Modeling Polarization for Phase Retrieval

Presented by:

OSA Polarization Technical Group

Modeling Polarization for Phase Retrieval

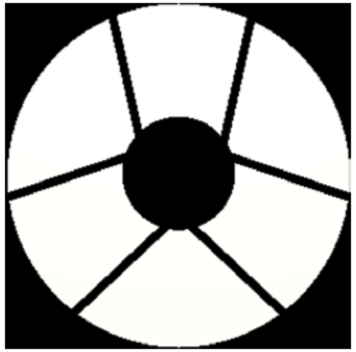
Scott Paine

Outline

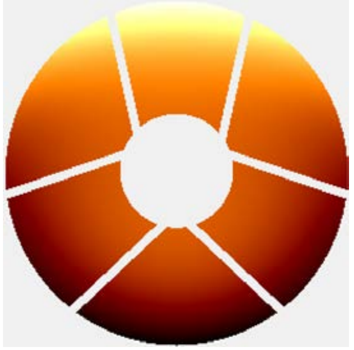
- Introduction to phase retrieval wavefront sensing
  - Gerchberg-Saxton methods
  - Nonlinear optimization
- Building a phase retrieval model
  - Scalar wavefront theory
  - Backpropagating error
- Incorporating polarization aberrations
  - Jones pupil
  - PSM
  - Pauli-Zernike coefficients
- Full Model for Polarization

Wavefront Sensing

- Want to recover unknown wavefront using PSF image



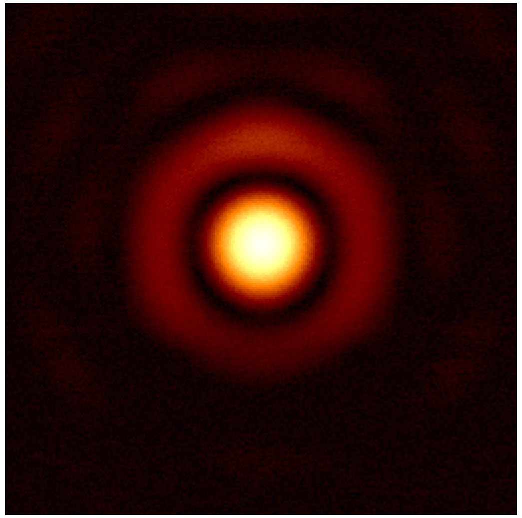
$A(u,v)$ (known)



$W(u,v)$ (unknown)

Optical Propagation
&
Detection

Observed PSF (with noise)



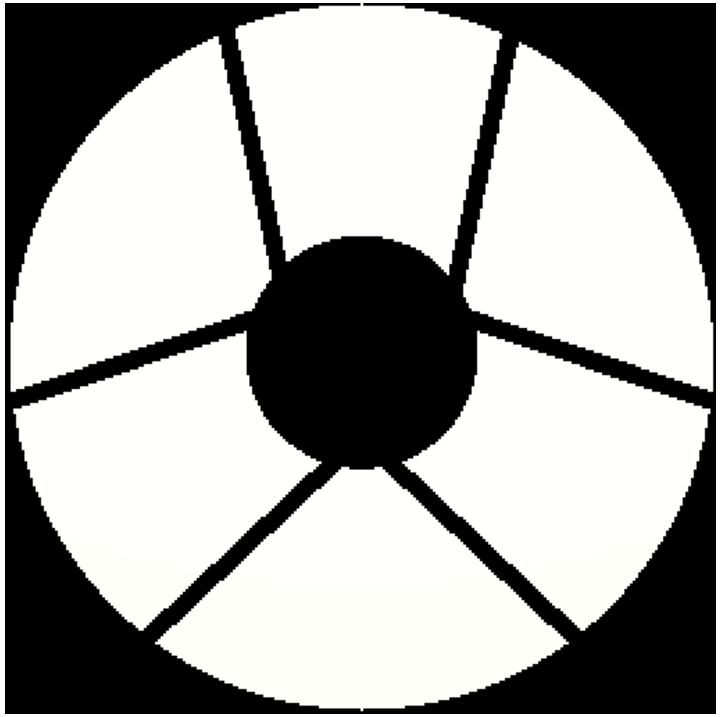
Wavefront Sensing

- Known quantities
  - Pupil function
  - Possibly prior wavefront knowledge (i.e. known defocus)
  - Sampling

- Measured quantities
  - PSF Intensity

Gerchberg-Saxton Algorithm

Known



Pupil Function

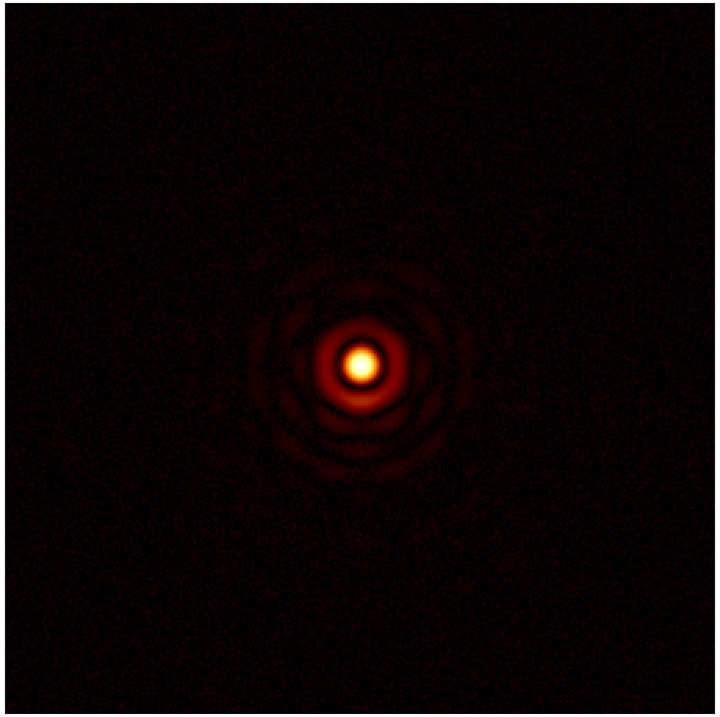
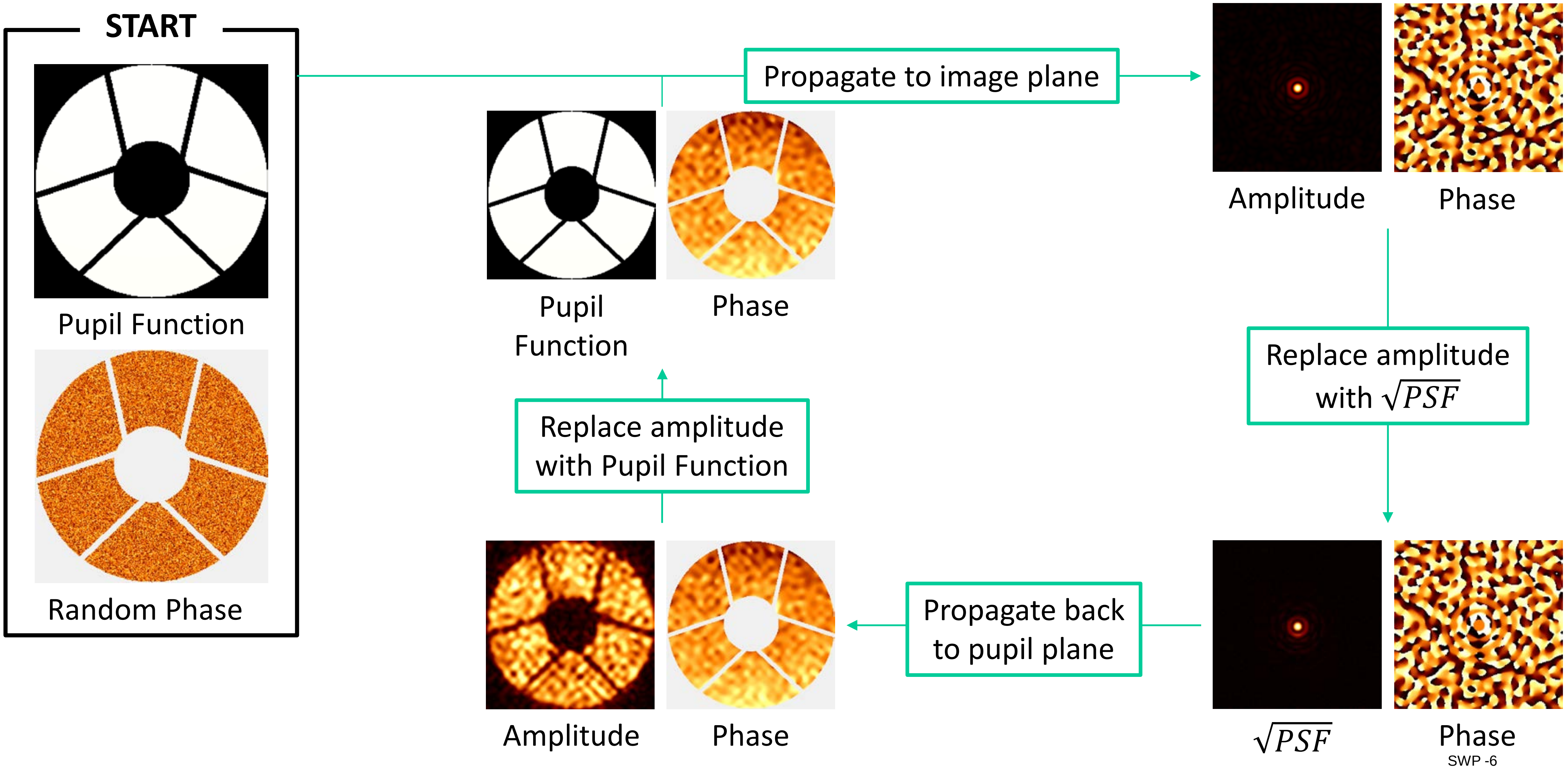
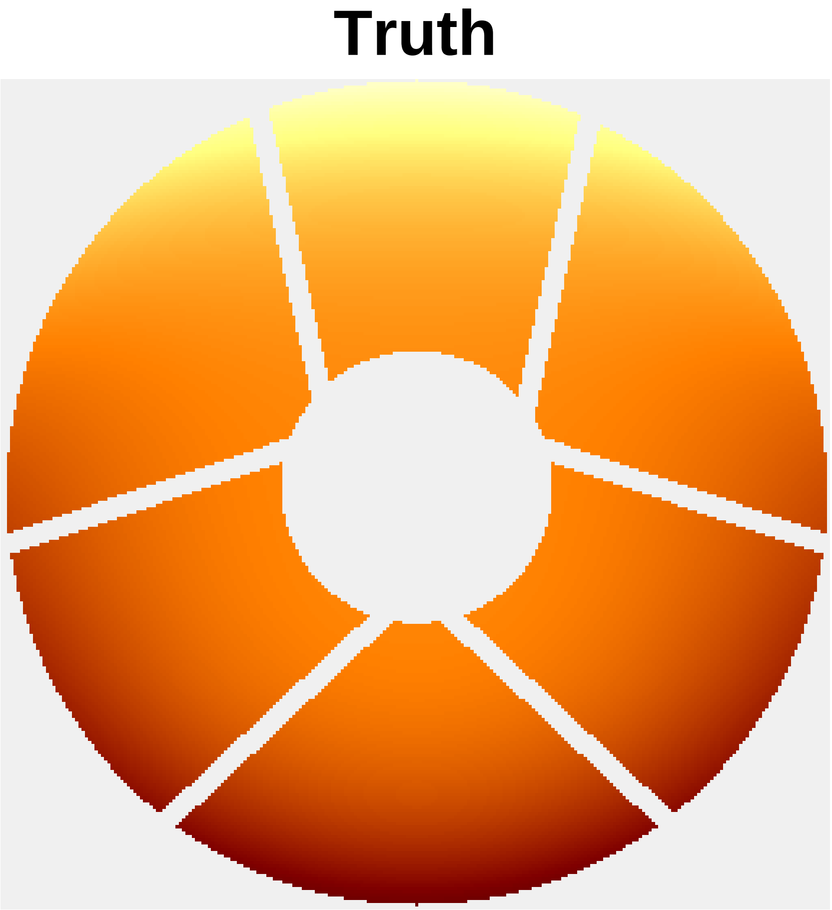


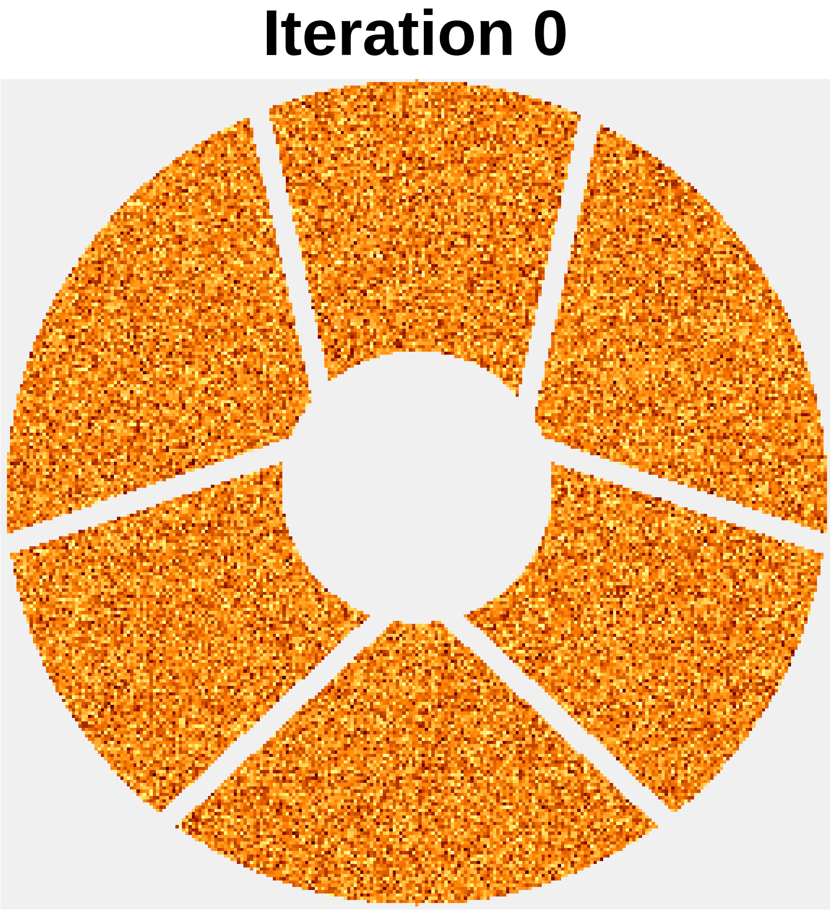
Image Amplitude ($=\sqrt{PSF}$)

