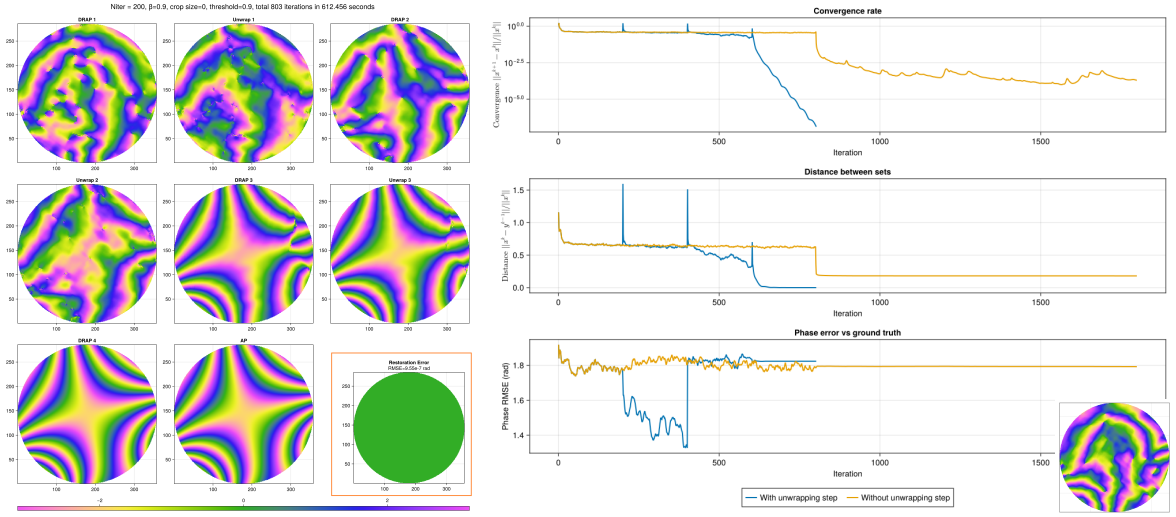


Figure 4: Phase evolution and final retrieved phase error shown in the inset plot (left), convergence curves (right) for DRAP with intermediate unwrapping every 200 iterations on the simulated feasible case for $\beta = 0.9$. The phase is perfectly reconstructed (up to a global piston) after 803 (4×200 DRAP + 4 intermediate unwrapping) iterations, when the relative update became smaller than 10^{-6}. For comparison, the DRAP without unwrapping was run also for 800 iterations and then continued with Gerchberg-Saxton (AP) for 1000 more iterations, still failing to converge to the true solution (inset).

From the evolution of the phase and the convergence curves in Fig. 4, we see that the unwrapping step acts as a regularizer, or a smoother: although it temporarily increases the error with respect to the amplitude constraints, it eliminates the residues and steers the algorithm towards the true solution.

3.3 Simulated Images, Infeasible Case

Now we proceed with the infeasible case, where the input images were obtained by cropping the PSF to a smaller size, adding Gaussian noise with zero mean and standard deviation of two graylevels, and 8-bit quantization. We consider two cropping sizes: 100×100 pixels and 256×256 pixels.

For both crop sizes, we run the DRAP algorithm with $\beta = 0.9$ starting from the "spectral" initial phase with threshold level 0.9.

For the crop size of 100×100 pixels, the baseline DRAP algorithm without unwrapping needed about 12000 iterations to converge. In contrast, with the intermediate unwrapping step every 1000 iterations, the algorithm converges after only 5000 iterations, representing a 2.4× acceleration in number of iterations. Figure 5, top shows the results of the enhanced and baseline algorithms run for the same number of iterations.

The wall time for the enhanced algorithm was about 3.5 seconds, compared to 6.2 seconds for the baseline, representing a speedup of 1.7×. This is because the unwrapping steps require additional computations; however, no optimization of the code or precision of the unwrapping algorithm was performed, and we expect that with further improvements the wall time speedup can approach the iteration speedup.

