Modifications to the Diffraction Algorithm in PhoSim for Space Telescopes

Colin Burke*

February 13, 2017

Abstract

The diffraction algorithm in PhoSim version 3.6 is valid only for non diffraction-limited optical systems. However, astronomers have recently expressed interest in using PhoSim for modeling diffraction-limited space telescopes. This approach requires a major modification to the diffraction algorithm in PhoSim—improving the numerical accuracy of the fast Fourier transform of the telescope pupil and correcting the second kick angular displacement scaling. In this work, I describe these modifications to the source code in detail to ensure diffraction-limited telescopes produce accurate point spread functions. Commands are included for simulations of space telescopes, where the atmospheric Kolmogorov diffraction is turned off and the telescope diffraction is calculated correctly. A simple parabolic telescope named "generic" is created to test PhoSim in the diffraction-limit. I show that the new diffraction algorithm in PhoSim, described in this work, produces an accurate intensity distribution on the focal plane which mimics the expected Airy pattern for circular-aperture telescopes. An additional option is implemented for the input of non-trivial aperture shapes via a FITS input file, which is Fourier transformed directly and correctly scaled on the focal plane. This work is an important addition to PhoSim, since the size and morphology of the point spread function is dominated by the diffraction for space telescopes.

1 Introduction

For non diffraction-limited telescopes, the size and morphology of a point spread function (PSF) generated by PhoSim is dictated by atmospheric turbulence and optical aberrations. A non diffraction-limited telescope satisfies the following condition:

$$\max\left[\frac{D}{r_0}, \frac{\alpha}{F}\right] >> \lambda/D \tag{1}$$

where D is the aperture diameter, λ is the photon wavelength, r_0 is the Fried parameter, F is the focal length, and α is the characteristic size of aberrations from the telescope and

*Purdue University, Department of Physics and Astronomy, West Lafayette, IN, 47907. Please email colin.burke.j@gmail.com.

camera. This can be rewritten as

$$\max\left[\frac{D}{r_0}, \frac{\alpha}{\lambda f}\right] >> 1 \tag{2}$$

where the substitution $f = F/D$ is recognized as the focal ratio.

Under the working assumption of a non diffraction-limited system, Peterson et al. (2015) show that the diffraction component of the PSF in the Fraunhofer limit can be approximated as

$$\mathrm{PSF}(\hat{n}) = T\left(\hat{n} - \frac{\lambda}{2\pi}\nabla\phi\right) \tag{3}$$

where $\hat{n}$ is the unit vector pointing to a particular focal plane, $T(\hat{n})$ is the profile of the diffraction from the telescope, and $\nabla\phi$ is the gradient of the "phase screen" ϕ. The phase screen is a representation of a phase shift as a function of spatial position due to atmospheric turbulence.

PhoSim computes the PSF in a sequential two kick photon-by-photon Monte Carlo algorithm which fully accounts for the net effect of the phase screens on all spatial scales:

First Kick: The Fourier transforms of the phase screens are convolved in a manner that results in a time-averaged long exposure, and individual photons are displaced accordingly. The full details of the first kick (atmospheric turbulence) are not relevant to this work; the interested reader may instead consult Peterson et al. (2015), Peterson et al. (2013), or Peterson et al. (2012).

Second Kick: A radially-averaged angular displacement scaling proportional to $\lambda^{-\frac{1}{5}}$ is applied to individual photons, accounting for the overall scaling from the full diffraction calculation (Peterson et al., 2015; Fried, 1965) The C++ code for the second kick function is shown below for version 3.6:

```
int Image::secondKick(Vector *largeAngle, Photon *aph) {

long index;
double r, phi;

r = random.uniform();
find(screen.hffunc, 10000, r, &index);
r = ((double)(index))*0.5*1e-3/(SCREEN_SIZE*screen.fine_sizeperpixel)*aph->wavelengthFactor;
phi = 2*M_PI*random.uniform();
largeAngle->x += r*cos(phi);
largeAngle->y += r*sin(phi);

return(0);

}
```

The `wavelengthFactor` variable is $\lambda^{-\frac{1}{5}}\lambda_{\text{nominal}}$ and the `1e-3` factor is just the unit conversion.