Gerchberg-Saxton Algorithm



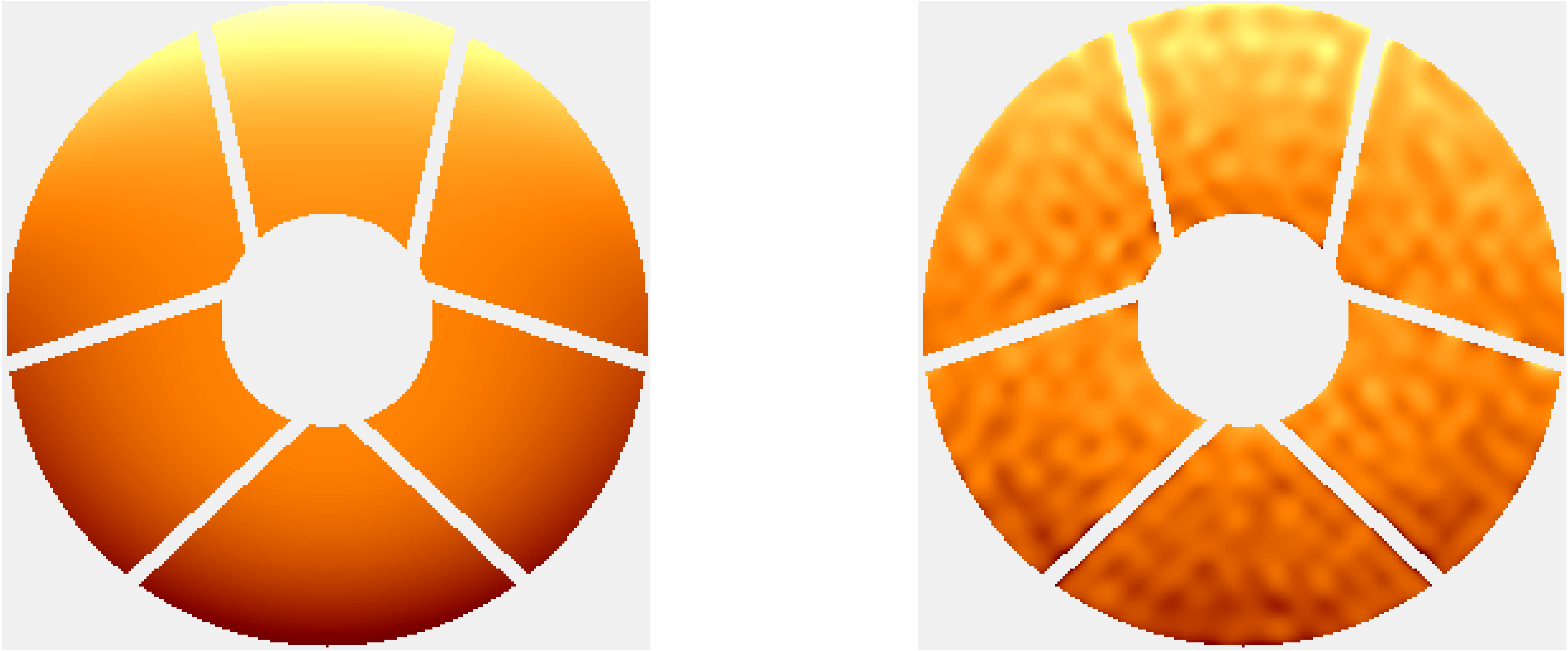
Gerchberg-Saxton Phase



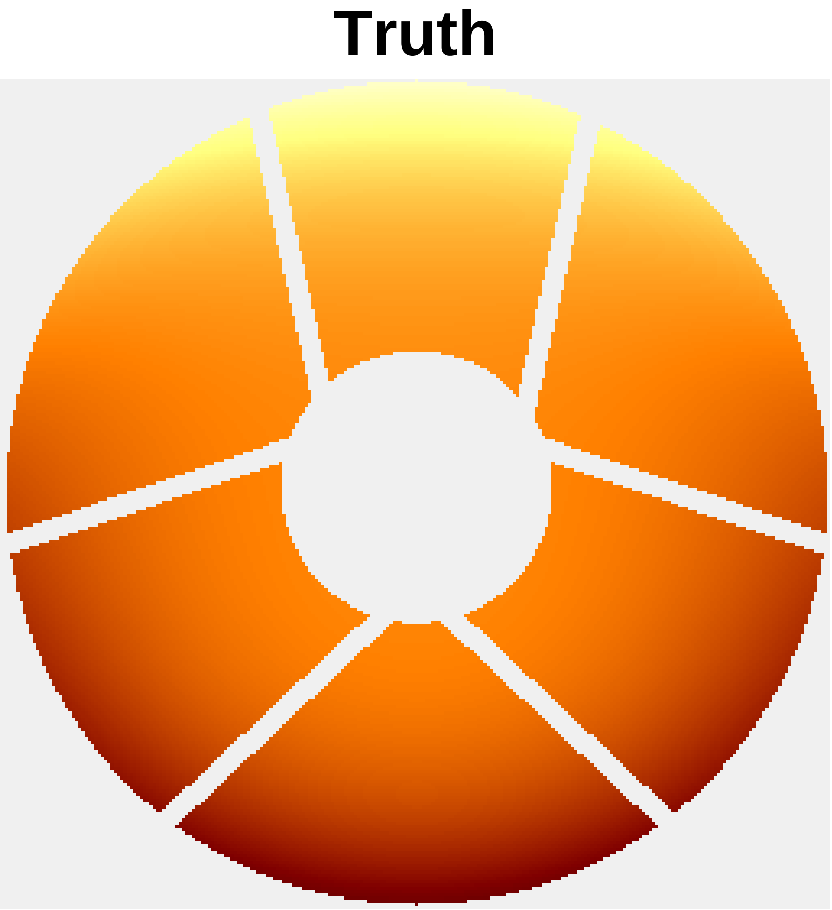


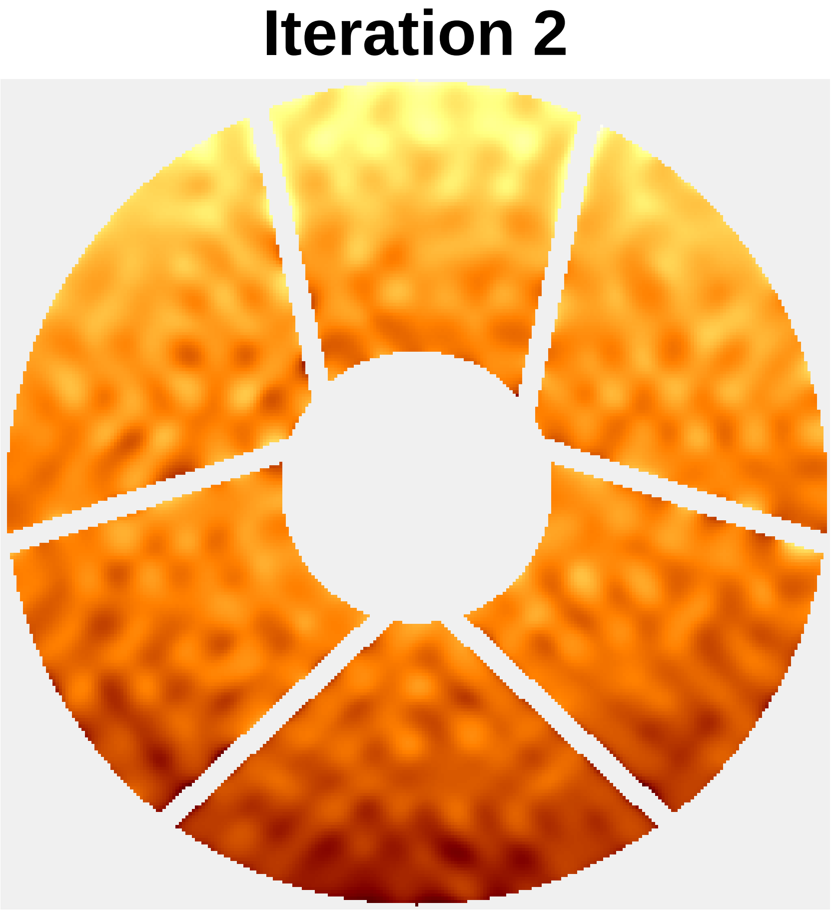
Gerchberg-Saxton Phase

Truth **Iteration 1**

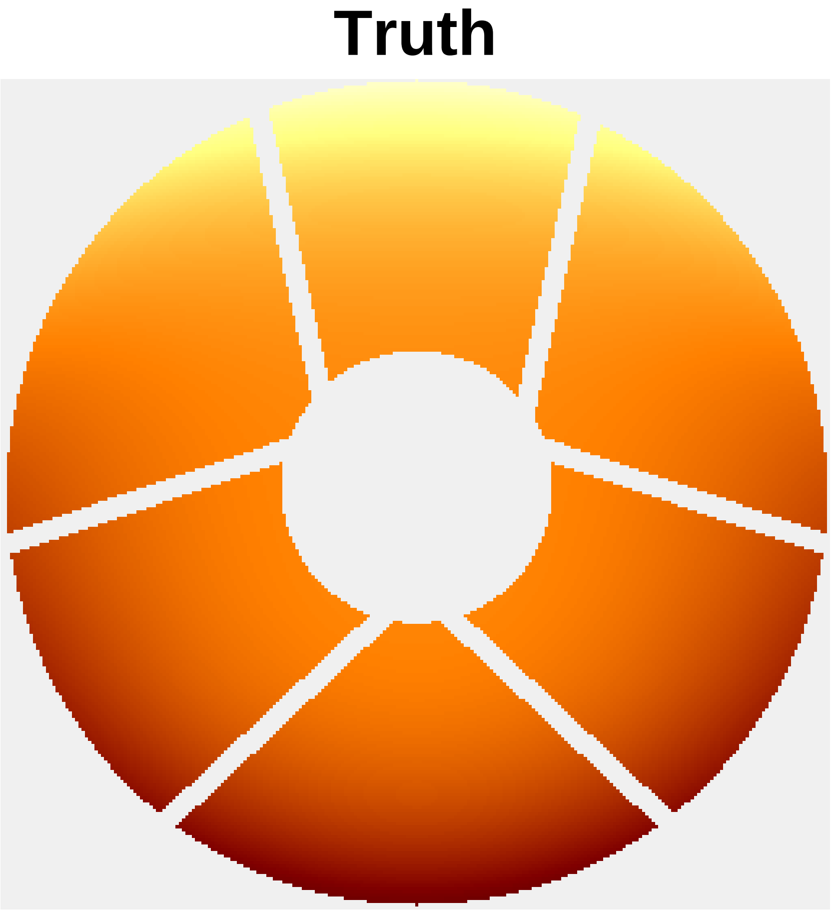


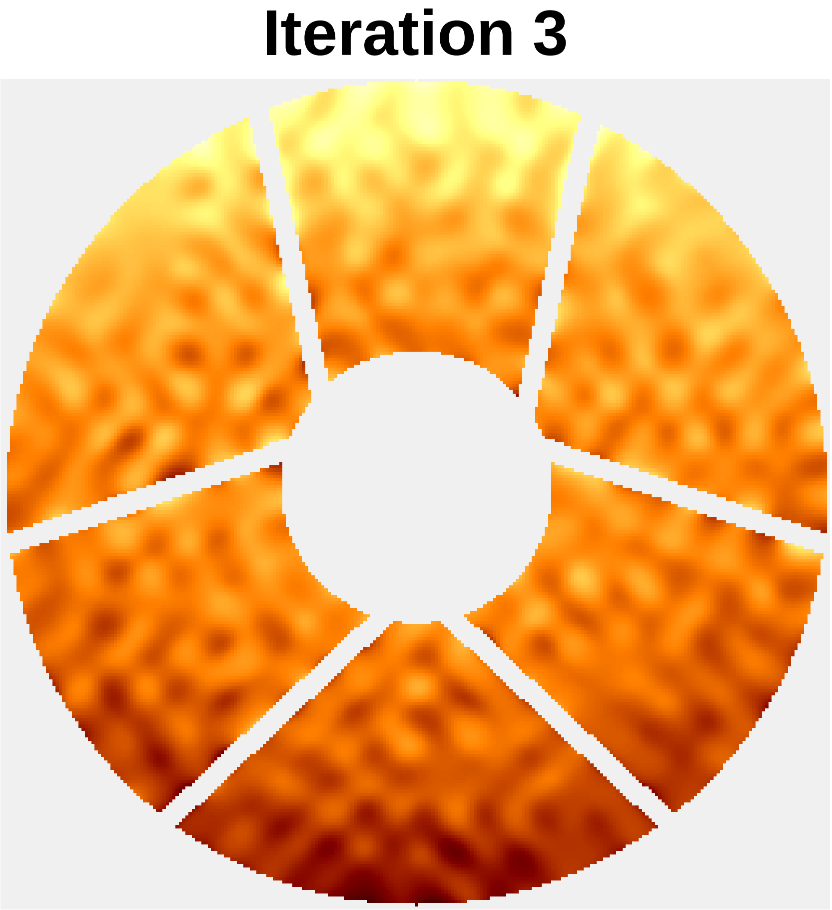
Gerchberg-Saxton Phase





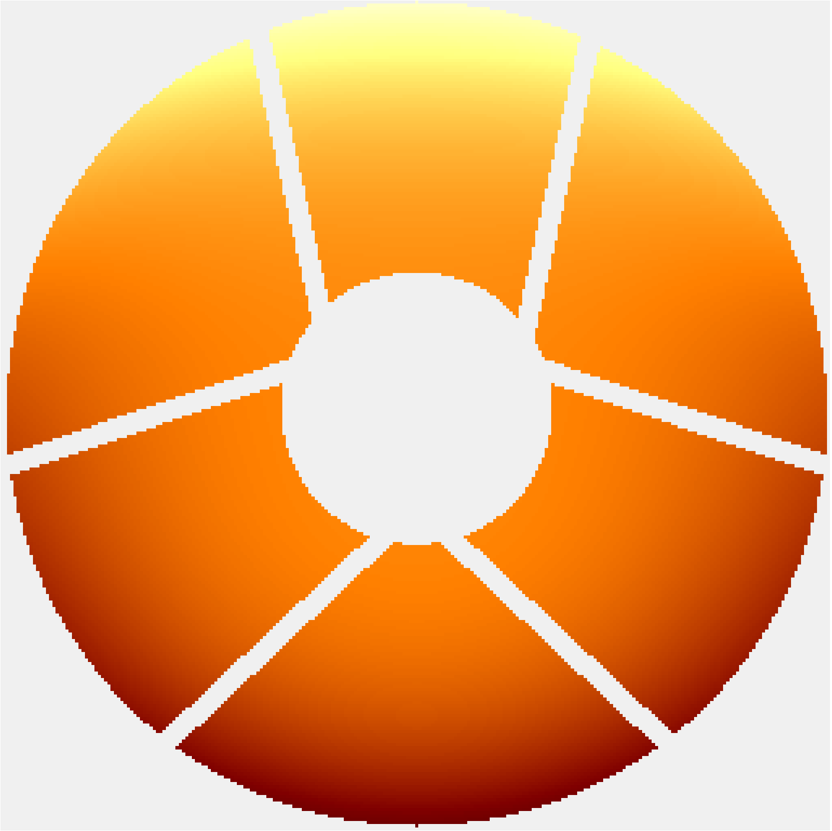
Gerchberg-Saxton Phase



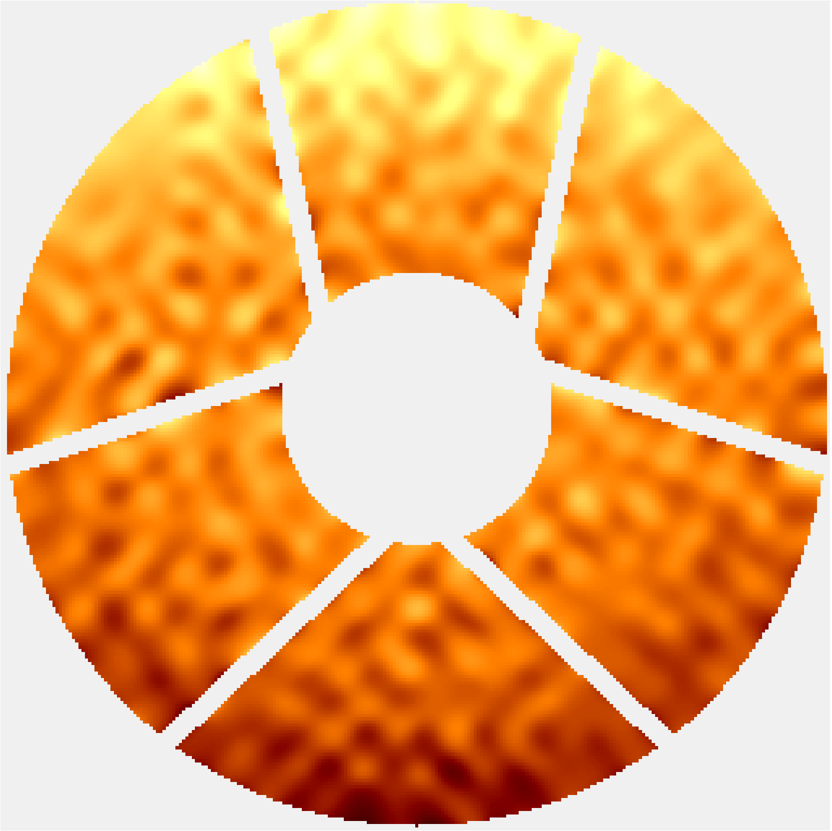


Gerchberg-Saxton Phase

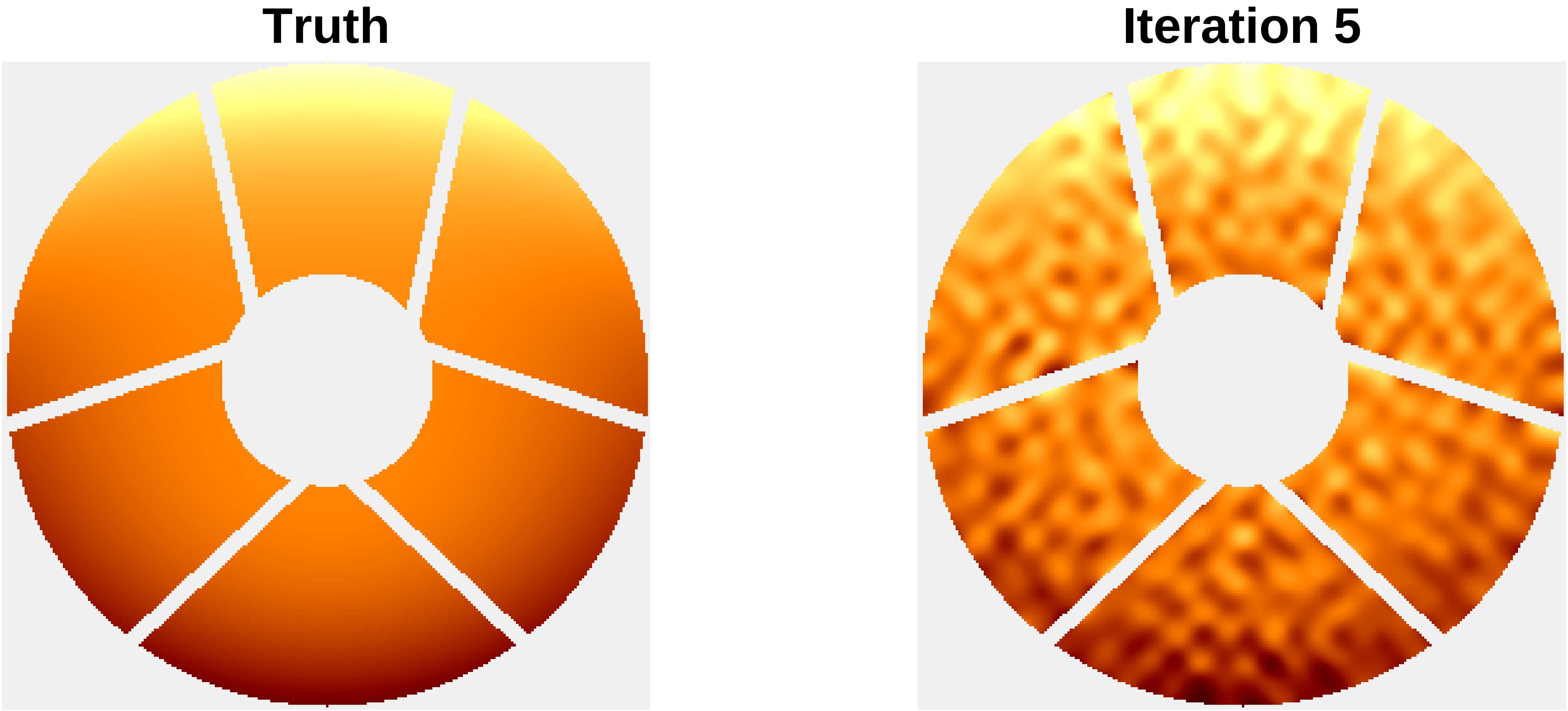
Truth



Iteration 4

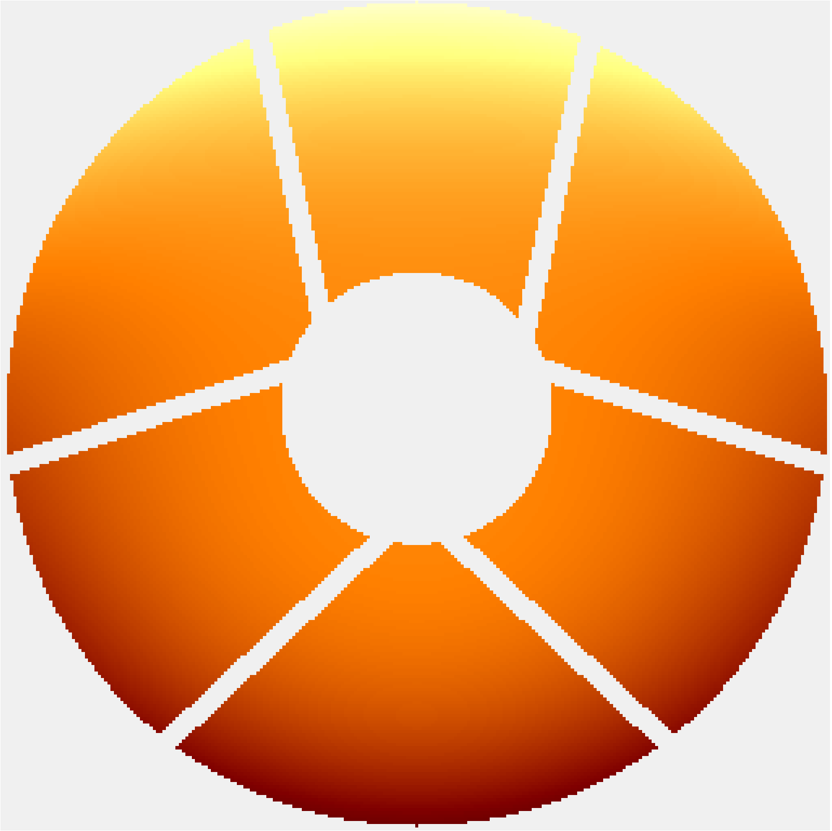


Gerchberg-Saxton Phase

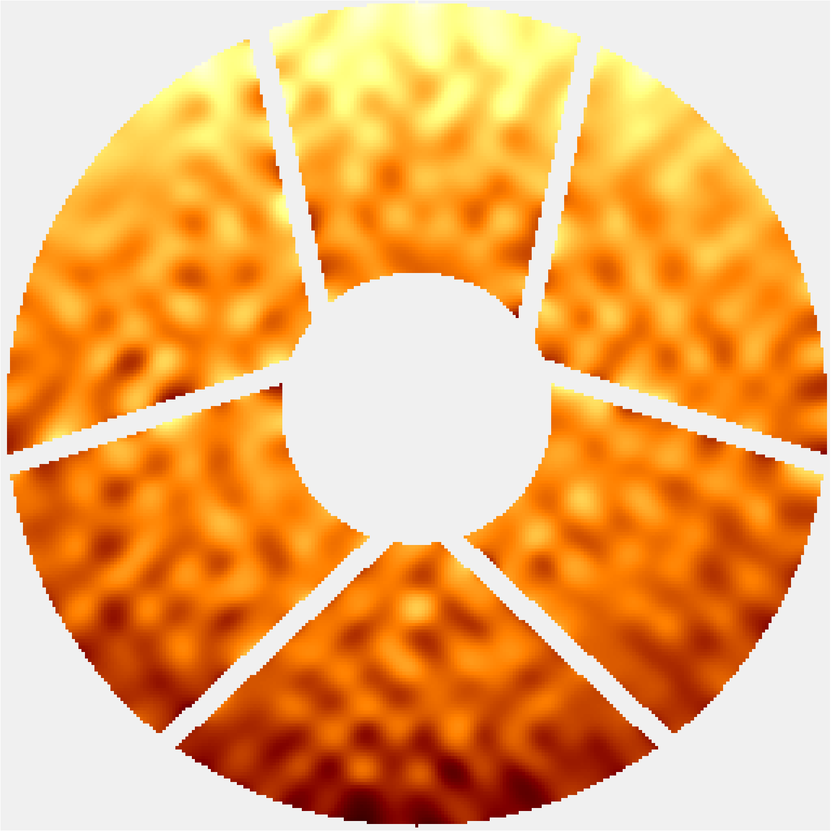


Gerchberg-Saxton Phase

Truth

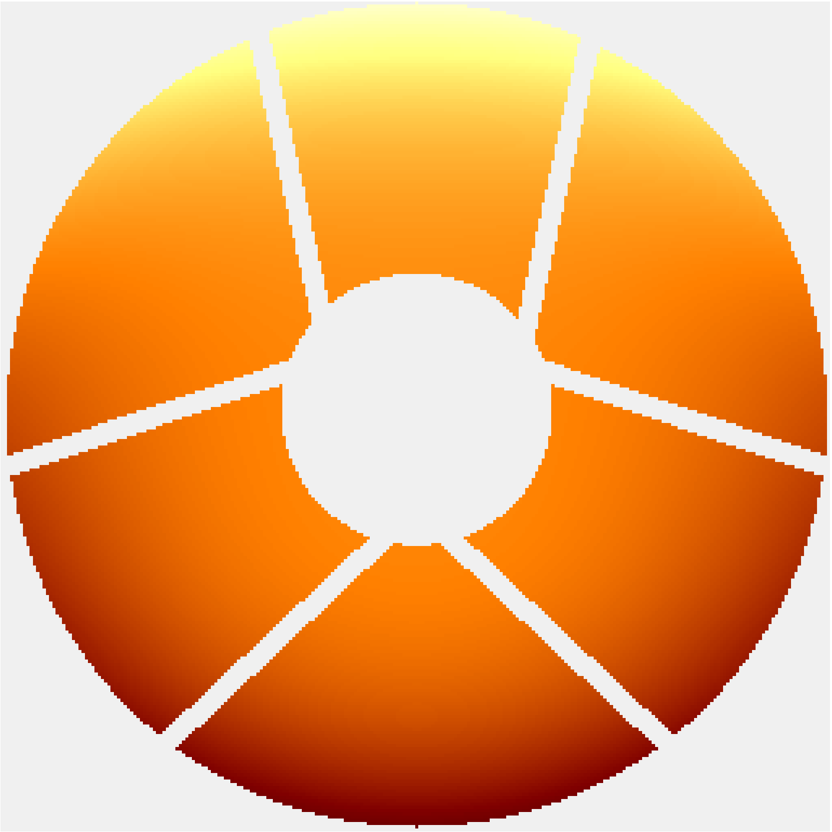


Iteration 6

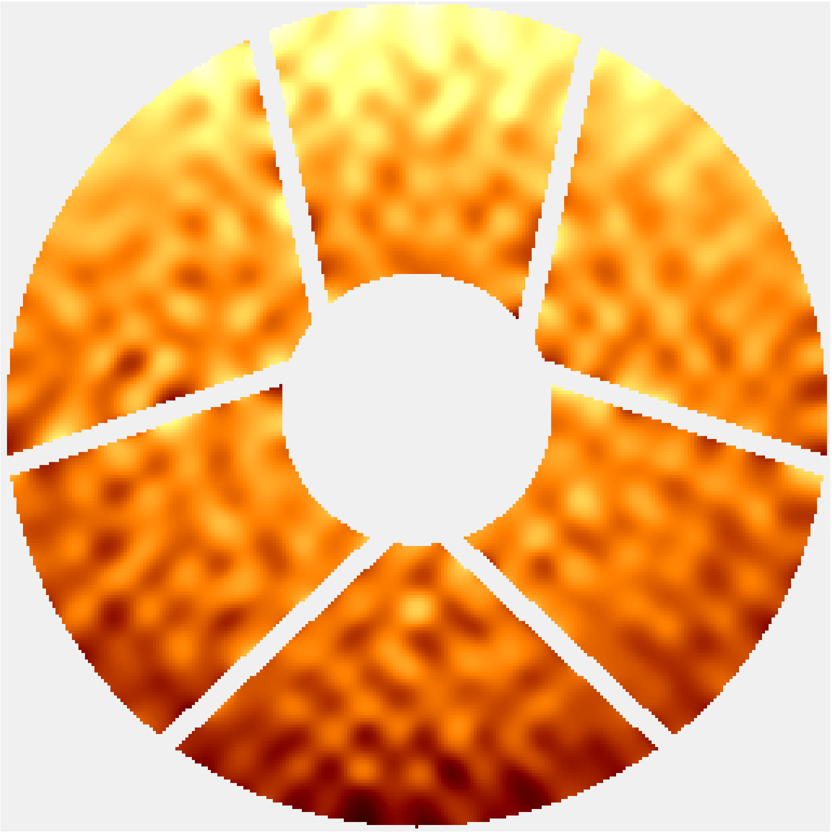


Gerchberg-Saxton Phase

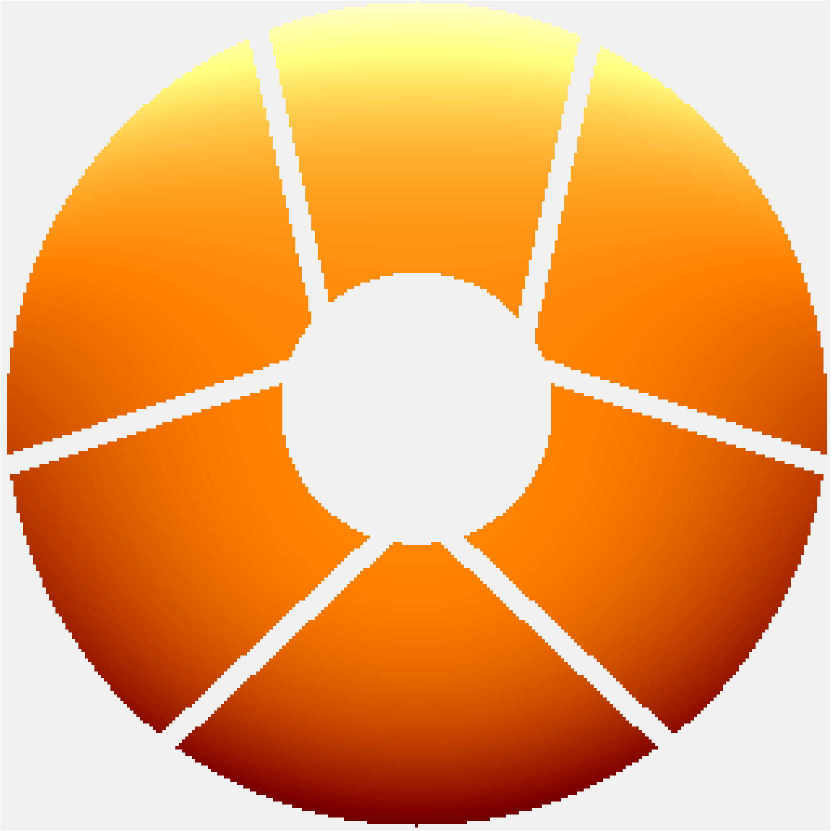
Truth

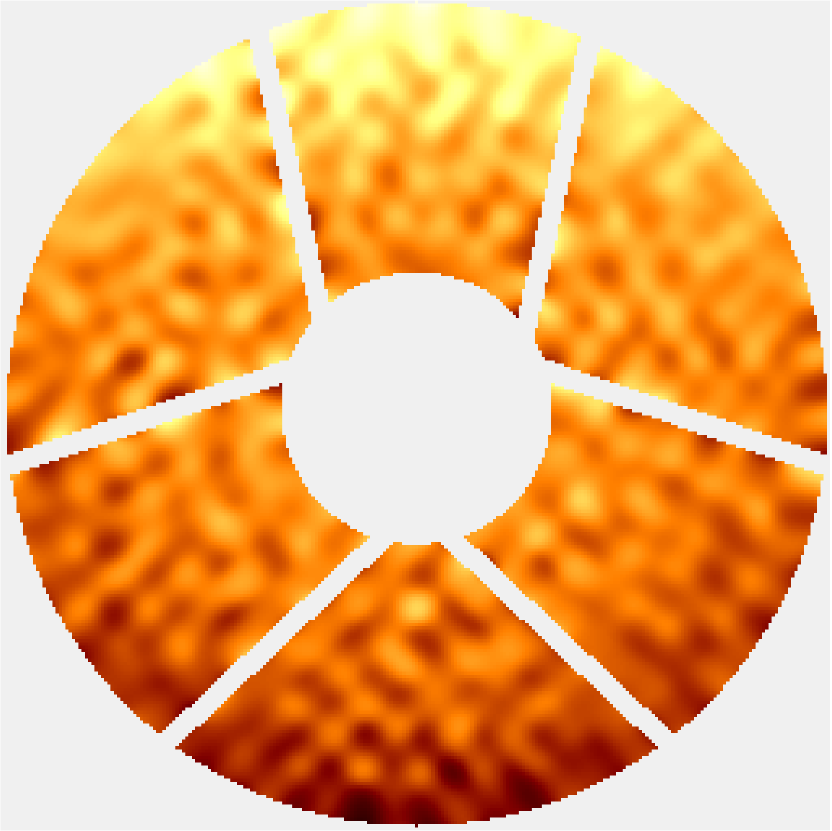


Iteration 7



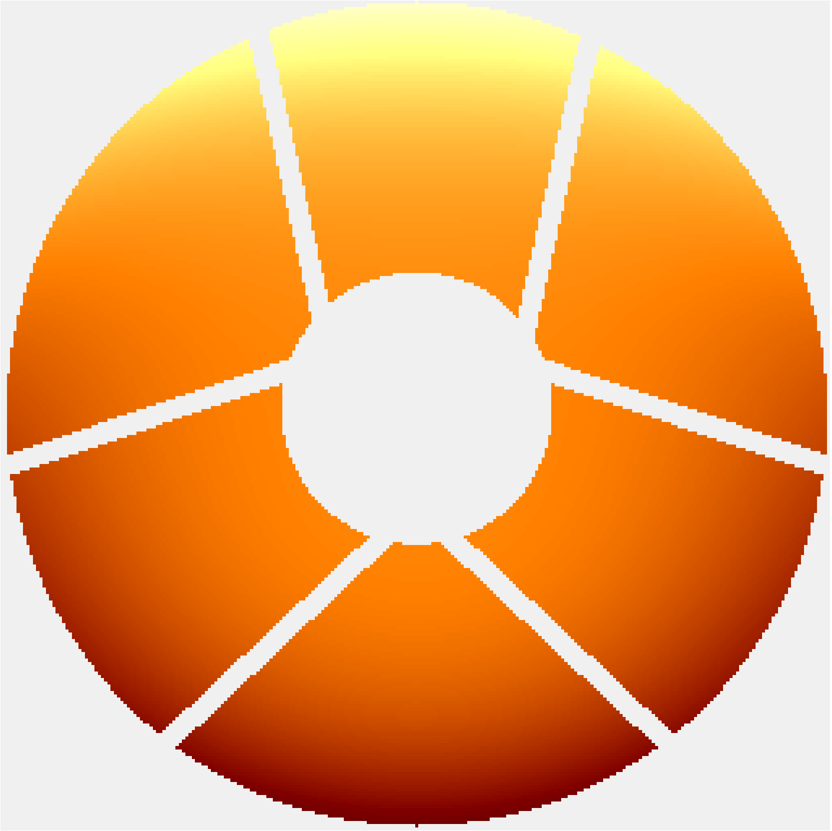
Gerchberg-Saxton Phase



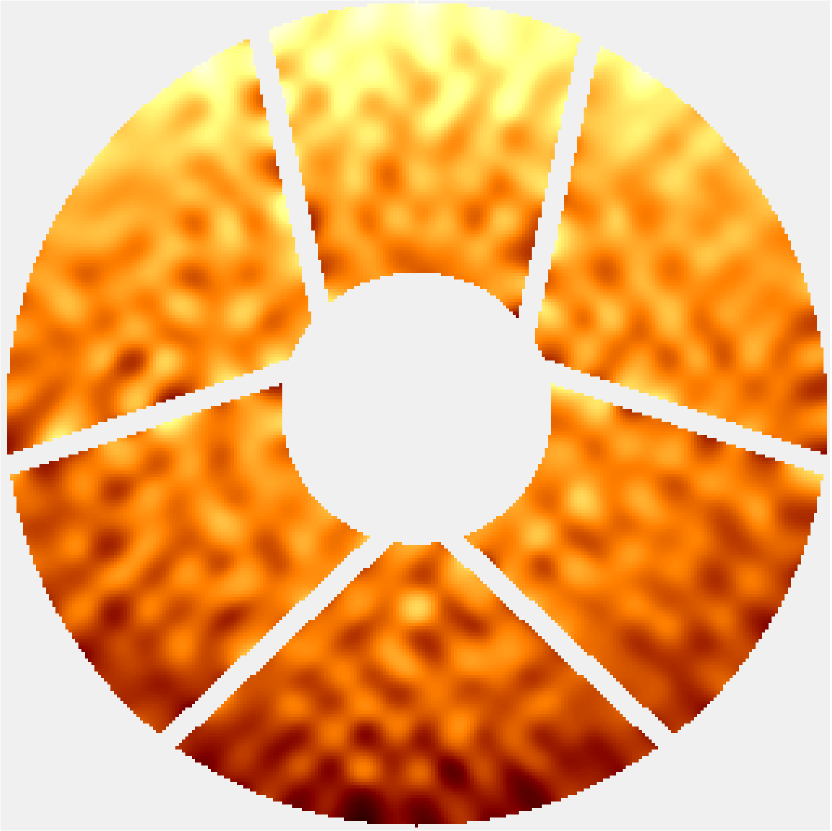


Gerchberg-Saxton Phase

Truth

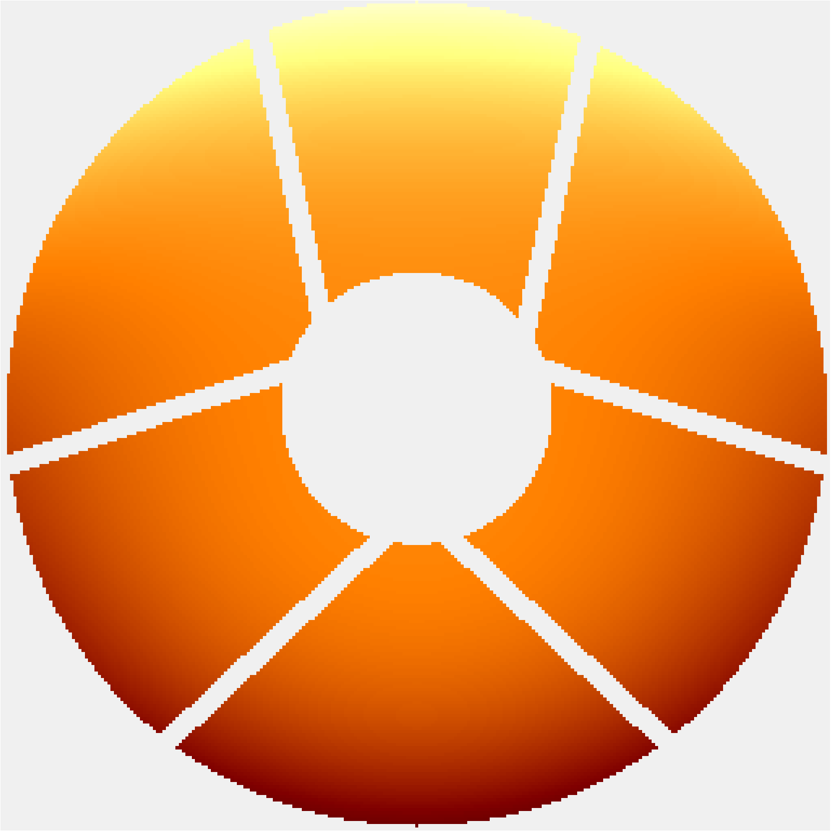


Iteration 9

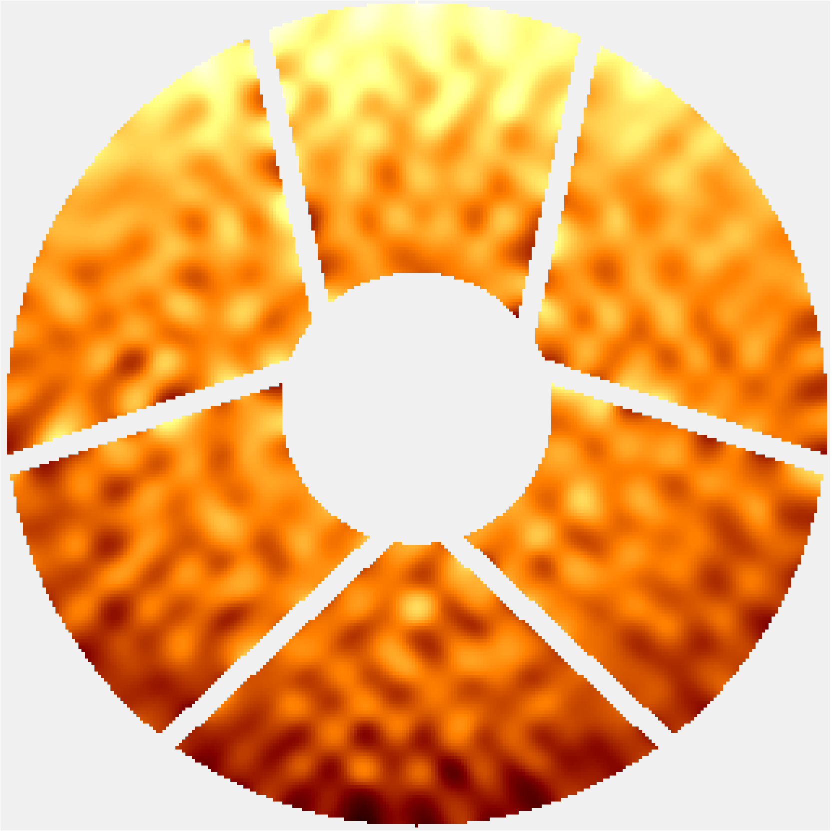


Gerchberg-Saxton Phase

Truth

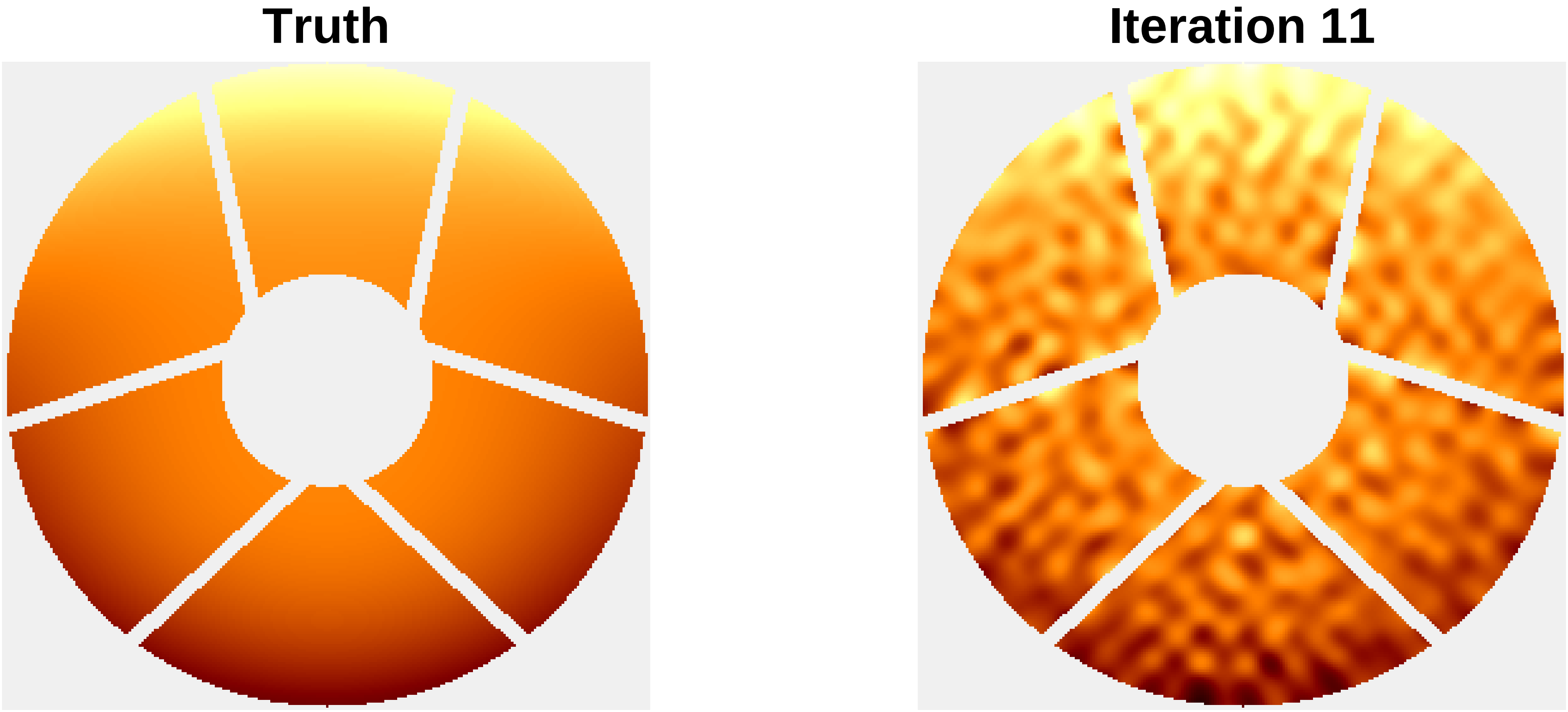


Iteration 10

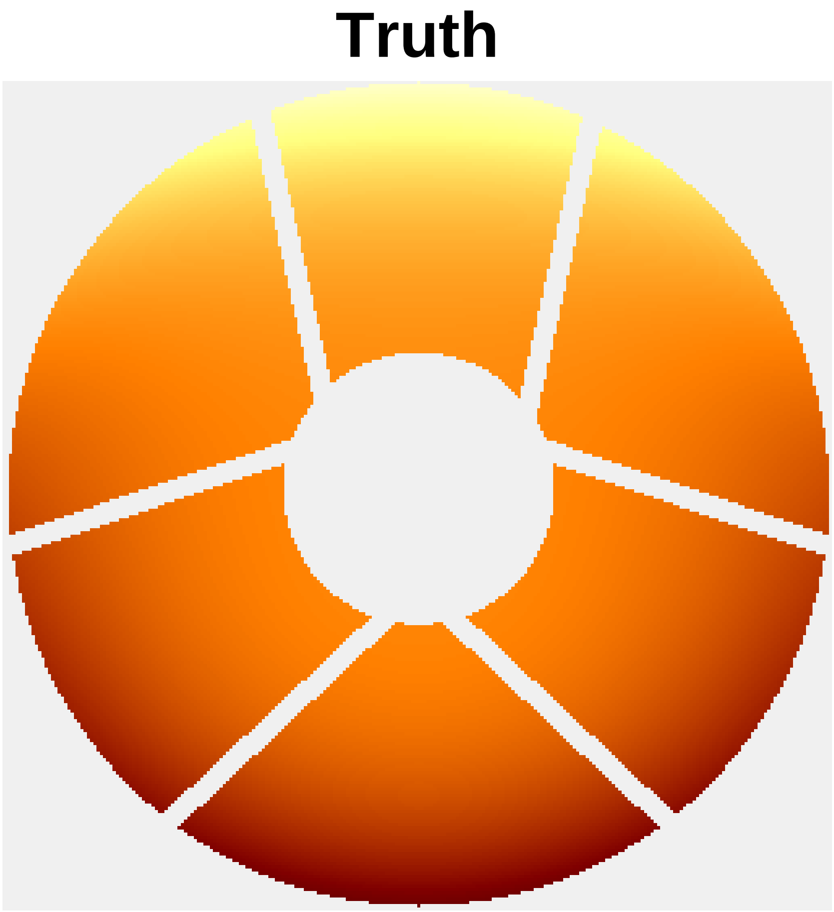


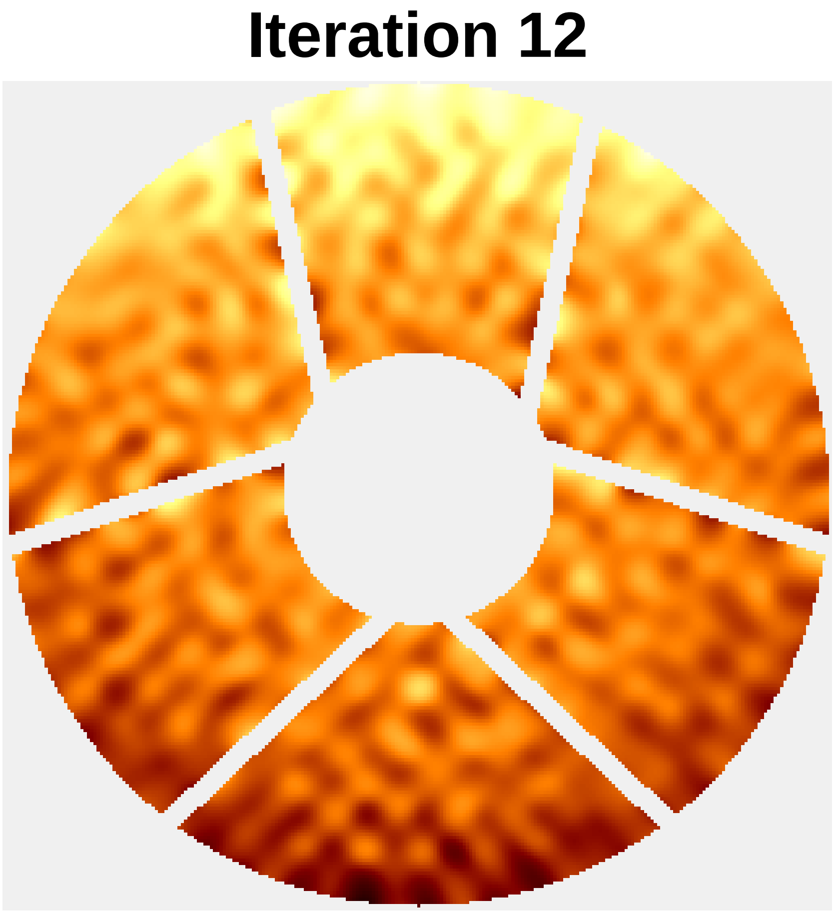
The Institute of OPTICS

Gerchberg-Saxton Phase



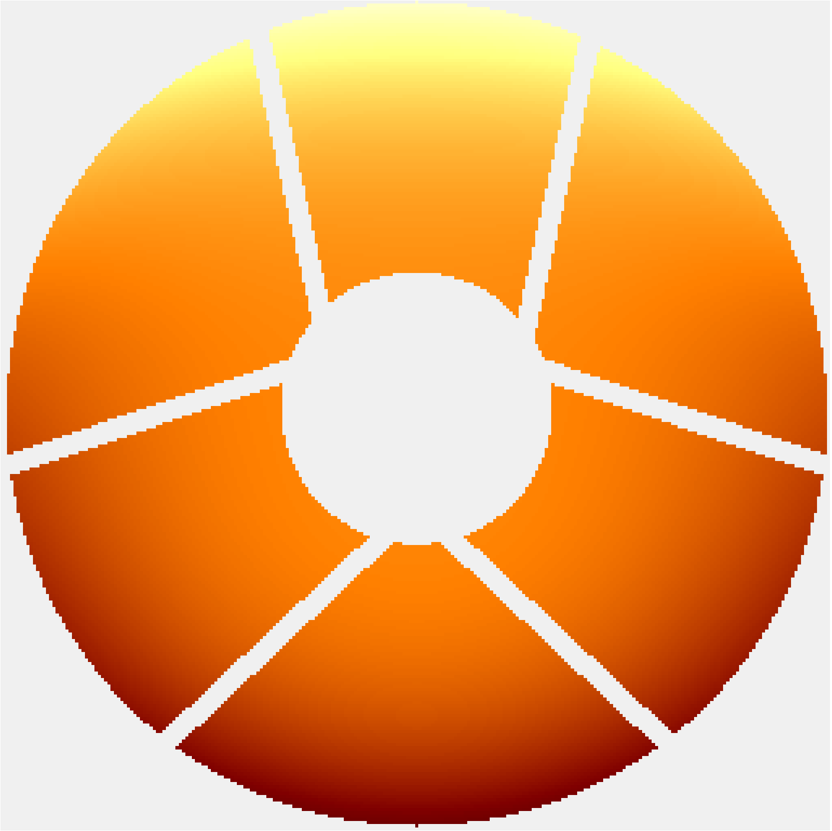
Gerchberg-Saxton Phase



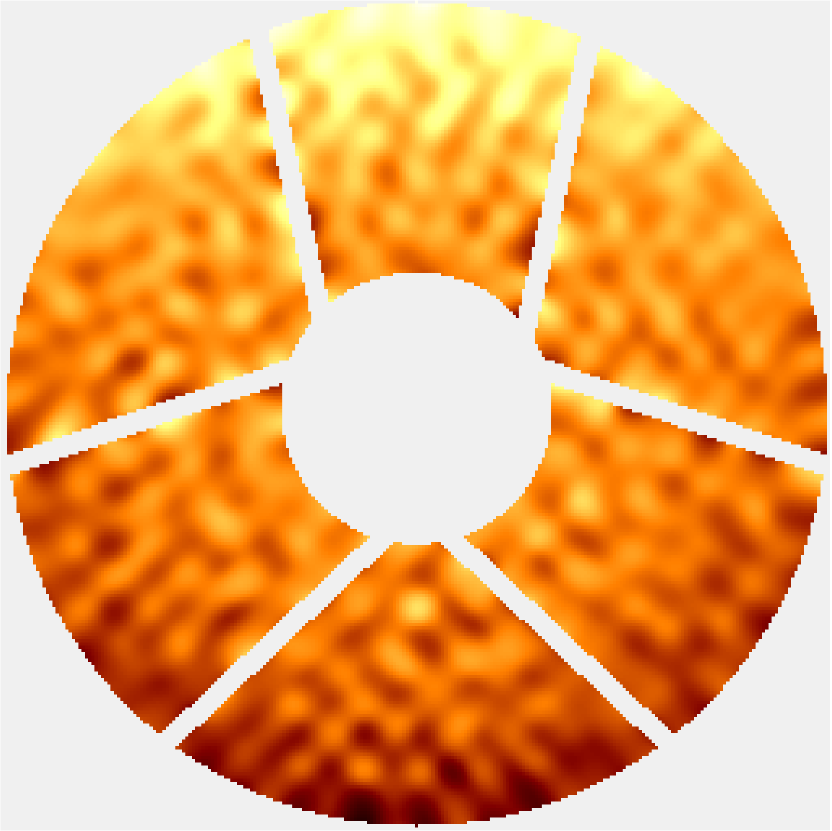


Gerchberg-Saxton Phase

Truth

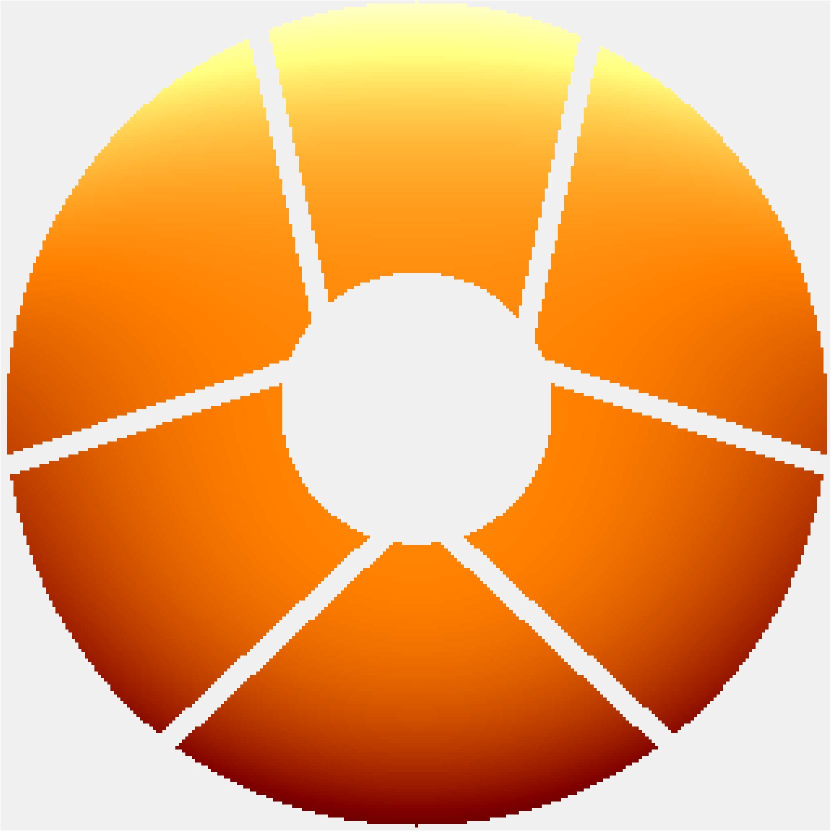


Iteration 13

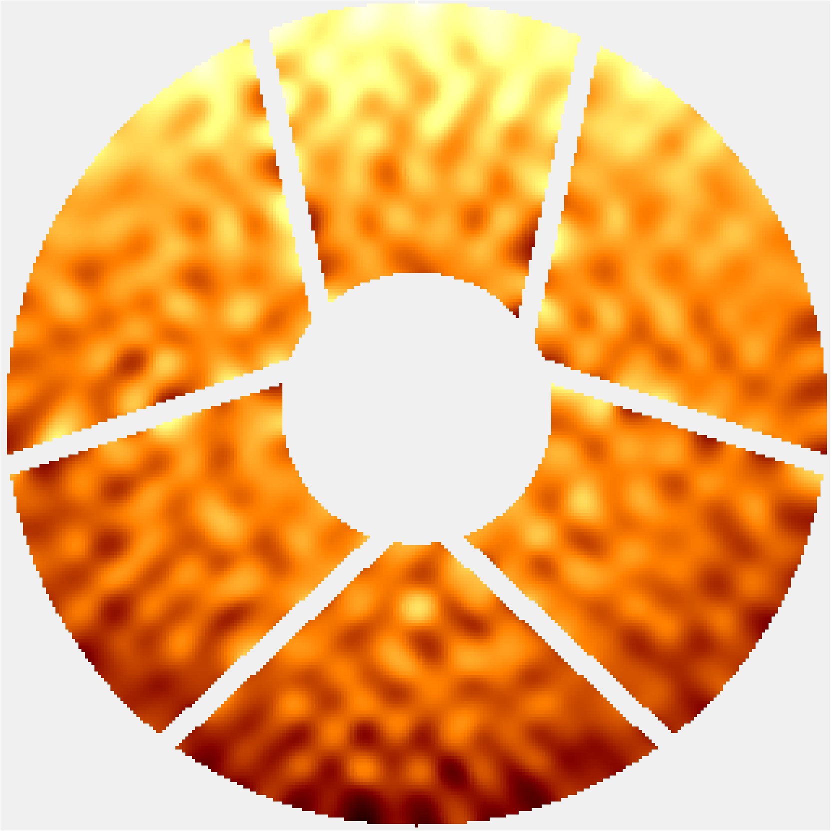


Gerchberg-Saxton Phase

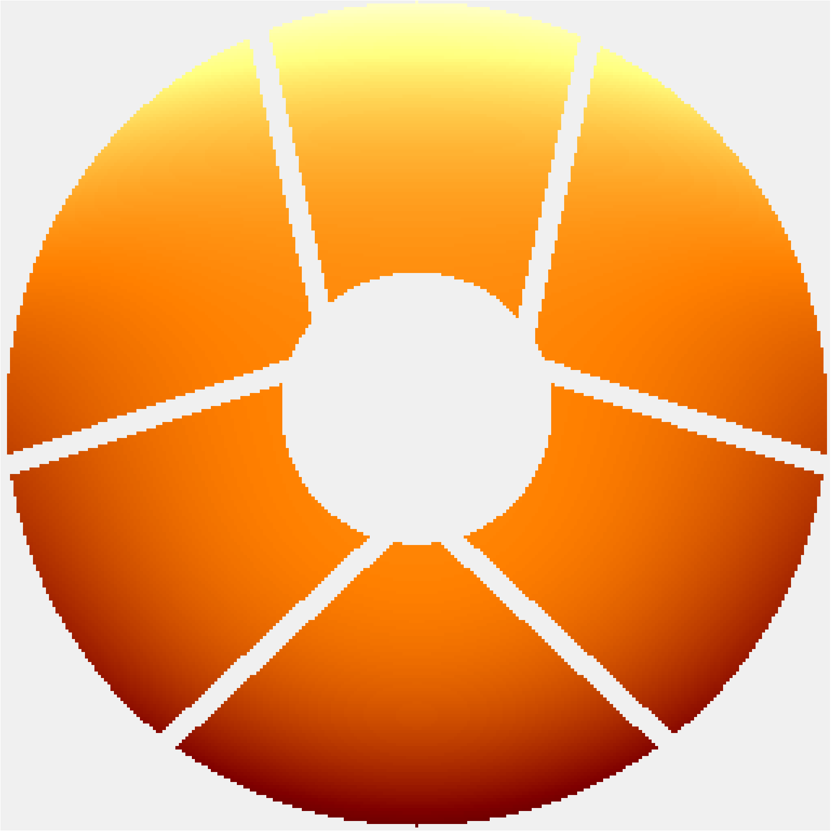
Truth

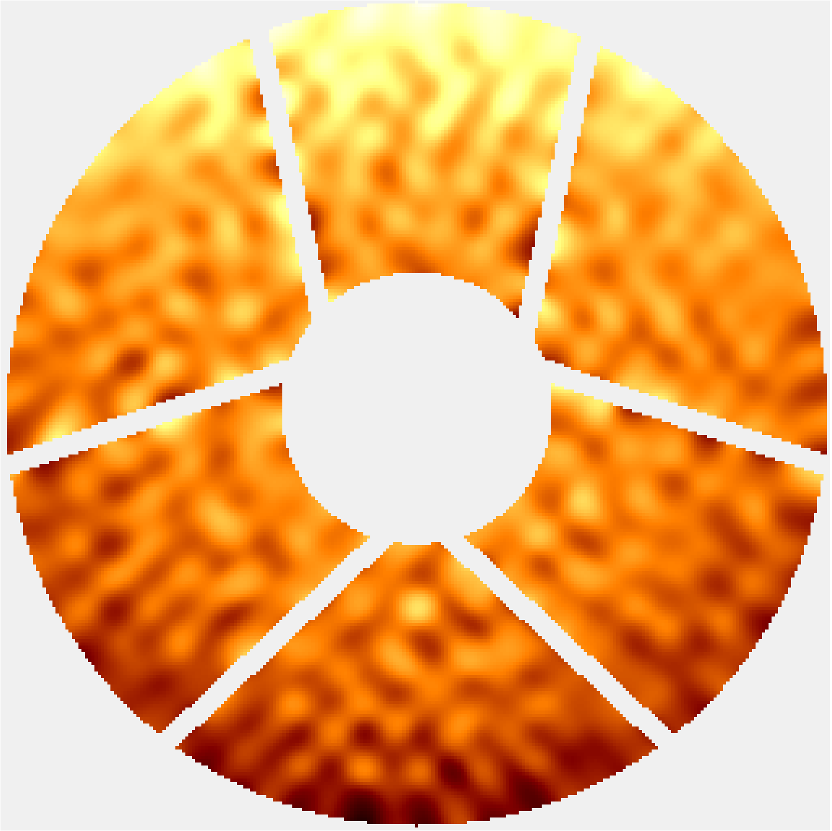


Iteration 14

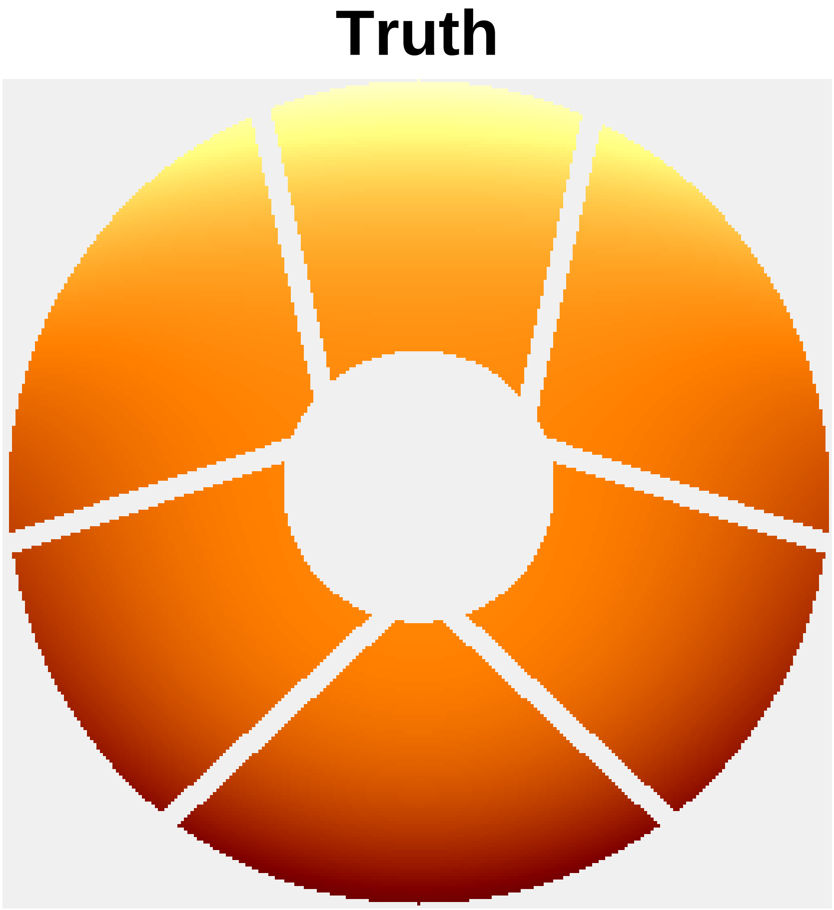


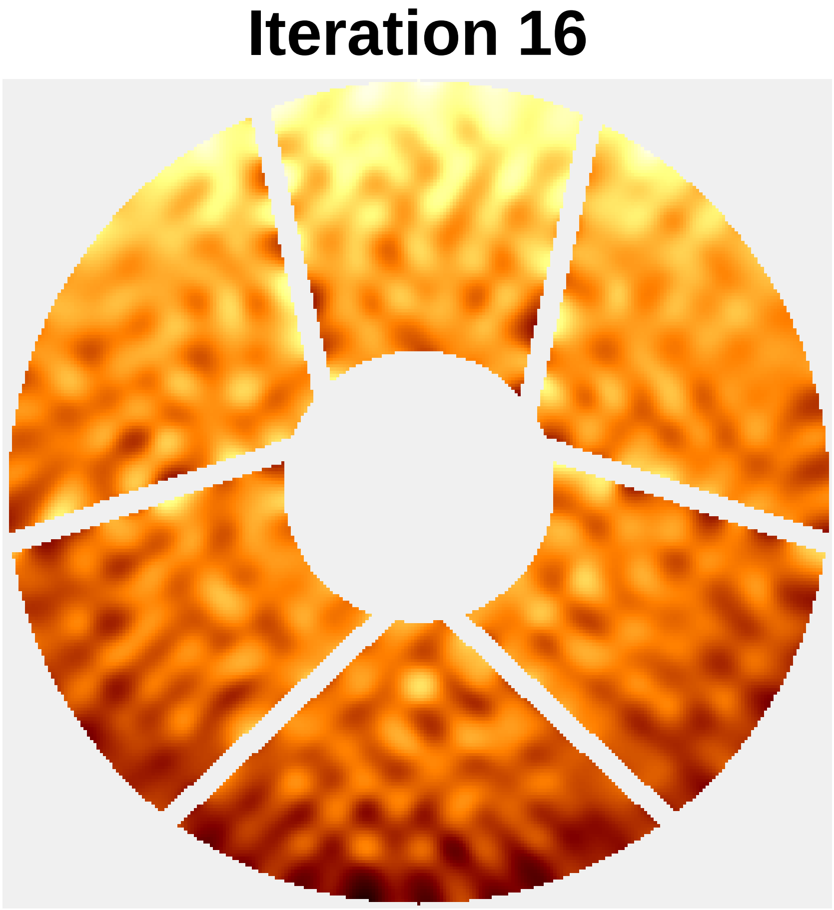