A similar improvement is observed for the crop size of 256×256 pixels. Here, the baseline DRAP algorithm without unwrapping needed about 15000 DRAP iterations (plus finishing 1000 iterations of AP) to converge, while with the intermediate unwrapping step every 500 iterations, the algorithm converges after 10 periods, that is, 5000 DRAP iterations (plus finishing 1000 iterations of AP), representing a 2.66× acceleration in number of iterations; 27 seconds against 63 seconds in wall time, respectively. Figure 5, bottom shows the results of the enhanced and baseline algorithms run for the same number of iterations.

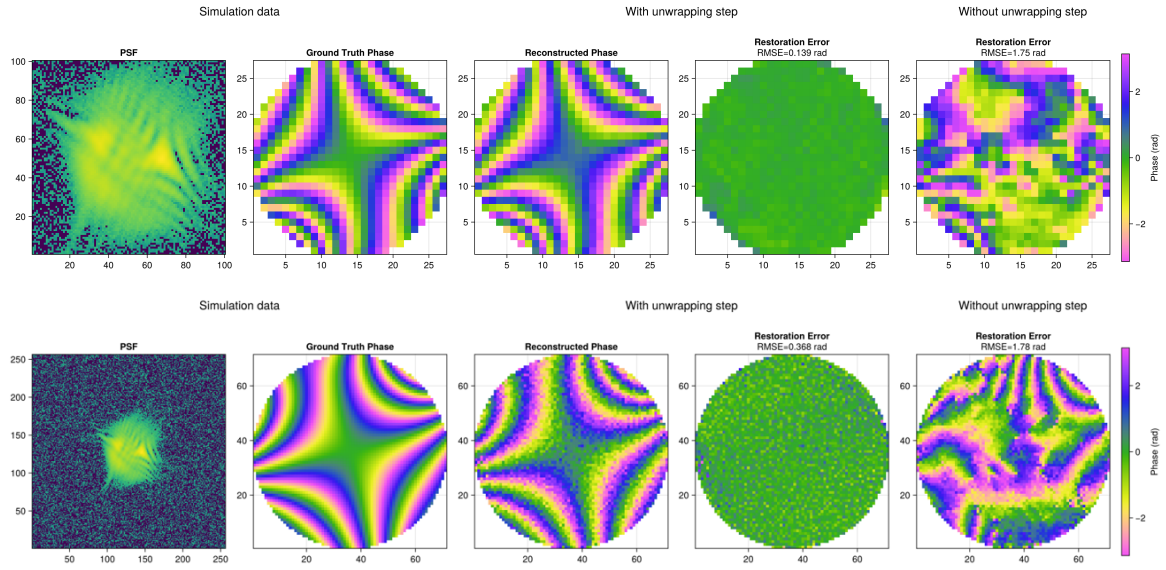


Figure 5: Results of the proposed method for simulated noisy cropped images compared with the DRAP method run for the same number of iterations. Top: crop size=100, 5 periods of length $N = 1000$, and ≤ 1000 steps of AP. Bottom: crop size = 256, 10 periods of length $N = 500$, and ≤ 1000 steps of AP. Without intermediate unwrapping, the algorithm needs about 12000 (top) and 15000 (bottom) steps to converge.

3.4 Experimental Images

We present three representative examples from the experimental datasets, demonstrating different typical behavior of the proposed enhancement: positive effect, negative effect, and neutral.

The first set of measurements was obtained with setup 1. The aberration, a low-order Zernike polynomial of moderate amplitude, was introduced by a pre-calibrated deformable mirror. In this case, both the baseline DRAP

algorithm and the enhanced algorithm with intermediate unwrapping quickly converged to similar solutions after 150–1000 iterations, depending on the preprocessing parameters and the aberration itself.

The enhancement effect of the intermediate unwrapping step was neutral or even negative in this case, as shown in Fig. 6 and Fig. 7. This can be explained by the fact that the aberrations were small in amplitude, the number of informative pixels in the PSF was large compared to the number of unknowns (pixels in the pupil), and thus it was easy to tune the baseline algorithm parameters and ensure it converged quickly without getting stuck in local minima with phase residues. In such cases, the additional unwrapping step may even slow down convergence, as it temporarily increases the error with respect to the amplitude constraints; it also introduces additional computations and increases wall time.