This work is concerned with the algorithm in the limit that the atmospheric turbulence is small. I present a modification to the second kick function which is correct in this special case. Then, I briefly investigate the numerical accuracy of the Fourier transform of the telescope diffraction profile. Finally, I present a modified diffraction algorithm which fully accounts for the two limiting cases with added support for complicated pupil geometries.

2 Modifying Second Kick

One of the benefits of physics-based simulations, like PhoSim, is the ability to easily analyze optical systems with modified physics and environmental parameters. If one wishes to analyze a telescope in an environment where there is no atmospheric turbulence, one may simply turn off the atmospheric turbulence with the physics override command `clearturbulence`. In the limit of no atmospheric turbulence, the atmospheric turbulence strength in the definition of the Fried parameter approaches zero. The Fried parameter is

$$r_0 = \left[.423k^2 \int_0^h C_n^2(z)dz \right]^{-\frac{3}{5}} \tag{4}$$

where k is the wavenumber, $k = 2\pi/\lambda$, and C_n^2 is the atmospheric turbulence strength (Fried, 1965). Noting the negative exponent, it is easily seen that $\lim_{C_n^2 \to 0} r_0 = \infty$. Thus, in the limit $r_0 \to \infty$, the condition for a non diffraction-limited system becomes

$$\frac{\alpha}{F} >> \lambda/D \quad \text{or} \quad \frac{\alpha}{\lambda f} >> 1. \tag{5}$$

As should be expected, the atmospheric turbulence term becomes irrelevant. We see that if $\alpha << \lambda f$, this condition is false and the telescope is diffraction-limited instead. The condition for a diffraction-limited system then becomes

$$\frac{\alpha}{F} << \lambda/D \quad \text{or} \quad \frac{\alpha}{\lambda f} << 1. \tag{6}$$

In this limit, the atmospheric turbulence phase screen is irrelevant because $\nabla\phi \to 0$, and Equation 3 becomes

$$\text{PSF}(\hat{n}) = T\left(\hat{n}\right). \tag{7}$$

Now, the diffraction component of the PSF is determined only by the diffraction profile from the telescope.

In the limit that the atmospheric turbulence is small, this second kick algorithm is no longer physically accurate because an angular displacement scaling proportional to $\lambda^{-\frac{1}{5}}$ no longer applies. However, if the system is *still* non diffraction-limited in the limit that the atmospheric turbulence is small (satisfies Condition 5), then the optical aberrations determine the size and morphology of the PSF and the diffraction is irrelevant to begin with. Therefore, I will only bother to consider diffraction-limited systems for the remainder of this paper.

If the system is diffraction-limited, the second kick angular scaling is proportional to λ. This scaling is consistent with the normal Airy pattern for circular-aperture telescopes. I modify the second kick function accordingly:

```
int Image::secondKick(Vector *largeAngle) {

long index;
double r, phi;

r = RngDouble();
find(screen.hffunc, 10000, r, &index);
if (atmospheremode < 2)
r = ((double)(index))*1e-3/(SCREEN_SIZE*screen.fine_sizeperpixel)*aph->wavelength;
else
r = ((double)(index))*0.5*1e-3/(SCREEN_SIZE*screen.fine_sizeperpixel)*aph->wavelengthFactor;
phi = 2*M_PI*RngDouble();
largeAngle->x += r*cos(phi);
largeAngle->y += r*sin(phi);

return(0);

}
```

The `atmospheremode < 2` condition corresponds to no atmospheric turbulence.

Now, the second kick function is corrected for both limiting cases.