Gerchberg-Saxton Phase


Truth


Iteration 15

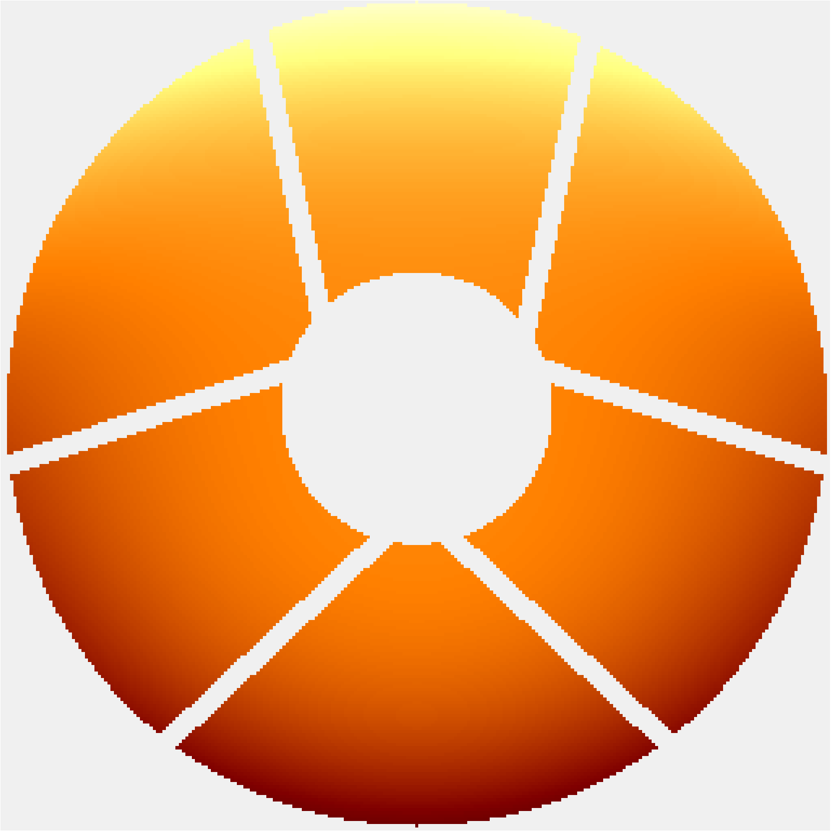
Gerchberg-Saxton Phase



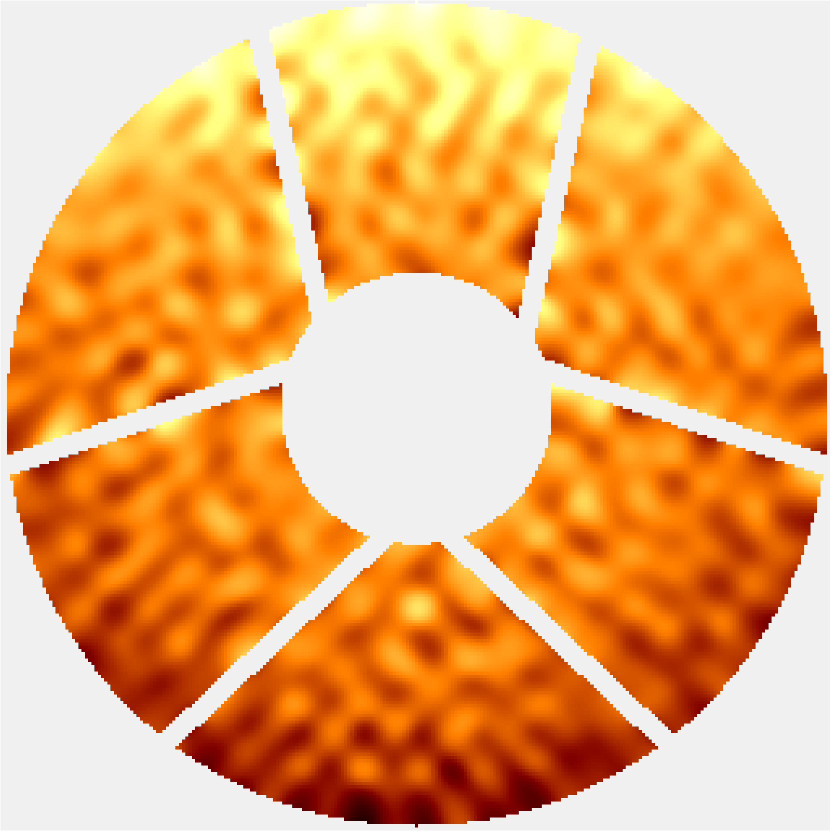


Gerchberg-Saxton Phase

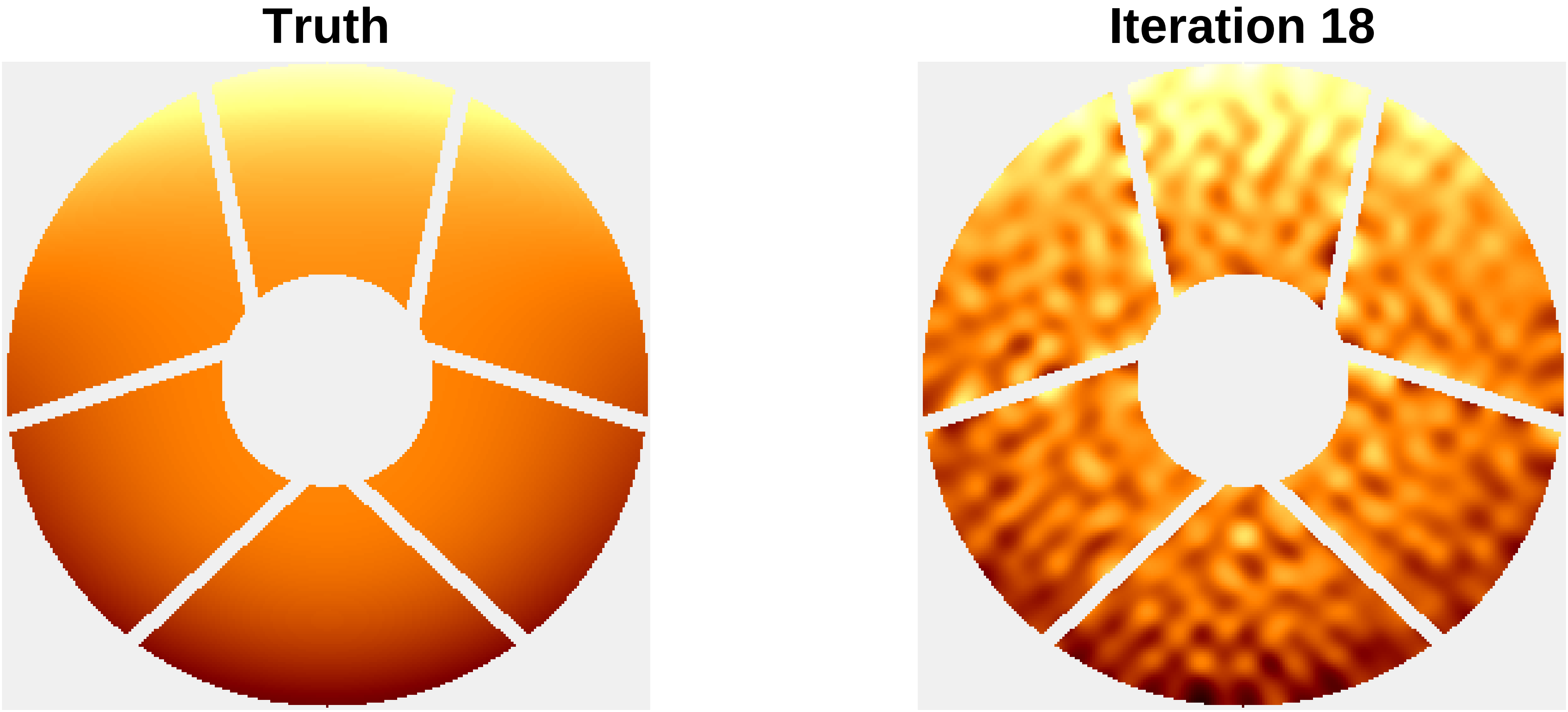
Truth



Iteration 17

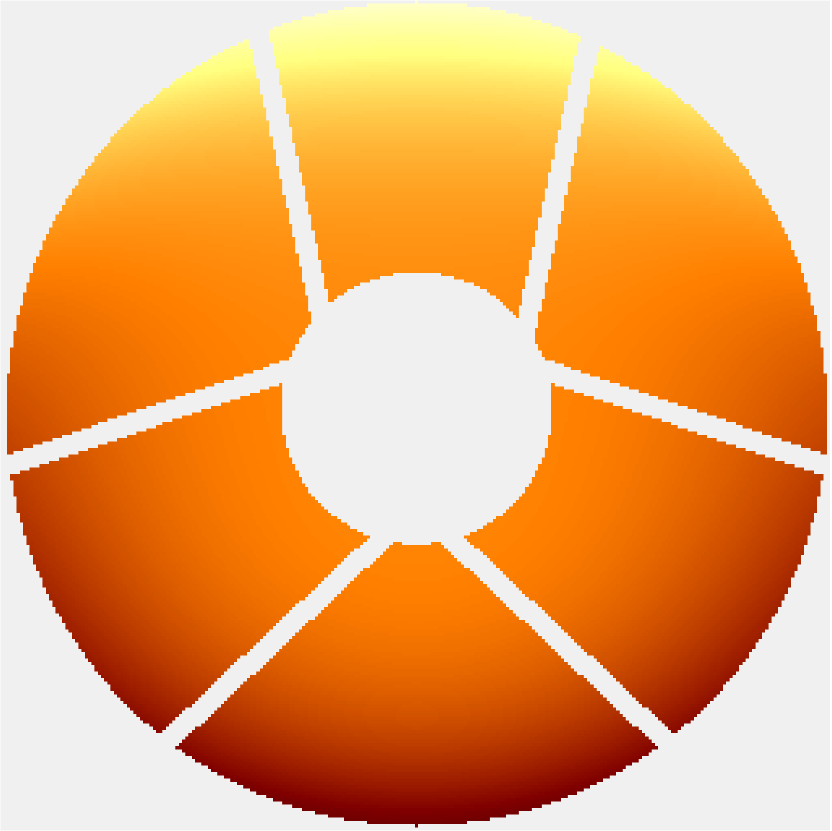


Gerchberg-Saxton Phase

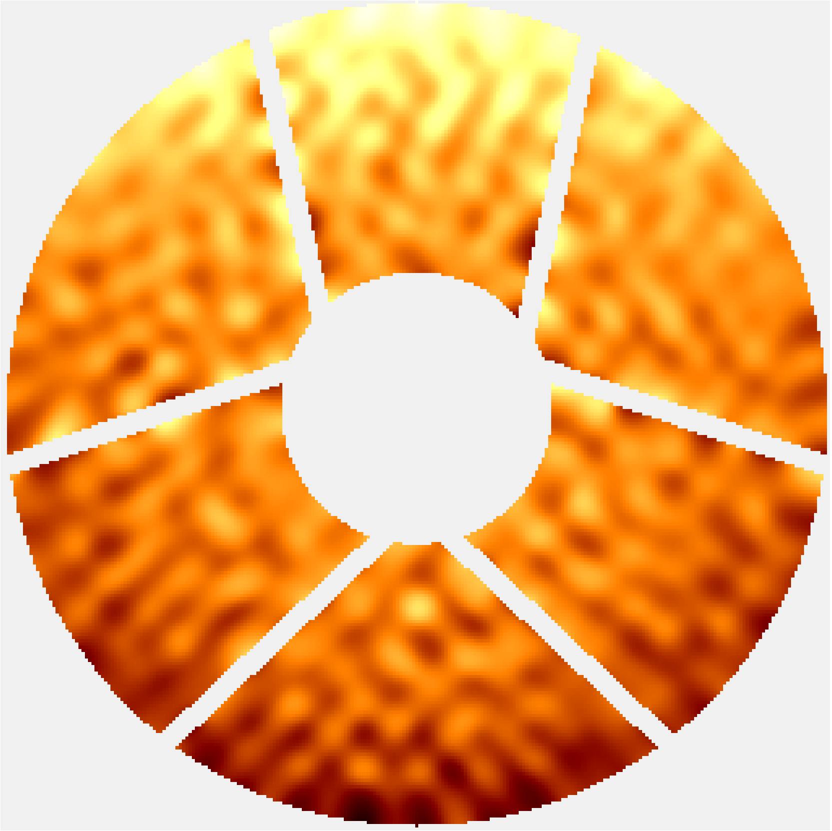


Gerchberg-Saxton Phase

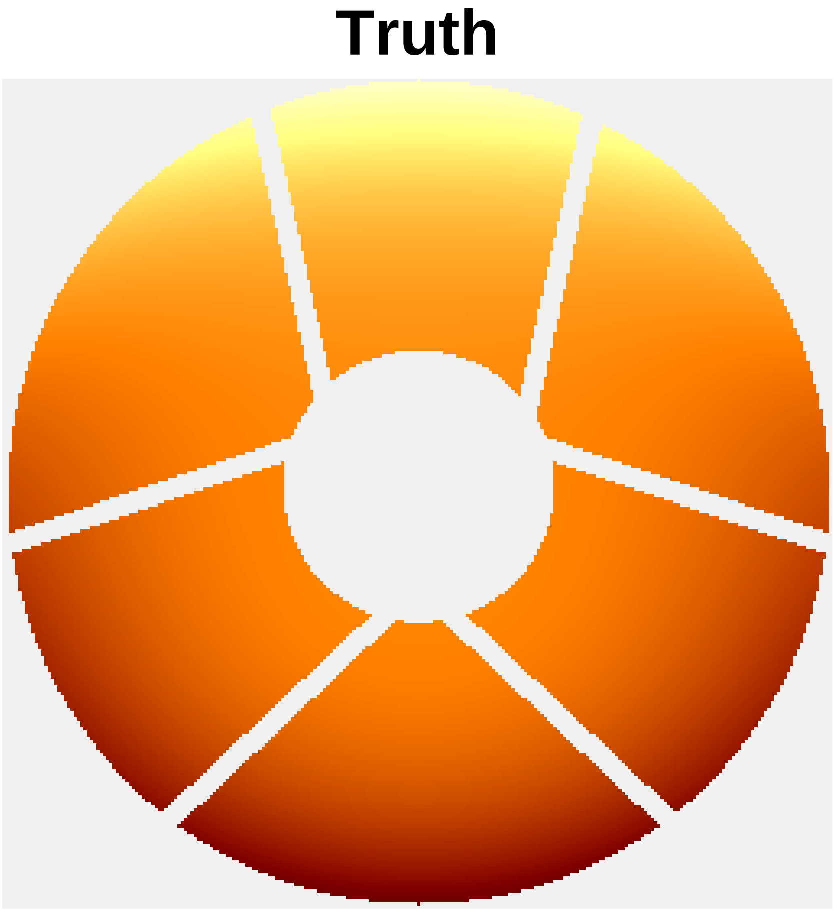
Truth

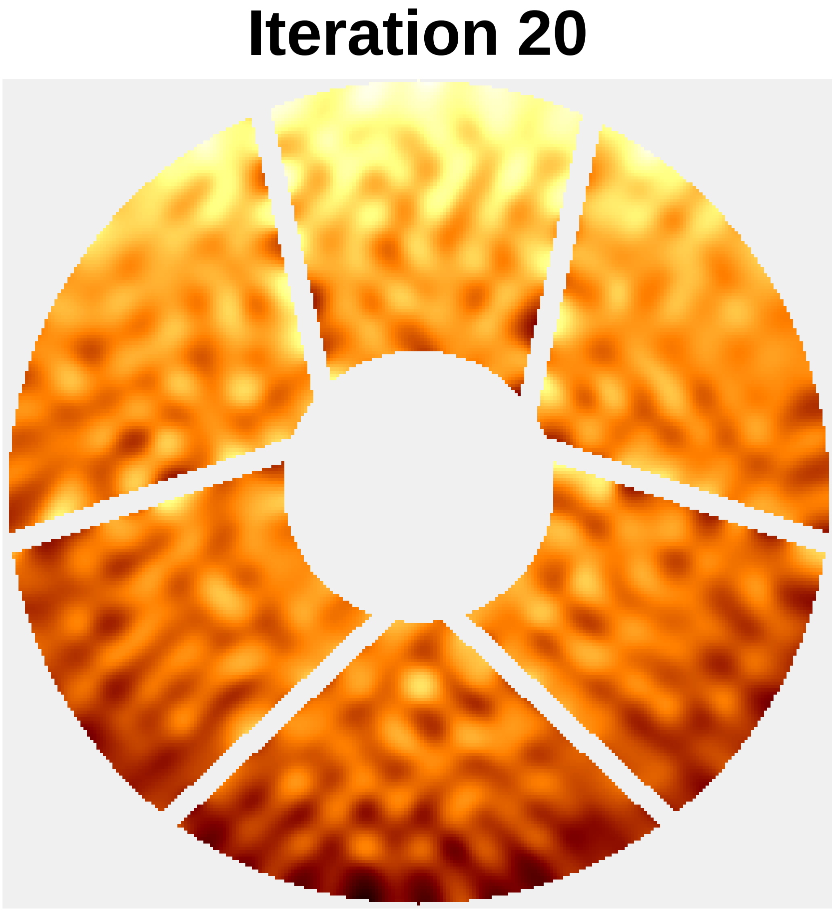


Iteration 19



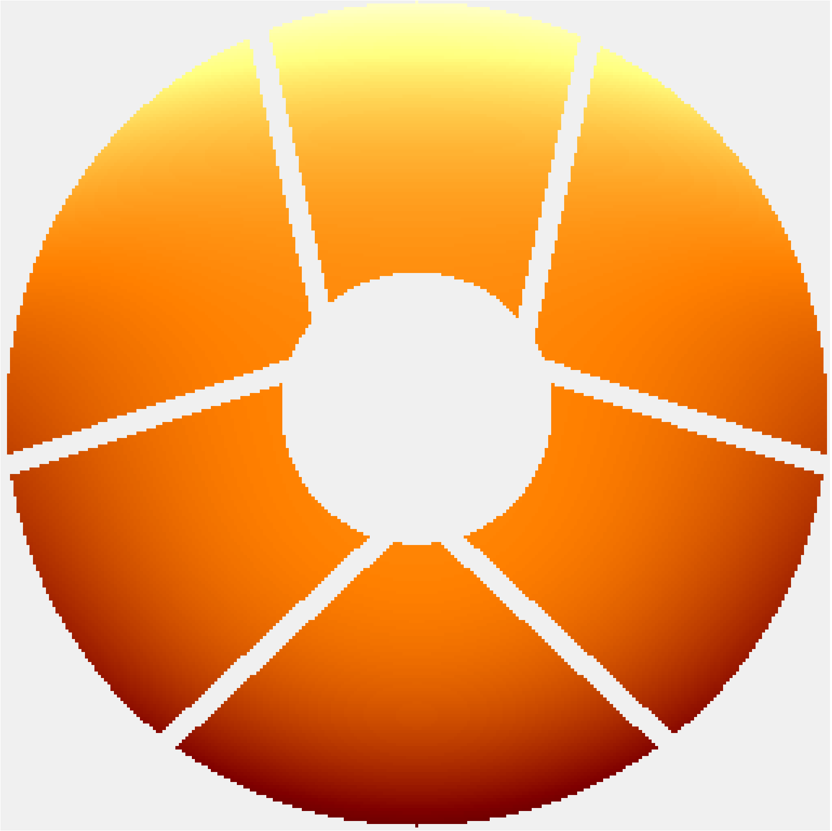
Gerchberg-Saxton Phase



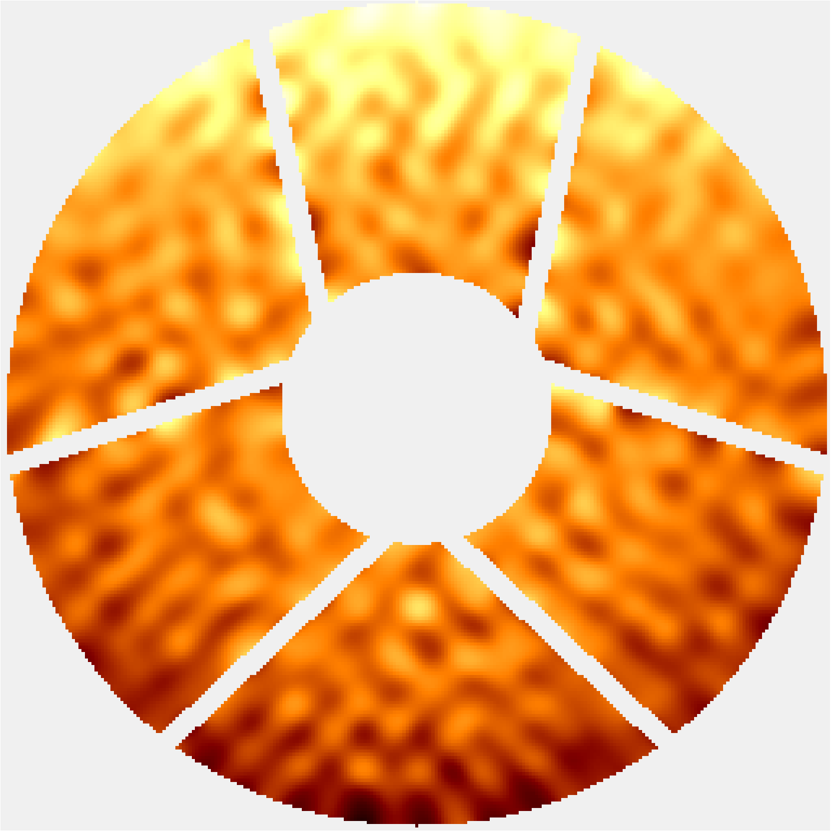


Gerchberg-Saxton Phase

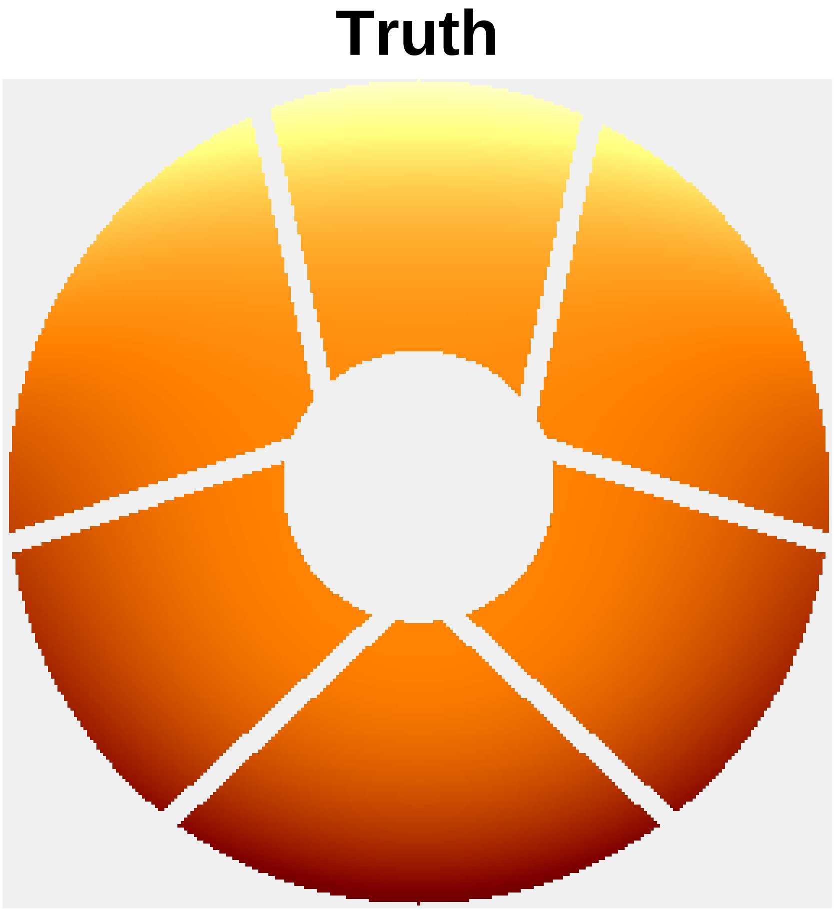
Truth

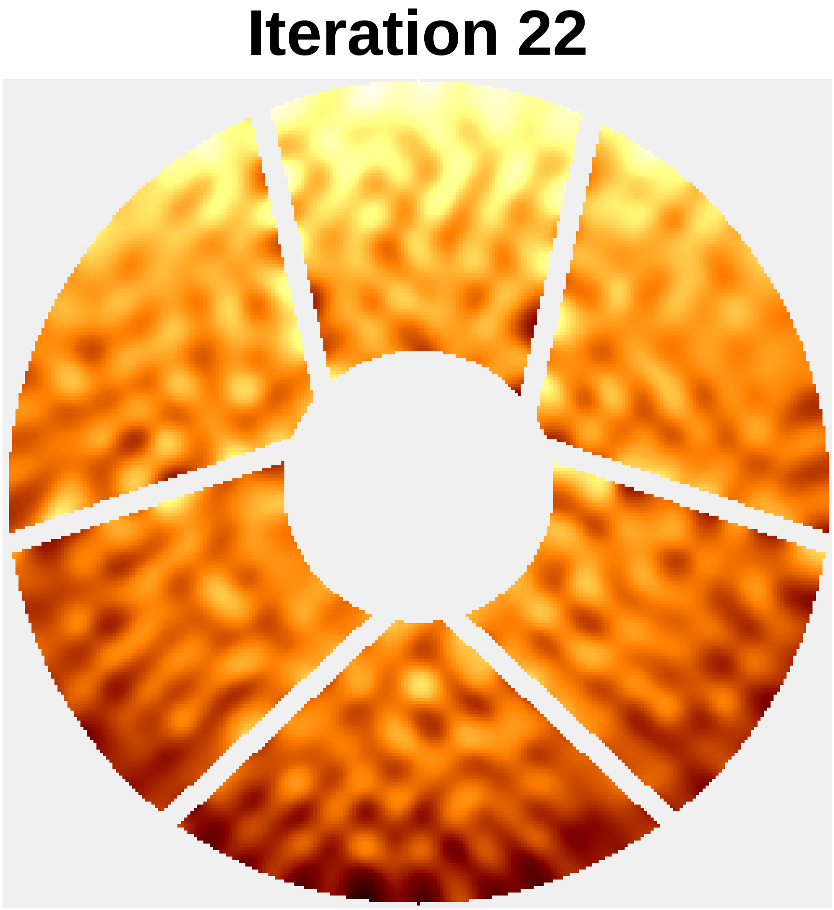


Iteration 21



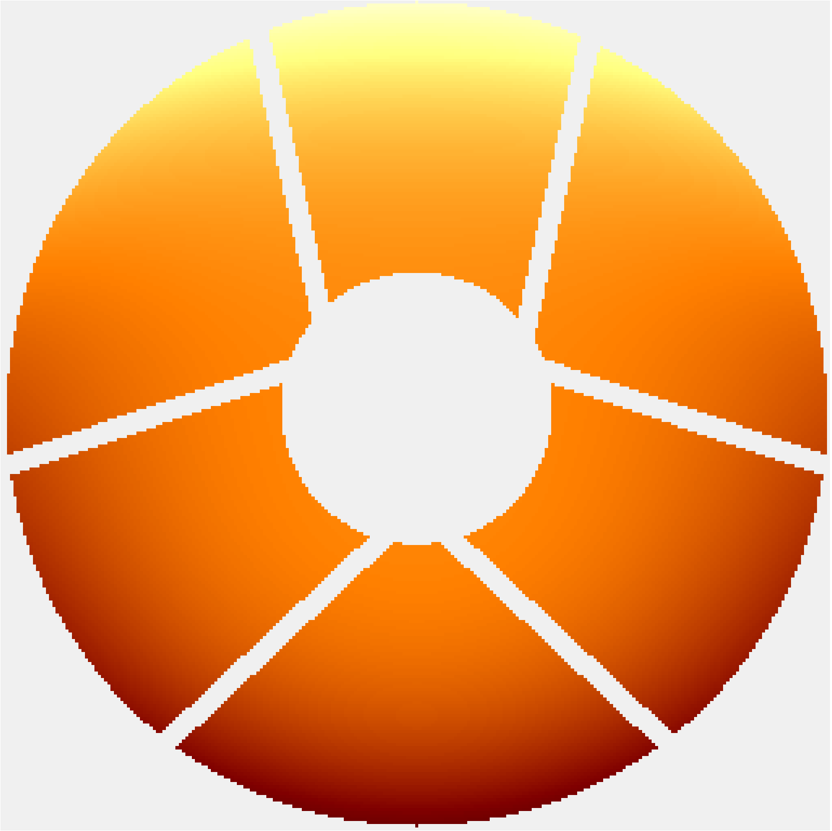
Gerchberg-Saxton Phase



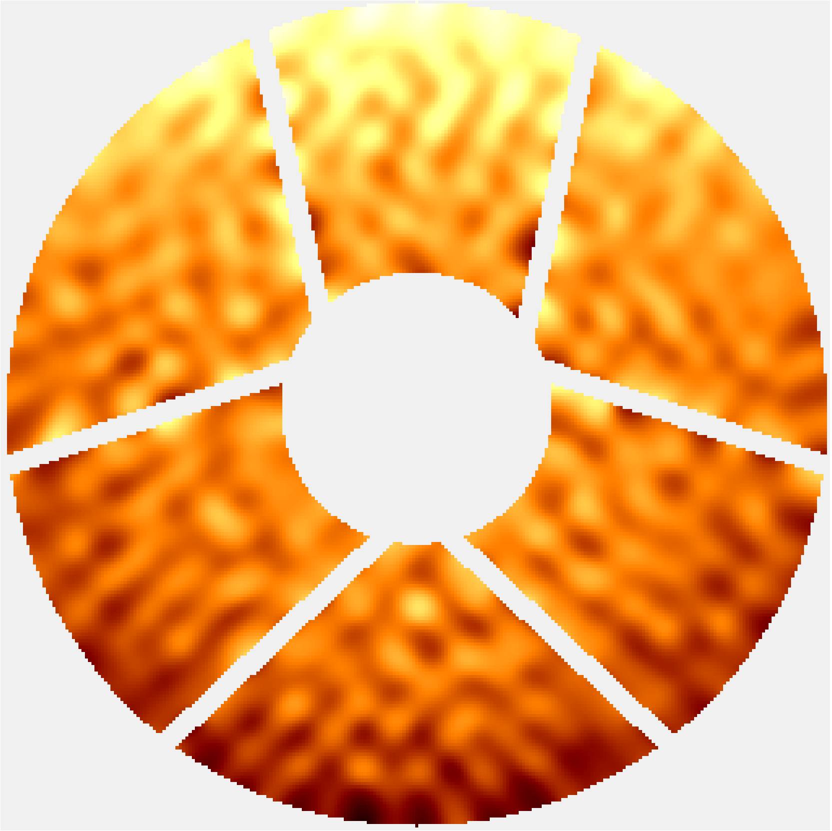


Gerchberg-Saxton Phase

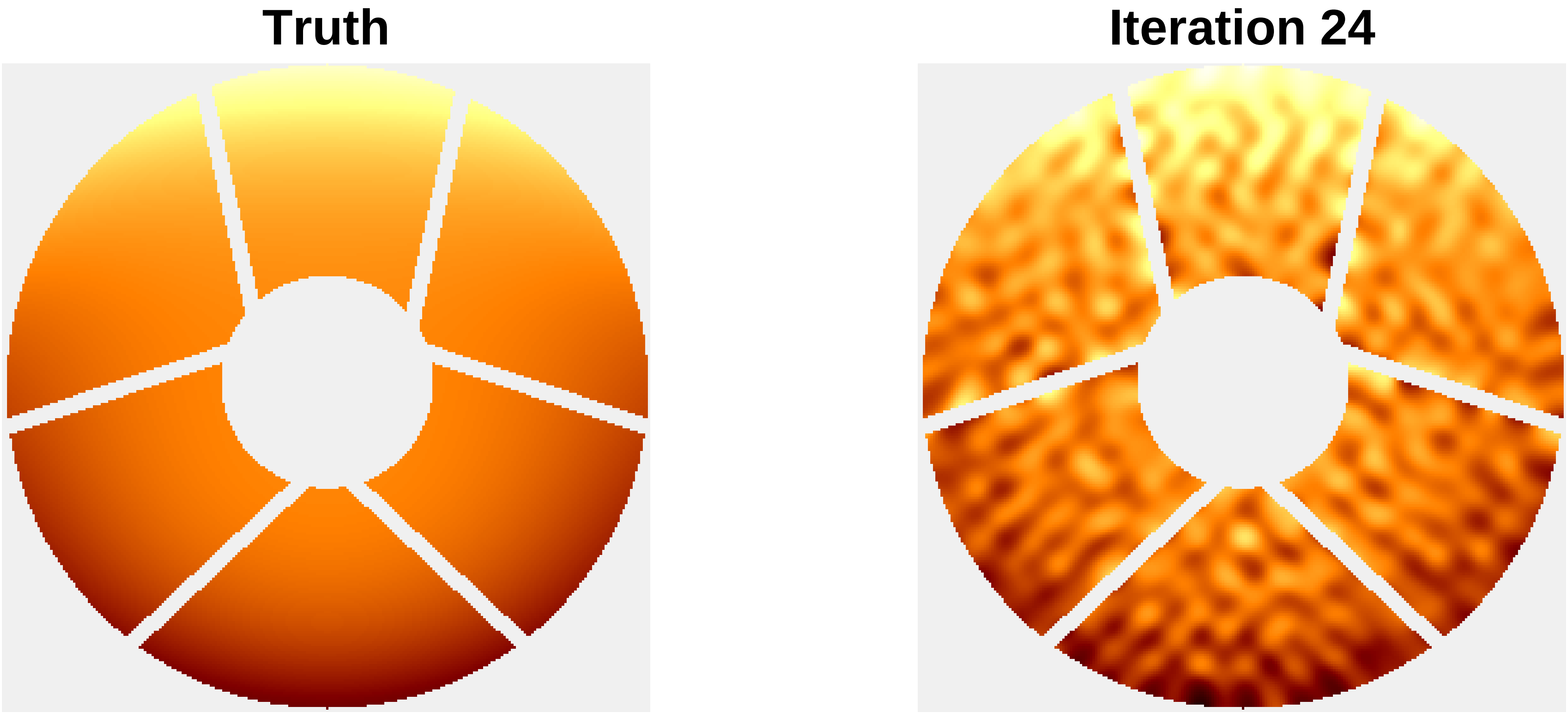
Truth



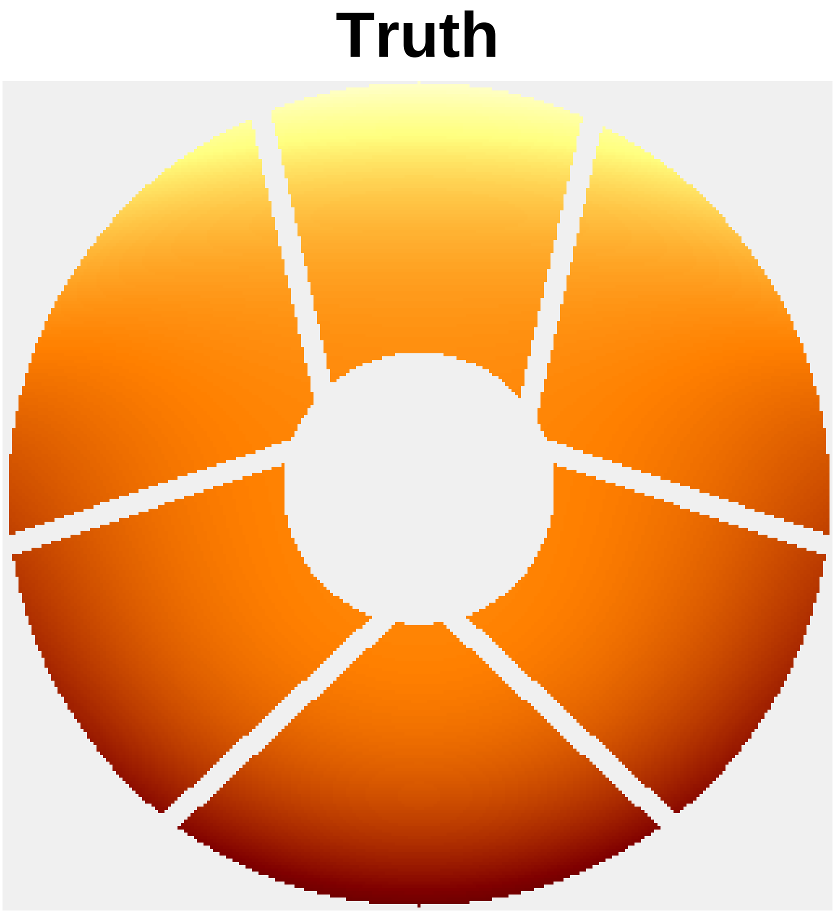
Iteration 23

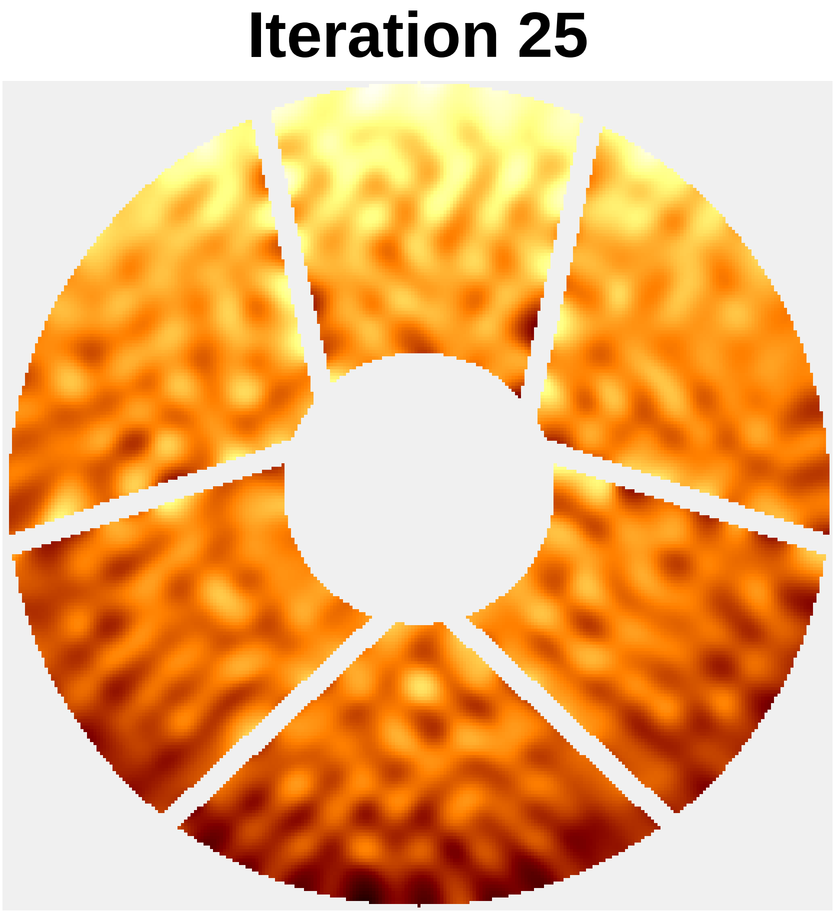


Gerchberg-Saxton Phase



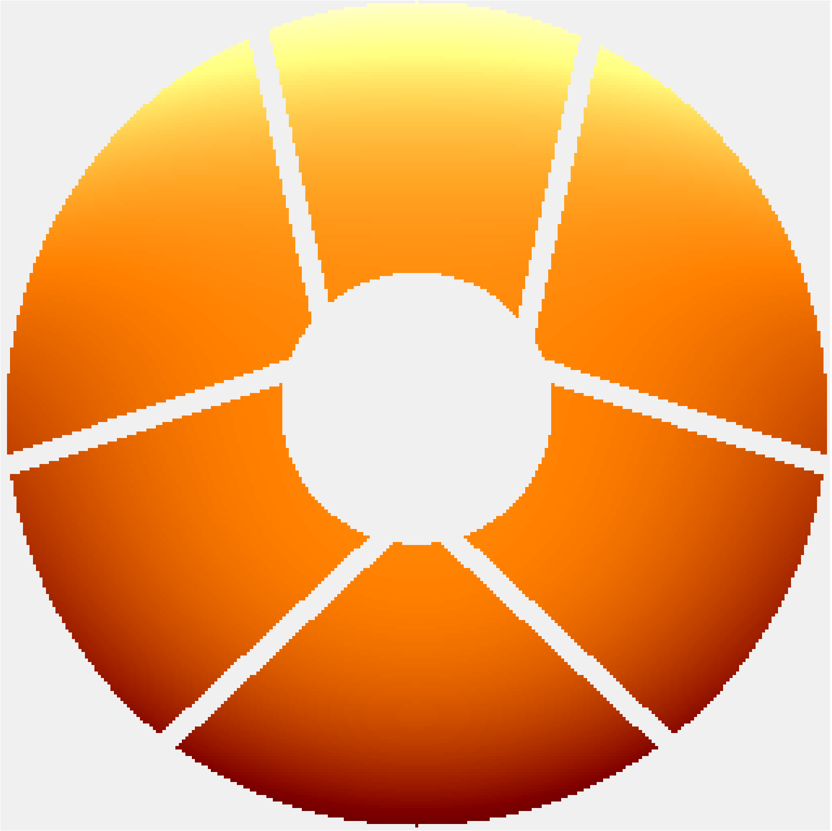
Gerchberg-Saxton Phase



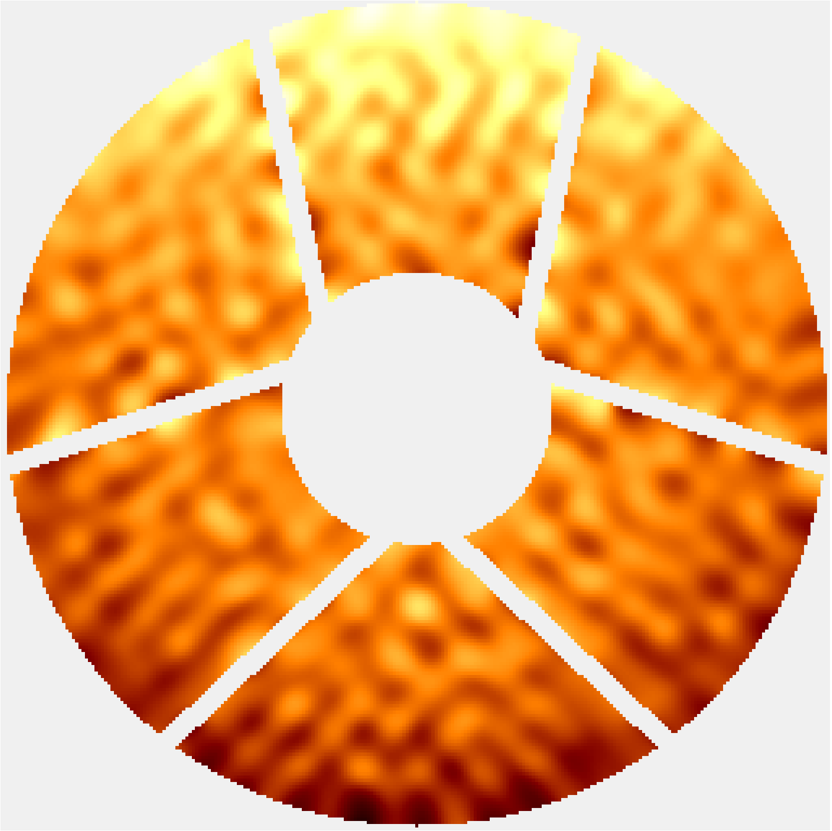


Gerchberg-Saxton Phase

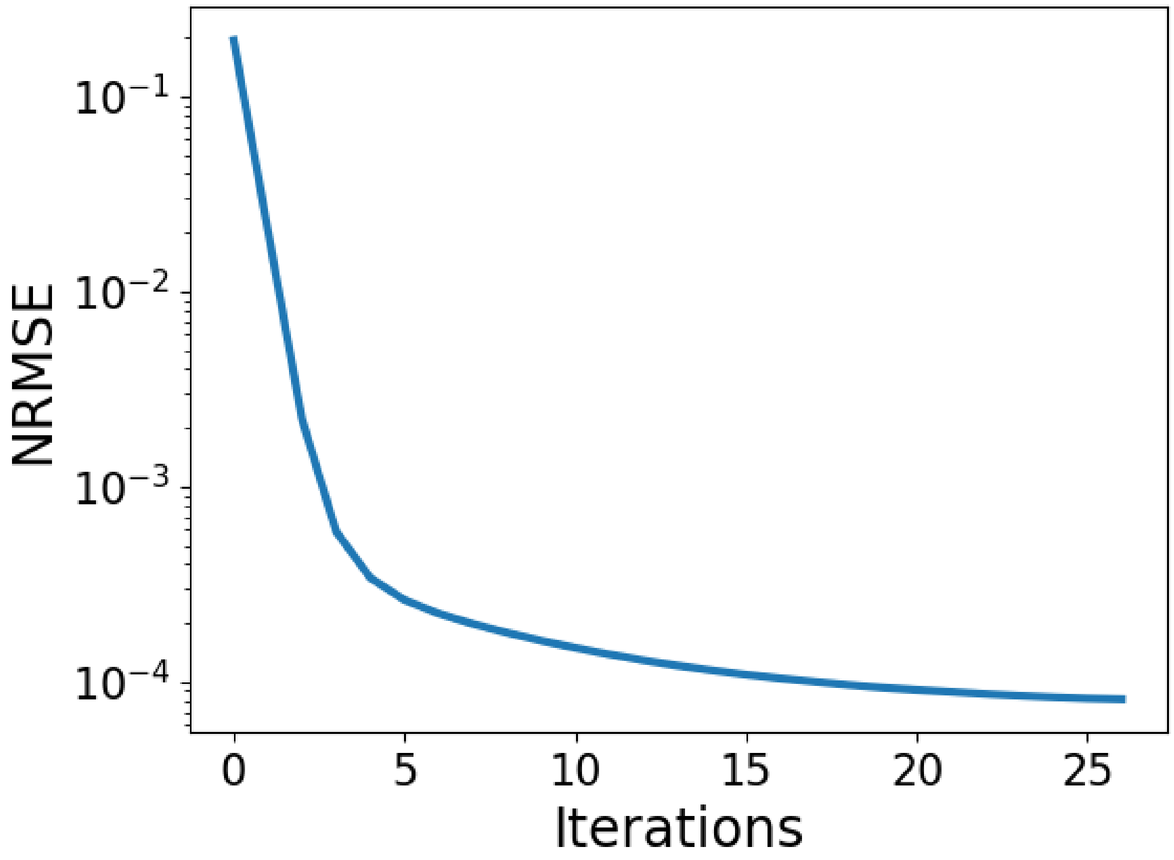
Truth



Iteration 26

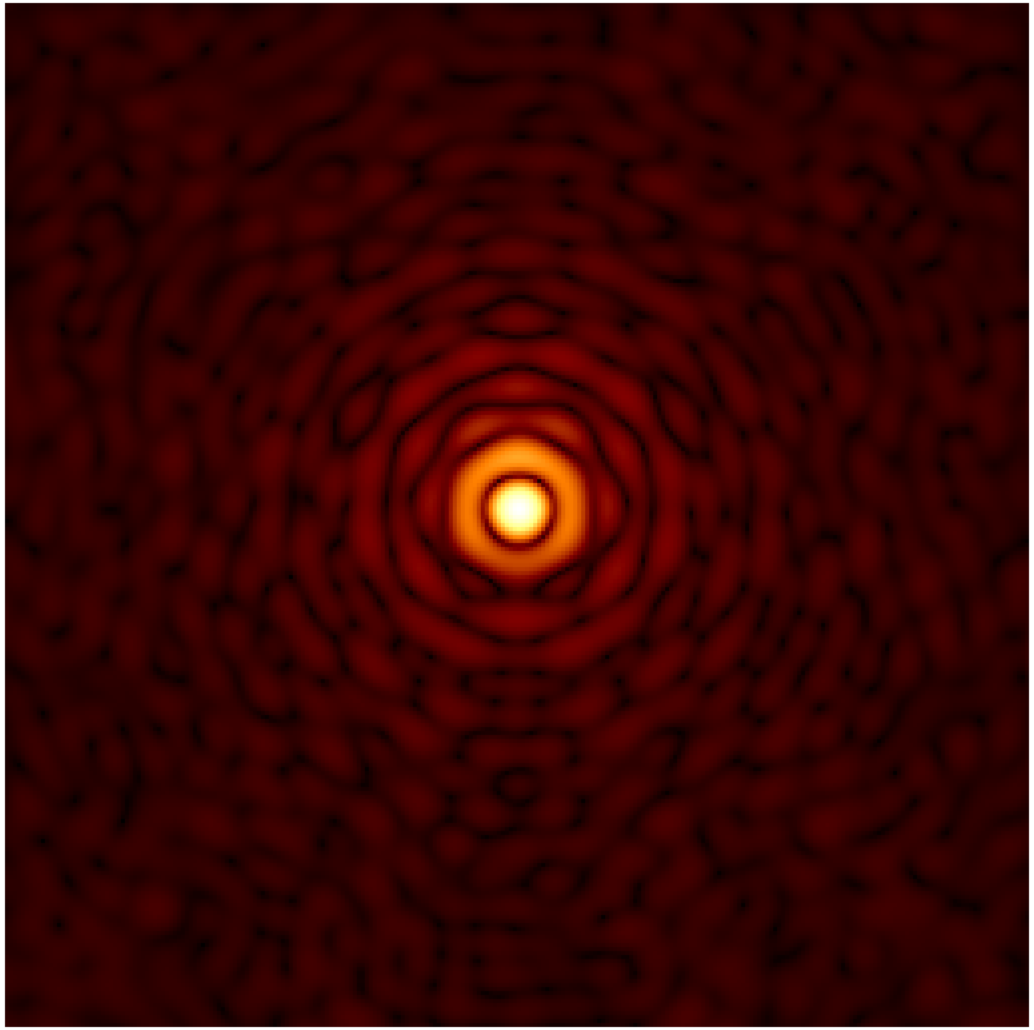


Gerchberg-Saxton Phase



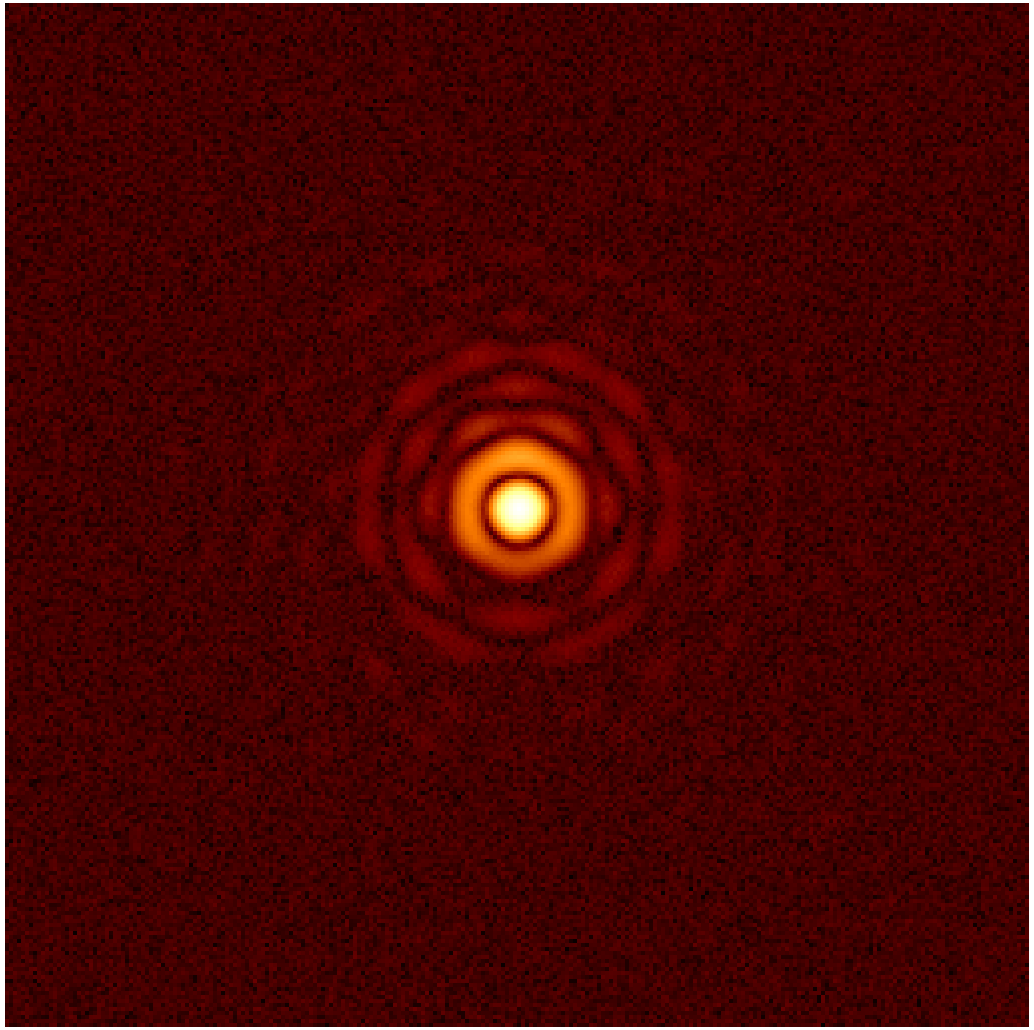
Normalized RMS error on generated PSF versus iteration for Gerchberg-Saxton Phase Retrieval

Gerchberg-Saxton Phase



Found PSF

Gerchberg-Saxton Phase



Data PSF

Gerchberg-Saxton problems

- Overfitting
  - Including noise in PSF update causes “quilting” in phase
  - Can attempt to fix by projecting phase onto polynomial basis
- Error not guaranteed to decrease
  - Generally, error will decrease, but error measurement is not coupled to update

Nonlinear Optimization