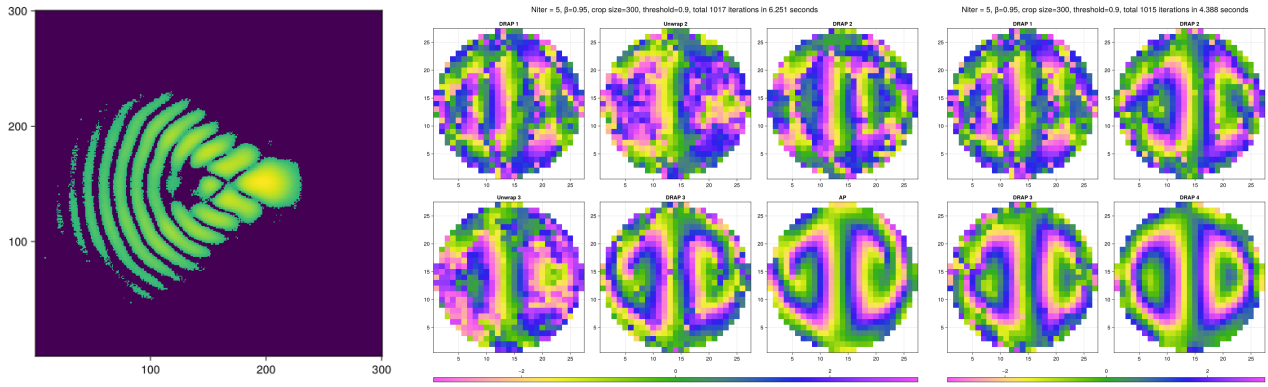


Figure 6: Example where the proposed enhancement had a reverse effect. Measured preprocessed PSF, in log scale (left); phase evolution using the enhanced algorithm with intermediate unwrapping (middle) and the baseline algorithm without unwrapping (right).

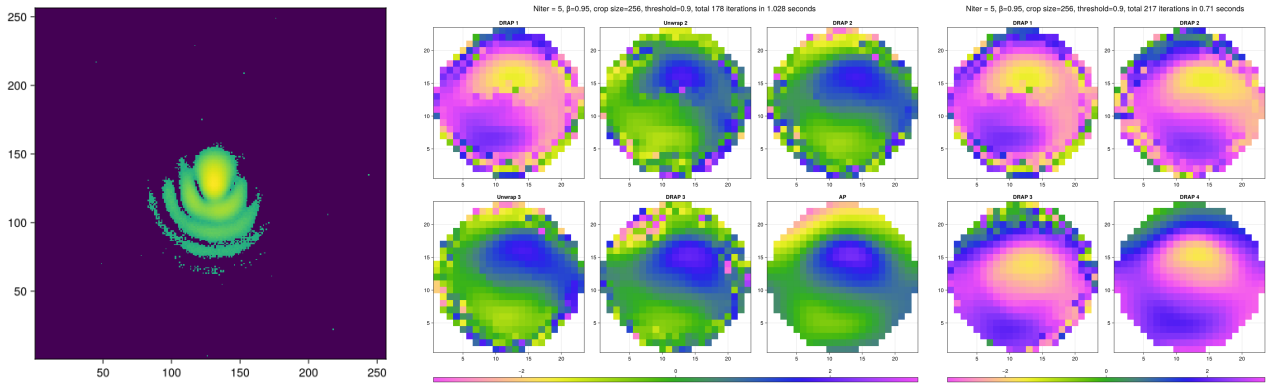


Figure 7: Example where the proposed enhancement had no effect. Measured preprocessed PSF, in log scale (left); phase evolution using the enhanced algorithm with intermediate unwrapping (middle) and the baseline algorithm without unwrapping (right).