3 Numerical Accuracy

For diffraction-limited systems, PhoSim performs a Fourier transform of a phase screen generated only by the specifications of the pupil diameter. Phase screens of size 1024 × 1024 are used. The following code snippet populates the phase screen with the pupil geometry:

```
if (radius > surface.innerRadius[0] && radius < surface.outerRadius[0]) {
screen.inscreen[SCREEN_SIZE*i + j][0] = cos(screen.phasescreen[i*SCREEN_SIZE + j]*norm);
screen.inscreen[SCREEN_SIZE*i + j][1] = sin(screen.phasescreen[i*SCREEN_SIZE + j]*norm);
} else {
screen.inscreen[SCREEN_SIZE*i + j][0] = 0.0;
screen.inscreen[SCREEN_SIZE*i + j][1] = 0.0;
}
```

The `norm` variable should be set to zero when the atmospheric turbulence is off.

PhoSim then utilizes the 2-dimensional discrete fast Fourier transform (FFT) algorithm via the FFTW C library to Fourier transform the phase screen. The discrete Fourier transform is defined for 2-dimensions as

$$Y_j = \sum_{N=0}^{N-1} e^{-2\pi i j\cdot(j/N)} X_j \tag{8}$$

where $k = 0, \ldots N-1$, and N is the array size (`SCREEN_SIZE`) (Lyons, 2010, p. 60).

A code snippit from the atmosphere diffraction function that calculates the FFT is shown below:

```
pb = fftw_plan_dft_2d(SCREEN_SIZE, SCREEN_SIZE, screen.inscreen, screen.outscreen,
FFTW_FORWARD, FFTW_ESTIMATE);
fftw_execute(pb);
fftw_destroy_plan(pb);
```

With this code, the phase screen generated by PhoSim will have correct units and scaling, but the results will be inconsistent due to numerical differences that arise when using different pupil radii. For extremely large-aperture telescopes which approach the physical size of the phase screen (10.24 meters), the FFT will fail to produce accurate results, either due to insufficient zero padding or simply being larger than the phase screen so that the geometry is not fully enclosed within the phase screen. Additionally, for extremely small-aperture systems, (such as the human eye or a typical backyard telescope), the FFT will be inaccurate due a poor effective resolution of the aperture.

It is standard practice to pad the outside of an array with zeros before the FFT to ensure the algorithm generates a large number of frequency bins (Lyons, 2010, pp. 98–102). The padding factor γ is defined as an integer power of two, which scales the array by a factor of $1/\gamma$, then pads the outside with zeros, keeping the array size the same (Soummer et al., 2007). To visualize this, consider a 1-dimensional array of size 16 with no extra zero padding:

$$0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0$$

and the equivilent array with 4 times zero padding:

$$0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0.$$

Now I have decreased the effective resolution of an entrance pupil by a factor which approaches γ for large array sizes. But I gain accuracy after the FFT. This is a beneficial trade-off. The effects of zero padding on the FFT algorithm can easily be seen in Figure 1 by computing two 1-dimensional discrete FFTs of size 1024: one with no extra padding, and another with $\gamma = 8$. The plot with zero padding (right) obviously results in a much better representation of the expected Fraunhofer intensity distribution due to diffraction through a 1-dimensional entrance pupil (i.e., a slit). However, zero padding is not necessary if the characteristic size of the geometric diffraction is within one pixel.