- Create a physical modelling function
  - Input is parameters to be optimized
  - Output is single-value cost function to decrease
    - MSE
    - NRMSE
    - Bias-and-gain invariant NMSE [1]
- Obtain search direction
  - Gradient-based methods
    - Finite differences
    - Algorithmic differentiation
  - Stochastic methods also exist
- Update parameters using search direction

1. S. T. Thurman and J. R. Fienup, “Phase retrieval with signal bias,” J. Opt. Soc. Am. A **26**, 1008–1014 (2009).

Finite Differences

- Approximate gradient using small steps:

$$\frac{\partial E}{\partial x_i} \approx \frac{f\left(\vec{\mathbf{x}} + \overrightarrow{\boldsymbol{\delta}_{\mathbf{i}}}(\epsilon)\right) - f(\vec{\mathbf{x}})}{\epsilon}$$

  - $\vec{\mathbf{x}}$ is a the current estimate of the parameters, represented as an array
  - $\overrightarrow{\boldsymbol{\delta}_{\mathbf{i}}}(\epsilon)$ is an array that is zero everywhere except for i, where is has a value of ϵ
    - ϵ is known as “step size”
- Requires many evaluations of modelling function
  - Can only “probe” one parameter at a time

Finite Differences

- Good for functions that have few parameters, non-analytical functions, and a fast physical model
- Fall apart for functions with many parameters
  - Complexity scales with number of input parameters

Algorithmic Differentiation

- Use chain rule to determine gradients:

$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial f}\frac{\partial f}{\partial x_i}$$

Algorithmic Differentiation Example

- Forward:

$$g_n = \cos(4\, a_n)$$

$$m_n = g_n^2$$

$$l_n = \exp\left(-\frac{m_n}{2}\right)$$

$$E = \sum_n (l_n - D_n)^2$$

- Reverse (note that $\bar{y}$ indicates derivative of y with respect to E):

$$\overline{l_n} = 2(l_n - D_n)$$

$$\overline{m_n} = \overline{l_n}\left[-\frac{1}{2}\exp\left(-\frac{m_n}{2}\right)\right]$$