The most interesting results of the proposed enhancement were obtained on experimental data from setup 2 (Fig. 2, right). The aberration resulted from the response function of a deformable mirror. Due to a large curvature of the initial shape of the mirror, the baseline DRAP algorithm failed to converge for all tested β values and preprocessing parameters, even after 30000 iterations. The reconstructed phases with the baseline algorithm featured many phase residues as shown in Fig. 8 (f). However, with the intermediate unwrapping step it was quite easy to tune the parameters to obtain a physically plausible solution, corresponding to an interferometric measurement of the same response function. The results are shown in Fig. 8.

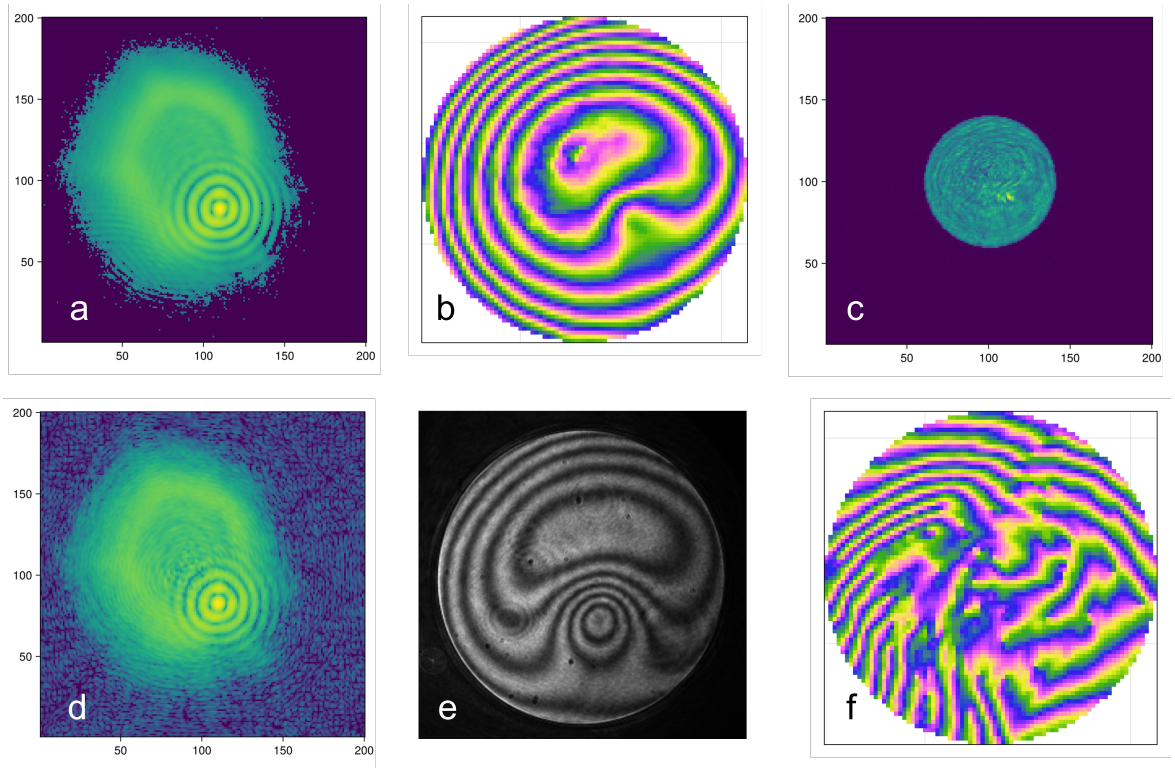


Figure 8: Example where the proposed enhancement helped to get out of a local minimum. a) Measured preprocessed PSF, in log scale; b) restored phase with the proposed algorithm using 15 periods of length $N = 250+ < 3000$ iterations of AP; c) and d) restored intensities in the pupil and image planes; e) recorded interferogram of the same response function; f) example result of DRAP failing to converge after 30000 iterations.