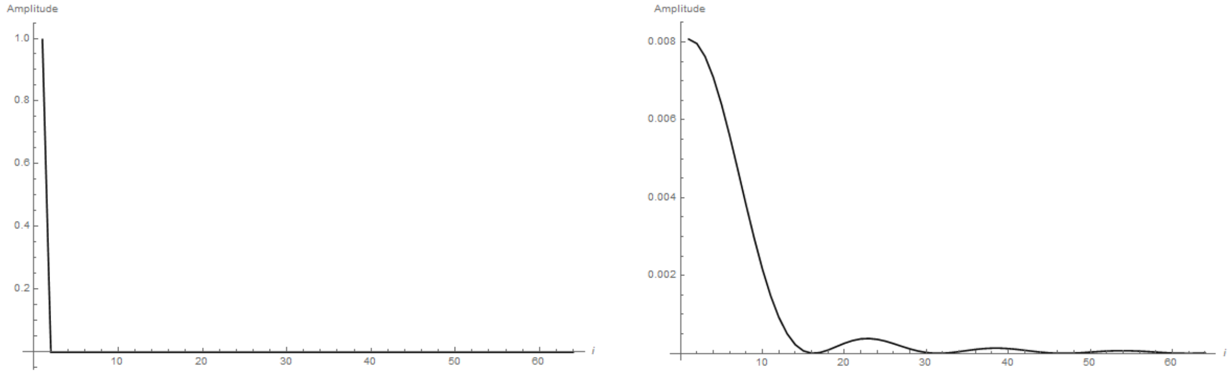


Figure 1: A plot showing the square of the moduli of the discrete FFT outputs normalized to 1 (y-axis), out to $j = 64$ (x-axis) with no zero padding (left) and $\gamma = 8$ (right). Array size: 1024.

This concept is easily extended to 2-dimensional FFTs, where the intensity distribution should result in the Airy pattern for circular-aperture telescopes. I modify the diffraction algorithm to include zero padding. The new radius condition must be independent of the actual pupil size D to use consistent zero padding.

$$\frac{r}{R}\frac{N}{2}\frac{1}{\gamma} < r_\phi < \frac{N}{2}\frac{1}{\gamma} \tag{9}$$

where N is the size of the phase screen, r_ϕ is the iterative positional radius of the phase screen $r_\phi = \sqrt{(i - N/2 + 0.5)^2 + (j - N/2 + 0.5)^2}$ where i and j go from 0 to $N - 1$, R is the primary mirror outer radius, r is the primary mirror inner radius, and γ is the padding factor. Although zero padding is not necessary for sufficiently sized pixels, it does not cost any computation time and the decrease in effective pupil resolution has a minimal effect so long as γ is not too large. A padding factor of $\gamma = 8$ is a typical value (Soummer et al., 2007). The new code snippet for the improved population of the phase screen is shown below:

```
if (atmospheremode < 2) {
seeing = 0.0;
norm = 0.0;
paddingfactor = 32;
rinner = screen.fine_sizeperpixel*SCREEN_SIZE*surface.innerRadius[0]/(2*surface.outerRadius[0]*paddingfactor);
router = screen.fine_sizeperpixel*SCREEN_SIZE/(2*paddingfactor);
} else {
seeing = (totalseeing + 1e-6)*pow(1/cos(zenith), 0.6)*aph->wavelengthFactor;
norm = sqrt(0.0229*pow(0.98*(1e-4*0.5)/(seeing*ARCSEC),-5./3.))*3.25e8;
paddingfactor = 1;
rinner = surface.innerRadius[0]/paddingfactor;
router = surface.outerRadius[0]/paddingfactor;
}

for (i = 0; i < SCREEN_SIZE; i++) {
for (j = 0; j < SCREEN_SIZE; j++) {
radius = sqrt((i - SCREEN_SIZE/2 + 0.5)*(i - SCREEN_SIZE/2 + 0.5)+
(j - SCREEN_SIZE/2 + 0.5)*(j - SCREEN_SIZE/2 + 0.5))*screen.fine_sizeperpixel;
dx = (i - SCREEN_SIZE/2 + 0.5)*screen.fine_sizeperpixel*cos(aph->shiftedAngle) +
(j - SCREEN_SIZE/2 + 0.5)*screen.fine_sizeperpixel*sin(aph->shiftedAngle);
dy = -(i - SCREEN_SIZE/2 + 0.5)*screen.fine_sizeperpixel*sin(aph->shiftedAngle) +
(j - SCREEN_SIZE/2 + 0.5)*screen.fine_sizeperpixel*cos(aph->shiftedAngle);
if (radius > rinner && radius < router) {
screen.inscreen[SCREEN_SIZE*i + j][0] = cos(screen.phasescreen[i*SCREEN_SIZE + j]*norm);
screen.inscreen[SCREEN_SIZE*i + j][1] = sin(screen.phasescreen[i*SCREEN_SIZE + j]*norm);
} else {
screen.inscreen[SCREEN_SIZE*i + j][0] = 0.0;
screen.inscreen[SCREEN_SIZE*i + j][1] = 0.0;
}
}
}
```