$$\overline{g_n} = \overline{m_n}(2g_n)$$

$$\overline{a_n} = \overline{g_n}[-4\sin(4a_n)]$$

Algorithmic Differentiation

- Modular
  - Including another function means including another gradient step
  - Removing a function means removing a gradient step
- Exact
  - Derived from analytic equations
  - Holds as long at functions are differentiable
- Efficient
  - Requires fewer calculations than finite differences

Algorithmic Differentiation

- Derivatives for common operations with complex operations have been published by Jurling et al [2]
  - Fourier transforms
  - Basis set expansion
  - Complex exponentiation

2. A. S. Jurling and J. R. Fienup, "Applications of algorithmic differentiation to phase retrieval algorithms," J. Opt. Soc. Am. A **31**, 1348–1359 (2014).

Building a Scalar Phase Retrieval Model

- Use wave theory to propagate field from pupil plane to image plane
  - Pupil plane contains total pupil function, total wavefront error from entire system
  - Computationally simple

- Use wave theory to propagate to each surface individually
  - Each surface has contributing pupil function, wavefront error
  - More computationally complex

Pupil plane

- Wavefront
  - Best expressed via basis set such as Zernikes – prevents fitting to noise
  - Can include point-by-point wavefront in addition to Zernike basis to fit higher-order features

- Amplitude
  - Most simply the pupil function of the system
  - Can also express as sum of Zernikes for non-uniform illumination

Propagation

- Must have sampling information of system, establish sampling quotient:

$$Q = \frac{\lambda\left(f/_{\#}\right)}{d_x}$$

  - λ is wavelength, d_x is pixel pitch of detector plane, $f/_{\#}$ is f-number of system
  - $Q = 2$ is Nyquist sampling in detector plane, $Q < 2$ can lead to aliasing in simulation, $Q > 2$ is an oversampled detector