4. DISCUSSION AND OUTLOOK

The proposed method has several important practical limitations. First, it assumes the true solution is residue-free (unwrappable), which does not hold in some applications. Second, the unwrapping frequency parameter N becomes an additional hyperparameter requiring tuning; further work is needed to establish guidelines for its selection. Third, the enhancement was demonstrated only on specific values of the DRAP parameter β; for comparison purposes, we have used those values of β where the baseline algorithm converged (if it did converge). Further work is needed to explore its effectiveness across the full parameter space and with other projection-based algorithms. Finally, the acceleration is most pronounced when baseline convergence is slow; rapidly converging problems see proportionally less benefit.

From a theoretical perspective, as noted in Section 2, the proposed unwrapping step is not aligned with the projection-based framework. It would be interesting to explore alternative formulations that integrate phase unwrapping with projection operators more rigorously, potentially leading to new algorithmic insights.

It is not immediately clear why the number of residues decreases after successful runs. Indeed, every new period begins with a residue-free phase, but after N iterations of DRAP, the phase may again contain residues. Can the eliminated residues reappear? If not, what is the mechanism of their elimination? These questions remain open for future research.

Although we demonstrated the method only on single-plane phase retrieval, it can be applied to other projection-based phase retrieval algorithms, including phase-diverse phase retrieval. Future work will explore these extensions and their practical implications.

5. CONCLUSIONS

We have introduced phase unwrapping as an intermediate step in projection-based phase retrieval algorithms. The unwrapping step acts as a regularizer, adding a physical consistency constraint that helps avoid local minima characterized by phase residues. Beyond the acceleration, the enhancement enables convergence in cases where standard algorithms fail completely.

Practical applications include wavefront sensing and correction in applications where no other wavefront measurements are available or where the elimination of the non-common path aberrations is critical.

ACKNOWLEDGMENTS

The project is supported by the Chips Joint Undertaking and its members including the top-up funding by RVO (The Netherlands Enterprise Agency).

REFERENCES

[1] Fienup, J. R., "Phase retrieval algorithms: a personal tour [Invited]," *Appl. Opt.* **52**, 45 (Jan. 2013).

[2] Luke, D. R., "Phase Retrieval, What's New?," *SIAG/OPT Views News* **25**(1), 1–6 (2017).

[3] Gerchberg, R. W., "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik* **35**, 237–246 (1972).

[4] Fienup, J. R., "Phase retrieval algorithms: a comparison," *Appl. Opt.* **21**(15), 2758–2769 (1982).

[5] Luke, D. R., "Relaxed averaged alternating reflections for diffraction imaging," *Inverse Problems* **21**, 37–50 (Nov. 2004).

[6] Thao, N. H., Soloviev, O., and Verhaegen, M., "Convex combination of alternating projection and Douglas–Rachford operators for phase retrieval," *Adv. Comput. Math.* **47**, 33 (June 2021).

[7] Luke, D. R., Sabach, S., and Teboulle, M., "Optimization on Spheres: Models and Proximal Algorithms with Computational Performance Comparisons," *SIAM Journal on Mathematics of Data Science* **1**(3), 408–445 (2019).

[8] Ghiglia, D. and Pritt, M., [*Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*], Wiley (1998).

[9] Thao, N. H., "A convergent relaxation of the douglas–rachford algorithm," *Computational Optimization and Applications* **70**, 841–863 (Mar. 2018).

[10] Marchesini, S., "A unified evaluation of iterative projection algorithms for phase retrieval," *Rev. Sci. Instrum.* **78**(1) (2007).

[11] Itoh, K., "Analysis of the phase unwrapping algorithm," *Applied Optics* **21**, 2470 (July 1982).

[12] Talmi, A. and Ribak, E. N., "Wavefront reconstruction from its gradients," *J. Opt. Soc. Am. A* **23**(2), 288–297 (2006).

[13] Soloviev, O., "PhaseUtils.jl: Phase unwrapping and wavefront utilities," (2024). Julia package documentation.

[14] Soloviev, O. and contributors, "PhaseRetrieval.jl: Phase retrieval algorithms in julia," (2024). Julia package.