When the atmospheric turbulence is off, the `paddingFactor` variable is set to an appropriate value. When the atmospheric turbulence is one, `paddingFactor` is set to 1 to keep the units consistent with the atmosphere component of the phase screen. But I must now compensate for the smaller aperture by again dividing by γ in the second kick function when the atmospheric turbulence is off. The modified second kick function then becomes:

```
int Image::secondKick(Vector *largeAngle, Photon *aph) {

long index;
double r, phi;

r = random.uniform();
find(screen.hffunc, 10000, r, &index);
if (atmospheremode < 2) {
int paddingfactor = 32;
r = ((double)(index))*0.5*1e-3/(surface.outerRadius[0]*paddingfactor)*aph->wavelength;
}
else
r = ((double)(index))*0.5*1e-3/(SCREEN_SIZE*screen.fine_sizeperpixel)*aph->wavelengthFactor;
phi = 2*M_PI*random.uniform();
largeAngle->x += r*cos(phi);
largeAngle->y += r*sin(phi);

return(0);

}
```

Notice that the phase screen is not zero padded when the atmospheric turbulence is on. In this case, the pupil diffraction part of the PSF will be significantly less accurate than the improved version with zero padding. However, in this case the atmospheric component of the diffraction is dominant, so it is not a significant issue. A proper fix for the intermediate regime, $\max\left[\frac{D}{r_0}, \frac{\alpha}{\lambda f}\right] \sim 1$, would require reconciling the differences required to obtain accurate Fourier transforms for both the atmosphere and the telescope, while implementing the correct wavelength scaling for each. To my knowledge, physics-based computational modeling of this intermediate diffraction regime has not been extensively studied.

Now, the improved algorithm can be tested in a simple diffraction-limited scenario.

4 Method

To test the newly improved diffraction algorithm in the diffraction-limited scenario, I invoke use of a perfect telescope comprised of a parabolic mirror and a detector located at its focal point. This way, it should be trivial to test the fidelity of the PSF generated by PhoSim. The optical prescription is defined by Table 1.

Table 1: General contents for a simple parabolic telescope using variable names consistent with this document.

type	R (mm)	dz (mm)	outer radius (mm)	inner radius (mm)	κ	coating file	medium file
mirror	R	0	$D/2$	0	-1	none	vacuum
det	0	$F = R/2$	10	0	0	none	vacuum

A parabolic telescope has the useful property that on-axis rays are focused perfectly at the focal point, which lies at a distance of $R/2$ in front of the vertex of the parabolic mirror. The parabolic telescope's instrument and site characteristic (ISC) files are under `data/generic` in the PhoSim repository. A CommandFile named `diffractiononly` with the following contents is used to turn off all physics (including the atmospheric turbulence) except for the raytracing and telescope pupil diffraction:

```
opticsonlymode 1
backgroundmode 0
diffractionmode 1
```

Because the characteristic size of aberrations for on-axis rays is zero ($\alpha = 0$) and the atmospheric turbulence is off, we satisfy Condition 6 and are firmly in the diffraction-limit.

In order to further simplify analysis, I create a spectral energy distribution (SED) file with a non-zero irradiance value at only one wavelength. For the example of 500 nm, the SED contains the following information:

```
499.9 0
500.0 1
500.1 0
```

In general, a good monochromatic SED file would look like:

```
x-0.1 0
x 1
x+0.1 0
```

where x is the wavelength of interest in nanometers.

Finally, a CatalogFile named `diffractiontest` is created with a source at right ascension and declination of $(0, 0)$ (on-axis rays) using the monochromatic SED described above. Tests can then be completed with the following command:

```
./phosim examples/diffractiontest -c examples/diffractiononly -i generic
```