- Pad pupil plane with zeros to size $P = Q\ N$, where N is the size of one side of a square array that just encapsulated the entire pupil function
  - Crop intensity in image plane to size of detector window

Scalar Phase Retrieval Example (from [2])

Step	Forward Model	Reverse Model
Express wavefront with Zernikes	$W(u,v) = \sum_n a_n Z_n(u,v)$	$\overline{a_n} = \sum_{u,v} \bar{W}(u,v) Z_n(u,v)$
Create field with pupil function and wavefront	$g(u,v) = A(u,v) \exp\left[i \frac{2\pi}{\lambda} W(u,v)\right]$	$\bar{W}(u,v) = \frac{2\pi}{\lambda} \Im\{\bar{g}(u,v) g^*(u,v)\}$
Propagate field to image plane	$G(\xi,\eta) = \mathcal{F}_{\substack{u \to \xi \\ v \to \eta}}\{g(u,v)\}$	$\bar{g}(u,v) = \mathcal{F}^{-1}_{\substack{\xi \to u \\ \eta \to v}}\{\bar{G}(\xi,\eta)\}$
Take modulus of image plane to obtain intensity	$I(\xi,\eta) = \lvert G(\xi,\eta) \rvert^2$	$\bar{G}(\xi,\eta) = 2\bar{I}(\xi,\eta) G(\xi,\eta)$
Take weighted sum of square differences for cost function	$E = \sum_{\xi,\eta} w(\xi,\eta)[I(\xi,\eta) - D(\xi,\eta)]^2$	$\bar{I}(\xi,\eta) = 2w(\xi,\eta)[I(\xi,\eta) - D(\xi,\eta)]$

Tips for Reverse Model Formulation

- Ensure that dimensionality matches (e.g. $\bar{I}$ should be 2-dimensional)
- Ensure that real outputs have real gradients, complex outputs have complex gradients
  - $\bar{G}$ should be complex, but $\bar{W}$ should be entirely real-valued
- Use finite differences to ensure that gradients are correct
  - Difference between algorithmic differentiation and finite differences should be on the same order of magnitude as step size ϵ

Incorporating polarization aberrations

- Scalar model cannot account for polarization aberrations
  - Polarizing elements in system
  - Reflective elements with light coming in near Brewster's angle
  - Birefringence
- Use combination of methods from Breckenridge et al in [3], Yamamoto et al in [4]

3. J. B. Breckinridge, W. S. T. Lam, and R. A. Chipman, "Polarization aberrations in astronomical telescopes: The point spread function," Publ. Astron. Soc. Pac. **127**, 445 (2015).
4. N. Yamamoto, J. Kye, and H. J. Levinson, "Polarization aberration analysis using Pauli-Zernike representation," Proc. SPIE **6520**, 6520 – 6520 – 12 (2007).

Jones Pupils

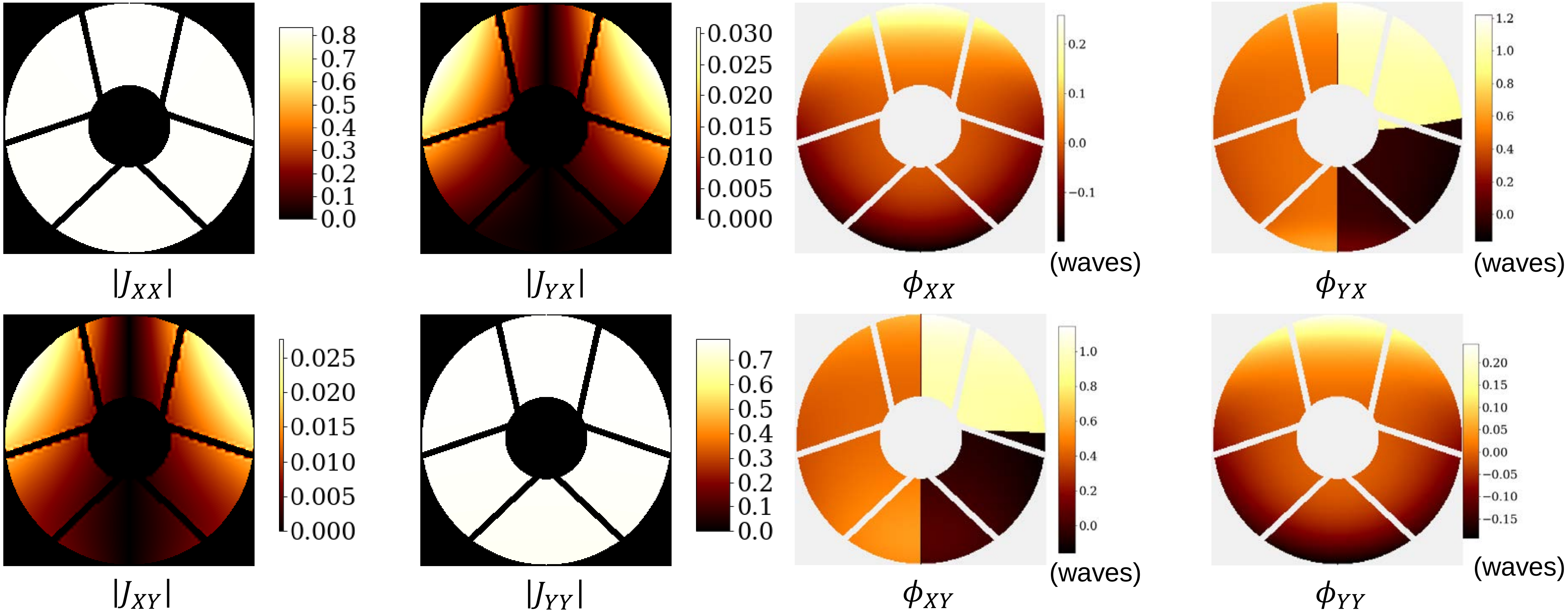
- For each sampled spatial point in arbitrary pupil plane, there is a Jones matrix
  - Describes how polarized light evolves in the system for that spatial point
- Create 2x2 array of pupil planes:

$$\begin{pmatrix} J_{XX}(u,v) & J_{XY}(u,v) \\ J_{YX}(u,v) & J_{YY}(u,v) \end{pmatrix}$$

  - For a given J_{ij}, i is output polarization state and j is input polarization state

Example: Jones Pupils for Wide-Field Interferometric Telescope (WFIRST)

- Obtained from raytrace of on-axis field point



Amplitude Response Matrix (ARM)

- Formed by propagating each Jones pupil element separately to image plane:

$$\begin{pmatrix} J_{XX}(u,v) & J_{XY}(u,v) \\ J_{YX}(u,v) & J_{YY}(u,v) \end{pmatrix} \xrightarrow[v\to\eta]{\mathcal{F}_{u\to\xi}} \begin{pmatrix} ARM_{XX}(\xi,\eta) & ARM_{XY}(\xi,\eta) \\ ARM_{YX}(\xi,\eta) & ARM_{YY}(\xi,\eta) \end{pmatrix}$$

- Propagation performed the same as with scalar theory

Point-Spread Matrix (PSM)

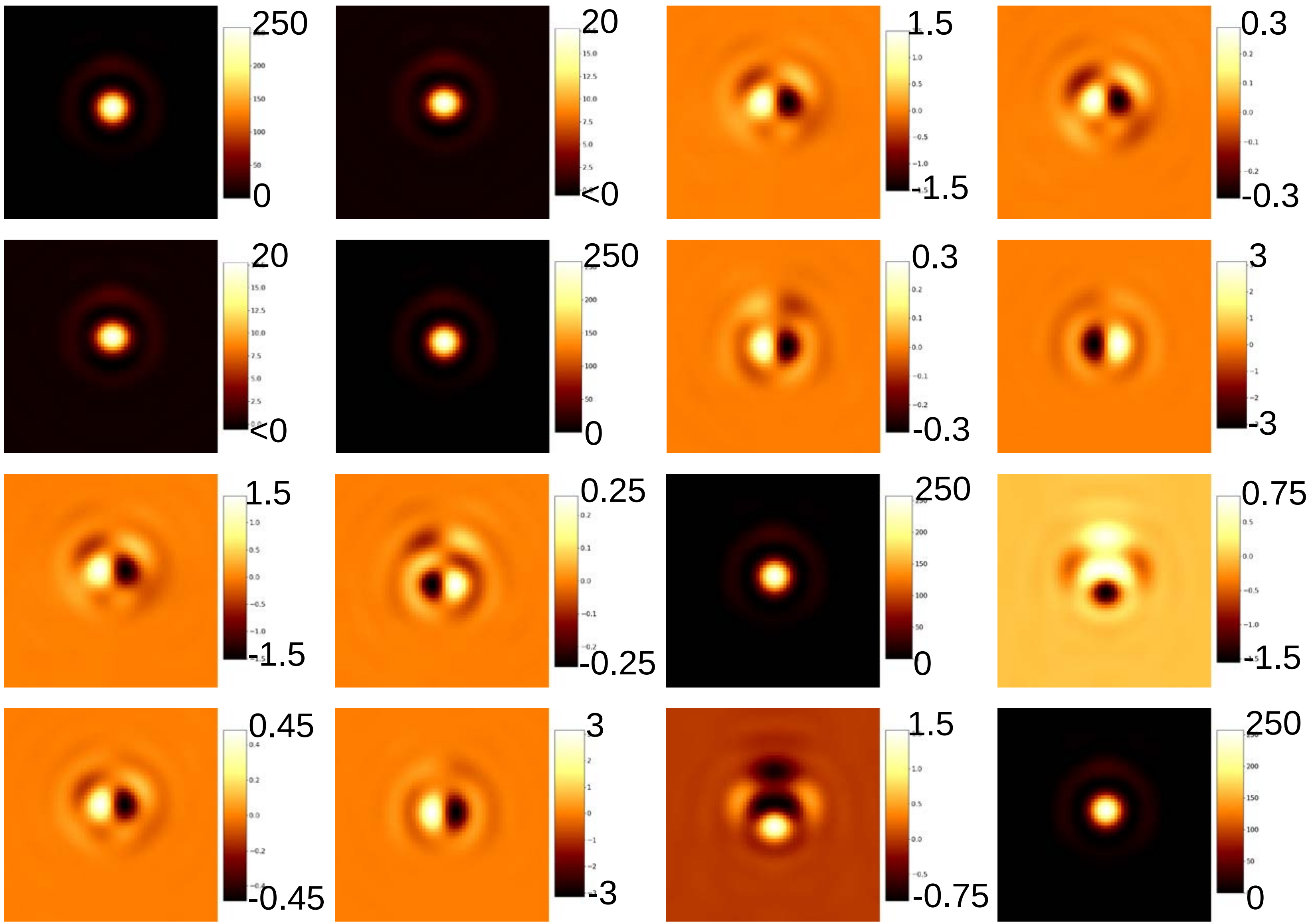
- Entirely real-valued

- For each spatial point (ξ, η), use methodology to turn Jones matrix into Mueller matrix using ARM components [5]

5. H. Fujiwara, “Jones-Mueller Matrix Conversion,” in “Spectroscopic Ellipsometry: Principles and Applications,” (John Wiley & Sons, Ltd, Chichester, UK, 2007), chap. Appendix 4, pp. 353–355.