5 Results

I compare the PhoSim PSF amplitude to the corresponding Airy diffraction function:

$$I(x) = I_o \left(\frac{2J_1(x)}{x} \right)^2 \tag{10}$$

where I_o is the maximum PSF intensity, J_1 is the Bessel function of the first kind of order one, and x is given as

$$x = \frac{2\pi D r}{\lambda L} \tag{11}$$

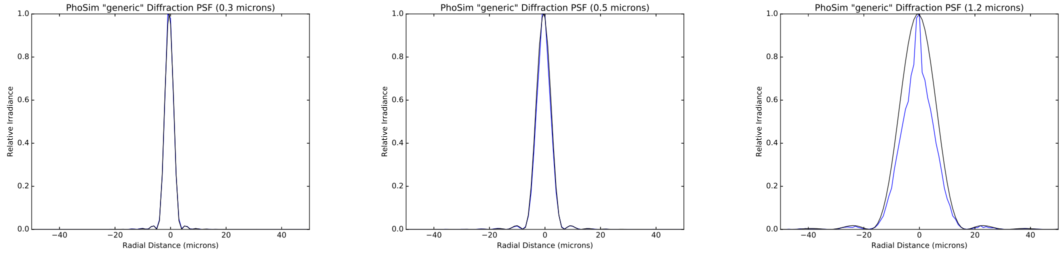
where D is the diameter of the primary mirror, r is the radial distance from the PSF center on the focal plane, and L is the optical path length. In this case L is equal to the focal length F to first order:

$$x \approx \frac{2\pi r}{\lambda f} \tag{12}$$

because the focal ratio $f = F/D \approx L/D$ (Hetch, 2016, p. 482).

Although the full morphology of the PSF at the focal point depends on the wavelength, focal ratio, and the pupil characteristics, the expected distribution should depend on the radial $1.22\lambda f$ proportionality. The value of γ must be chosen which is sufficient for analysis at the scale of interest. For normal pixel sizes, small value of γ should suffice since the pixel size is usually engineered to be no larger than the Rayleigh Criterion, and the distribution will appear to be almost entirely within a single pixel. However, for sub-pixel analysis, larger values of γ should be used to get better accuracy. Figure 2 shows the distribution on the generic telescope with an LSST-like focal ratio of $f \approx 1.2$, a pixel size of 0.1 μm, and $\gamma = 64$ for varying wavelengths across the optical band.

The λf scaling performs as expected. Since the second kick angular displacement scaling depends on λf, and based on Figure 2, it can be concluded that the distribution is less accurate as λf increases. Although γ can be increased to gain more accuracy up to a point. But if γ is increased too much, the effective resolution of the pupil will decrease. It is up to the user to choose a value of γ that is sufficient enough for the size scale (pixel size) of interest ($\gamma = 8$ should suffice in most cases). The default value is set to $\gamma = 32$, which should



(a) $\lambda = 0.3$ μm (b) $\lambda = 0.5$ μm (c) $\lambda = 1.2$ μm

Figure 2: Diffraction-limited monochromatic PSF cross-sections varying λ with $\gamma = 64$ and $f \approx 1.2$, perfect optics, and detector physics off. The blue line is the measured amplitude, and the black is Equation 10.

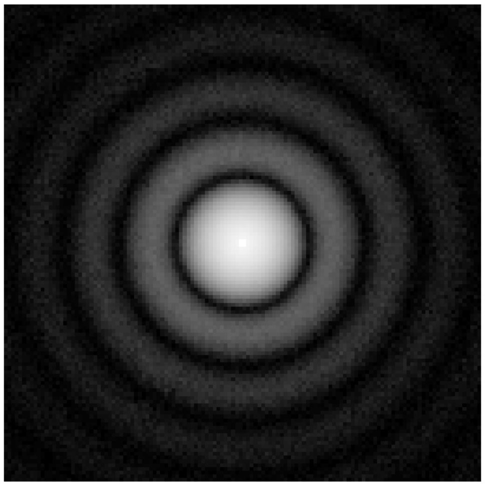


Figure 3: A high-fidelity $\gamma = 64$ Airy pattern on a PhoSim detector using the improved diffraction algorithm, shown here in log scale.

be more than sufficient for analysis done significantly below the size of an LSST pixel (10 μm). Figure 3 shows a diffraction profile with large zero padding.