PSM for WFIRST

$$\begin{pmatrix} PSM_{11} & PSM_{12} & PSM_{13} & PSM_{14} \\ PSM_{21} & PSM_{22} & PSM_{23} & PSM_{24} \\ PSM_{31} & PSM_{32} & PSM_{33} & PSM_{34} \\ PSM_{41} & PSM_{42} & PSM_{43} & PSM_{44} \end{pmatrix}$$



PSF Formulation

- Multiplying PSM by a Stokes vector will give length 4 vector of real-valued arrays
  - First element is total intensity
  - Remaining 3 elements are indicative of degree of X/Y, 45/135, and R/L polarization

- For formulating intensity, we only need a weighted sum of first four PSM elements

PSF Formulation

$$PSM_{11}(\xi,\eta) = \frac{1}{2}\left[|ARM_{XX}(\xi,\eta)|^2 + |ARM_{YY}(\xi,\eta)|^2 + |ARM_{YX}(\xi,\eta)|^2 + |ARM_{XY}(\xi,\eta)|^2\right]$$

$$PSM_{12}(\xi,\eta) = \frac{1}{2}\left[|ARM_{XX}(\xi,\eta)|^2 - |ARM_{YY}(\xi,\eta)|^2 + |ARM_{YX}(\xi,\eta)|^2 - |ARM_{XY}(\xi,\eta)|^2\right]$$

$$PSM_{13}(\xi,\eta) = \Re\{ARM_{XX}(\xi,\eta)ARM_{XY}^*(\xi,\eta)\} + \Re\{ARM_{YX}(\xi,\eta)ARM_{YY}^*(\xi,\eta)\}$$

$$PSM_{14}(\xi,\eta) = -\Im\{ARM_{XX}^*(\xi,\eta)ARM_{XY}(\xi,\eta)\} - \Im\{ARM_{YX}^*(\xi,\eta)ARM_{YY}(\xi,\eta)\}$$

$$I(\xi,\eta) = \sum_n S_n PSM_{1n}(\xi,\eta)$$

Pauli-Zernike Coefficients

- Jones pupil is difficult to separate into scalar and polarization-specific aberrations
- At each spatial point (u, v), decompose Jones matrix using Pauli matrices to obtain spatial coefficients, known as Pauli pupils:

$$\mathbf{J}(u,v) = \sum_n a_n(u,v)\boldsymbol{\sigma}_n$$

$$\boldsymbol{\sigma}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \boldsymbol{\sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\boldsymbol{\sigma}_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \boldsymbol{\sigma}_3 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

Pauli-Zernike Coefficients

- For a Jones matrix at a given spatial point (u, v) :

$$a_0(u,v) = \frac{J_{XX}(u,v) + J_{YY}(u,v)}{2}$$

$$a_1(u,v) = \frac{J_{XX}(u,v) - J_{YY}(u,v)}{2}$$

$$a_2(u,v) = \frac{J_{YX}(u,v) + J_{XY}(u,v)}{2}$$

$$a_3(u,v) = \frac{J_{YX}(u,v) - J_{XY}(u,v)}{2i}$$

Pauli-Zernike Coefficients

- Converting back is simple:

$$J_{XX}(u,v) = a_0(u,v) + a_1(u,v)$$
$$J_{YY}(u,v) = a_0(u,v) - a_1(u,v)$$
$$J_{XY}(u,v) = a_2(u,v) - ia_3(u,v)$$
$$J_{YX}(u,v) = a_2(u,v) + ia_3(u,v)$$

Pauli-Zernike coefficients

- Amplitude and phase of a_0 is the amplitude and phase of the system with no polarization aberrations
  - If $a_1 = a_2 = a_3 = 0$, $J_{XX} = J_{YY} = a_0$, and $J_{XY} = J_{YX} = 0$
  - Thus, $ARM_{XX} = ARM_{YY} = \mathcal{F}\{a_0\}$, $ARM_{XY} = ARM_{YX} = 0$
  - $PSM_{11} = \frac{1}{2}(|ARM_{XX}|^2 + |ARM_{YY}|^2 + |ARM_{XY}|^2 + |ARM_{YX}|^2) = |\mathcal{F}\{a_0\}|^2$
  - $PSM_{12} = \frac{1}{2}(|ARM_{XX}|^2 - |ARM_{YY}|^2 + |ARM_{XY}|^2 - |ARM_{YX}|^2) = 0$
  - $PSM_{13} = \Re(ARM_{XX}ARM_{YX}^*) + \Re(ARM_{YX}ARM_{YY}^*) = 0$
  - $PSM_{14} = -\Im(ARM_{XX}^*ARM_{YX}) - \Im(ARM_{YX}^*ARM_{YY}) = 0$
- Regardless of Stokes vector, total intensity will simply be $|\mathcal{F}\{a_0\}|^2$, which is scalar wavefront theory

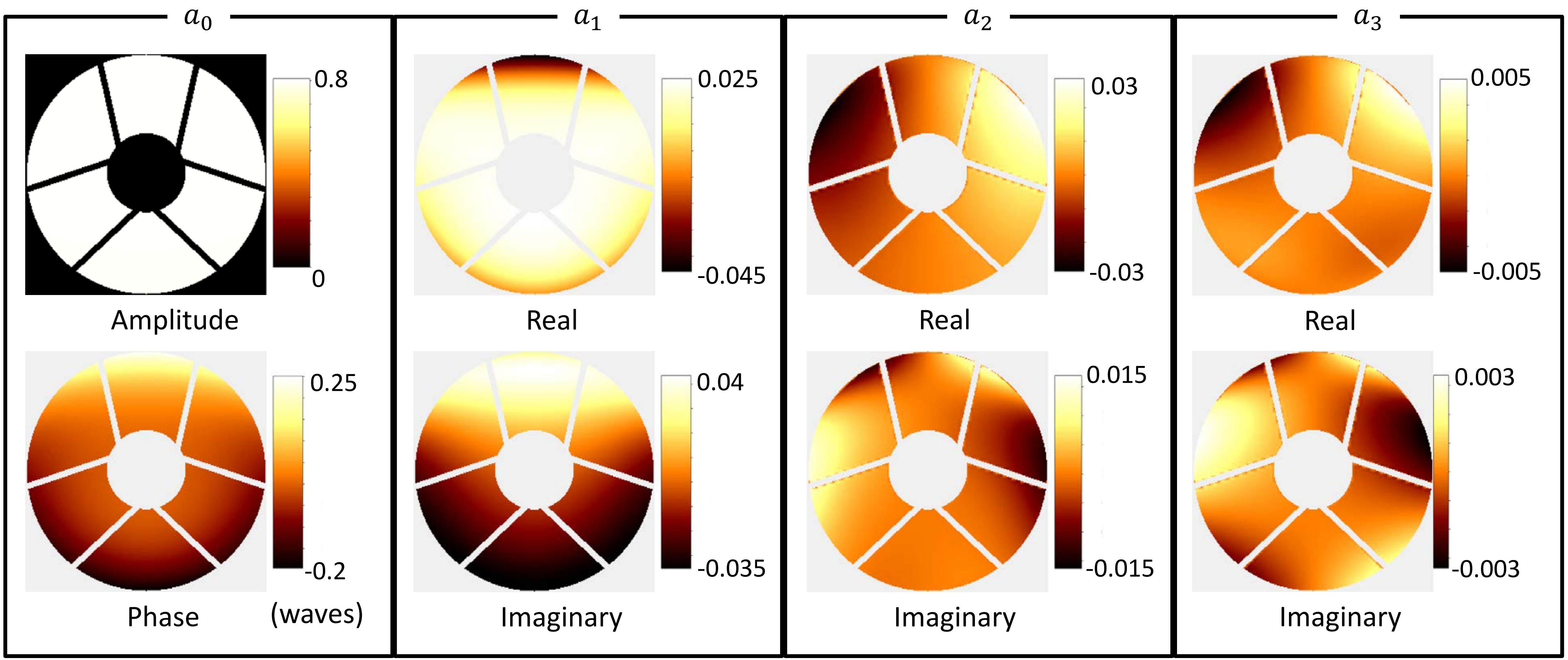
Pauli-Zernike coefficients

- Each Pauli matrix corresponds to an eigenpolarization state:
  - σ_0 has unpolarized eigenvectors (degenerate)
  - σ_1 has X/Y linearly polarized eigenvectors
  - σ_2 has 45/135 deg. linearly polarized eigenvectors
  - σ_3 has circularly polarized eigenvectors

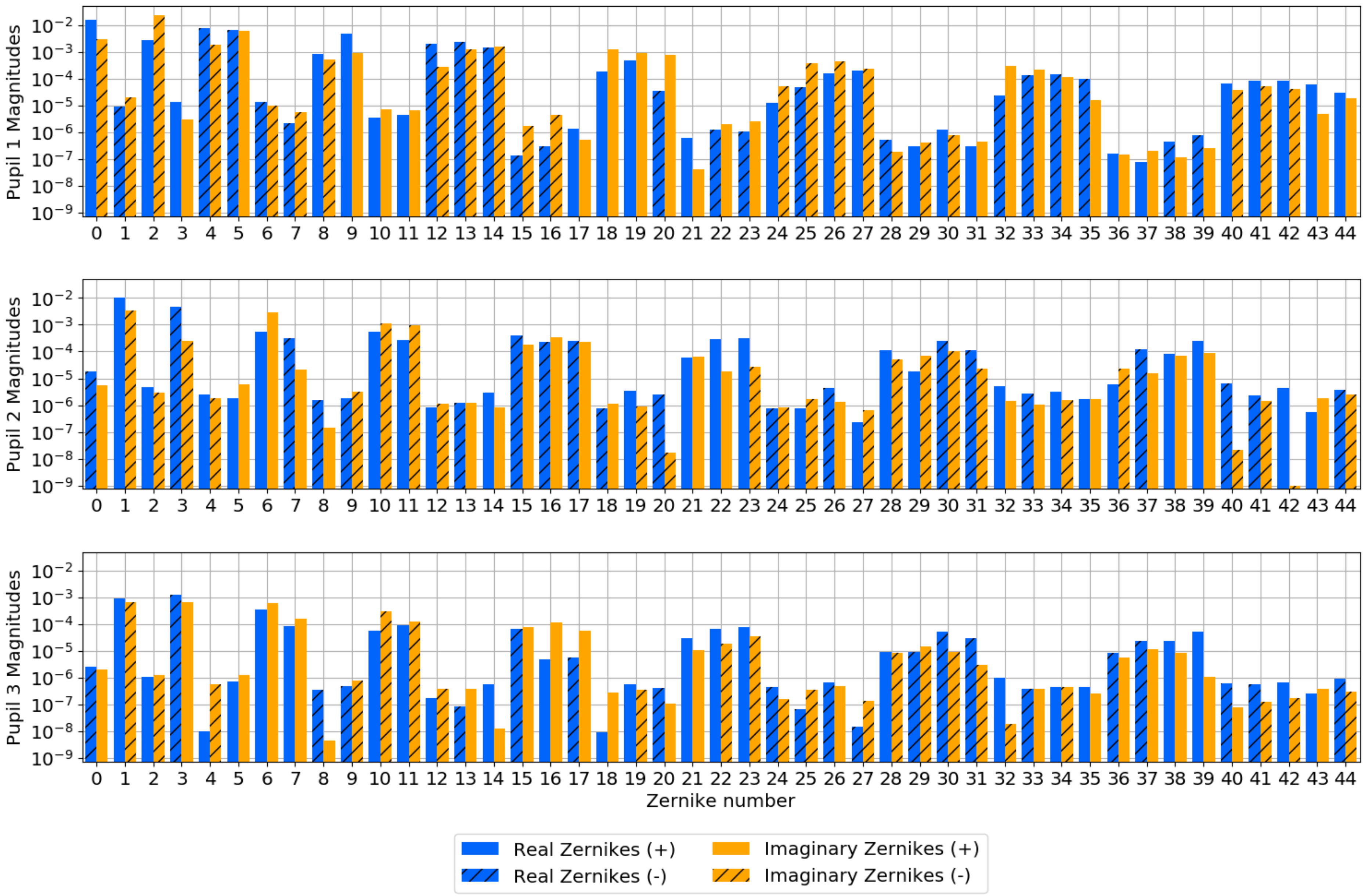
Pauli-Zernike coefficients

- Represent amplitude and phase of a_0 using Zernike decomposition
  - Same as scalar model
- For a_1, a_2, and a_3, represent real and imaginary parts using Zernike decomposition
- Can perform simulation with known amounts of polarization
  - Can parameterize polarization aberration using Zernike coefficients for optimization purposes
- Can adjust scalar wave phase and amplitude independently of polarization effects

WFIRST Pauli Pupils



WFIRST Pauli Pupils



Total Forward Model – Pauli Zernike-Coefficients

- We have a set of Zernike coefficients c_{mn} where n corresponds to Zernike index and m corresponds to the Pauli pupils as follows:

m	Representation
0	Amplitude of a_0
1	Phase of a_0
2	$\mathfrak{R}\{a_1\}$
3	$\mathfrak{I}\{a_1\}$
4	$\mathfrak{R}\{a_2\}$
5	$\mathfrak{I}\{a_2\}$
6	$\mathfrak{R}\{a_3\}$
7	$\mathfrak{I}\{a_3\}$

Total Forward Model – Generating a_0

$$A(u,v) = \sum_n c_{0n} Z_n(u,v)$$

$$\phi(u,v) = \frac{2\pi}{\lambda} \sum_n c_{1n} Z_n(u,v)$$

$$a_0(u,v) = A(u,v)\exp[i\phi(u,v)]$$

Total Forward Model – Other Pauli Pupils

$$a_1(u,v) = \left[\sum_n c_{2n} Z_n(u,v)\right] + i\left[\sum_n c_{3n} Z_n(u,v)\right]$$

$$a_2(u,v) = \left[\sum_n c_{4n} Z_n(u,v)\right] + i\left[\sum_n c_{5n} Z_n(u,v)\right]$$

$$a_3(u,v) = \left[\sum_n c_{6n} Z_n(u,v)\right] + i\left[\sum_n c_{7n} Z_n(u,v)\right]$$

Total Forward Model – Pauli to Jones Conversion

$$J_{XX}(u,v) = a_0(u,v) + a_1(u,v)$$

$$J_{YY}(u,v) = a_0(u,v) - a_1(u,v)$$

$$J_{XY}(u,v) = a_2(u,v) - ia_3(u,v)$$

$$J_{YX}(u,v) = a_2(u,v) + ia_3(u,v)$$

Total Forward Model - Propagation

$$\begin{pmatrix} J_{XX}(u,v) & J_{XY}(u,v) \\ J_{YX}(u,v) & J_{YY}(u,v) \end{pmatrix} \xrightarrow[v\to\eta]{\mathcal{F}_{u\to\xi}} \begin{pmatrix} ARM_{XX}(\xi,\eta) & ARM_{XY}(\xi,\eta) \\ ARM_{YX}(\xi,\eta) & ARM_{YY}(\xi,\eta) \end{pmatrix}$$

Total Forward Model – Forming the PSM

$$PSM_{11}(\xi,\eta) = \frac{1}{2}\left[|ARM_{XX}(\xi,\eta)|^2 + |ARM_{YY}(\xi,\eta)|^2 + |ARM_{YX}(\xi,\eta)|^2 + |ARM_{XY}(\xi,\eta)|^2\right]$$

$$PSM_{12}(\xi,\eta) = \frac{1}{2}\left[|ARM_{XX}(\xi,\eta)|^2 - |ARM_{YY}(\xi,\eta)|^2 + |ARM_{YX}(\xi,\eta)|^2 - |ARM_{XY}(\xi,\eta)|^2\right]$$

$$PSM_{13}(\xi,\eta) = \Re\{ARM_{XX}(\xi,\eta)ARM_{XY}^{*}(\xi,\eta)\} + \Re\{ARM_{YX}(\xi,\eta)ARM_{YY}^{*}(\xi,\eta)\}$$

$$PSM_{14}(\xi,\eta) = -\Im\{ARM_{XX}^{*}(\xi,\eta)ARM_{XY}(\xi,\eta)\} - \Im\{ARM_{YX}^{*}(\xi,\eta)ARM_{YY}(\xi,\eta)\}$$

Total Forward Model – Weighted Stokes Summation

$$I(\xi,\eta) = \sum_n S_n PSM_{1n}(\xi,\eta)$$

- From here, we have PSF intensity, which can be fed into a metric to obtain our error metric value
- Value of $\bar{I}(\xi,\eta)$ dependent on metric choice

Total Reverse Model – Gradients for Stokes and PSM components

$$\overline{PSM_{1n}}(\xi,\eta) = S_n \bar{I}(\xi,\eta)$$

$$\overline{S_n} = \sum_{\xi,\eta} PSM_{1n}(\xi,\eta)\bar{I}(\xi,\eta)$$

Total Reverse Model – Gradient for ARM

$$\overline{ARM_{XX}} = ARM_{XX}[\overline{PSM_{11}} + \overline{PSM_{12}}] + ARM_{XY}[\overline{PSM_{13}} + i\overline{PSM_{14}}]$$

$$\overline{ARM_{YY}} = ARM_{YY}[\overline{PSM_{11}} - \overline{PSM_{12}}] + ARM_{YX}[\overline{PSM_{13}} - i\overline{PSM_{14}}]$$

$$\overline{ARM_{XY}} = ARM_{XY}[\overline{PSM_{11}} - \overline{PSM_{12}}] + ARM_{XX}[\overline{PSM_{13}} - i\overline{PSM_{14}}]$$

$$\overline{ARM_{YX}} = ARM_{YX}[\overline{PSM_{11}} + \overline{PSM_{12}}] - ARM_{YY}[\overline{PSM_{13}} + i\overline{PSM_{14}}]$$

Total Reverse Model – Gradient for Jones Pupil

$$\begin{pmatrix} \overline{ARM}_{XX}(\xi,\eta) & \overline{ARM}_{XY}(\xi,\eta) \\ \overline{ARM}_{YX}(\xi,\eta) & \overline{ARM}_{YY}(\xi,\eta) \end{pmatrix} \xrightarrow[\eta\to v]{\mathcal{F}^{-1}_{\xi\to u}} \begin{pmatrix} \bar{J}_{XX}(u,v) & \bar{J}_{XY}(u,v) \\ \bar{J}_{YX}(u,v) & \bar{J}_{YY}(u,v) \end{pmatrix}$$

Total Reverse Model – Gradient for Pauli Pupils

$$\bar{a}_0(u,v) = \bar{J}_{XX}(u,v) + \bar{J}_{YY}(u,v)$$

$$\bar{a}_1(u,v) = \bar{J}_{XX}(u,v) - \bar{J}_{YY}(u,v)$$

$$\bar{a}_2(u,v) = \bar{J}_{XY}(u,v) + \bar{J}_{YX}(u,v)$$

$$\bar{a}_3(u,v) = i\,[\bar{J}_{XY}(u,v) - \bar{J}_{YX}(u,v)]$$

Total Reverse Model – Gradients for phase/amplitude of a_0

$$\bar{A}(u,v) = \Re\{\bar{a}_0(u,v)\exp[-i\,\phi(u,v)]\}$$

$$\bar{W}(u,v) = \frac{2\pi}{\lambda}\Im\{\bar{a}_0(u,v)a_0^*(u,v)\}$$

Total Reverse Model – Gradients for Zernike coefficients

$$\bar{c}_{0n} = \sum_{u,v} \bar{A}(u,v) Z_n(u,v)$$

$$\bar{c}_{1n} = \sum_{u,v} \bar{W}(u,v) Z_n(u,v)$$

Total Reverse Model – Gradients for Zernike coefficients

$$\bar{c}_{2n} = \sum_{u,v} \Re\{\bar{a}_1(u,v)\} Z_n(u,v)$$

$$\bar{c}_{3n} = \sum_{u,v} \Im\{\bar{a}_1(u,v)\} Z_n(u,v)$$

Total Reverse Model – Gradients for Zernike coefficients

$$\bar{c}_{4n} = \sum_{u,v} \Re\{\bar{a}_2(u,v)\} Z_n(u,v)$$

$$\bar{c}_{5n} = \sum_{u,v} \Im\{\bar{a}_2(u,v)\} Z_n(u,v)$$

Total Reverse Model – Gradients for Zernike coefficients

$$\bar{c}_{6n} = \sum_{u,v} \Re\{\bar{a}_3(u,v)\} Z_n(u,v)$$

$$\bar{c}_{7n} = \sum_{u,v} \Im\{\bar{a}_3(u,v)\} Z_n(u,v)$$

In summary

- Nonlinear optimization for phase retrieval is done best with algorithmic differentiation
- A model with polarization was created, and a reverse model was built according to rules from [2]
  - Uses Pauli-Zernike coefficients, PSF formulation from [3]
  - Allows for optimization of scalar aberrations, polarization aberrations, and source polarization

2. A. S. Jurling and J. R. Fienup, “Applications of algorithmic differentiation to phase retrieval algorithms,” J. Opt. Soc. Am. A **31**, 1348–1359 (2014).
3. J. B. Breckinridge, W. S. T. Lam, and R. A. Chipman, “Polarization aberrations in astronomical telescopes: The point spread function,” Publ. Astron. Soc. Pac. **127**, 445 (2015).

QUESTIONS