There are two factors to keep in mind when determining the level of PSF accuracy that is needed in PhoSim:

1. Multiwavelength treatment blurs PSF (chromatic aberration).

2. Detector effects blur the PSF (charge diffusion).

3. Time-averaged PSF effects blur the PSF (mostly jitter).

4. Finite pixel size limits usefulness.

The first three effects blur the diffraction pattern, while a finite pixel size means PhoSim only needs to produce reasonable results at this size scale (normally no larger than the Rayleigh Criterion: $1.22\lambda f$). However, in highly unusual cases with large focal ratios or wavelengths, the PSF may be systematically smaller if γ is not sufficiently large. As new telescopes are added to PhoSim, more work may be necessary to decide on a value for γ because it depends on the details of the optical system (pixel size, pupil geometry, λ_{max}, and f).

6 FITS Screen Input

The standard input parameters in PhoSim are sufficient for circular-aperture telescopes with simple spider obstructions. However, for more complex telescope pupils, such as a multi-segmented primary mirror or a non-trivial obstruction, I add the option of using a previously-generated FITS image as the telescope geometry component of the phase screen. This way, the pupil geometry can be specified by a FITS file which is discrete Fourier transformed directly. When the `pupilscreenmode 1` command is included in the CommandFile and the diffraction is on, PhoSim looks for a file named `pupilscreen.fits` in the ISC directory. The pixel scale and zero padding factor γ must be specified as header keywords.

To test the diffraction with a complicated pupil, I use the James Webb Space Telescope "Revision V" screen (Figure 4), provided as part of the WebbPSF package (Perren, 2012).

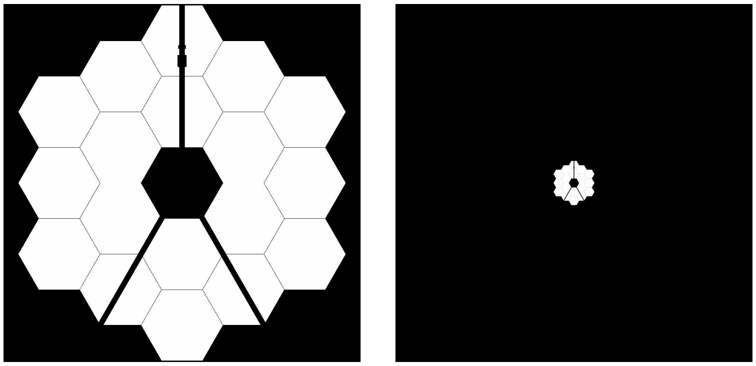


Figure 4: Image of the pupil diffraction screen for JWST with zero padding (left), and padded at $\gamma = 8$ (right).

The final result is shown in Figure 5.

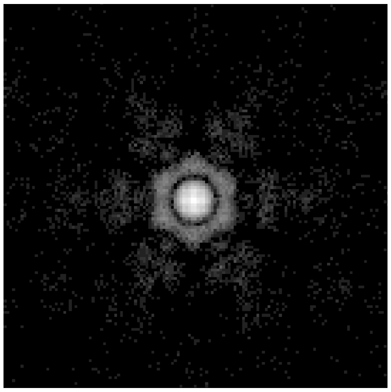


Figure 5: A PSF with the diffraction profile from the JWST telescope geometry, shown here in log scale.

7 Conclusions

A full fix of the diffraction algorithm including the region in-between a diffraction-limited and non diffraction-limited system is non-trivial, and would need to convolve both methods at a sampling of different wavelengths while considering the differing unit scales for both approaches. The new diffraction algorithm should be sufficient for most uses. However, if the goal is to do detailed analysis of a diffraction-limited PSF on sub-pixel scales, one must be very careful that the numerical accuracy is sufficient. To relieve these concerns, a more appropriate algorithm may be implemented in the future.

For diffraction-limited systems, the modified algorithm described in this work seems to produce accurate results. Although largely irrelevant for the typical PhoSim use case (e.g., non diffraction-limited telescope, pixel-sized detector sampling, detector physics on), this work is essential for modeling diffraction-limited telescopes or when analyzing the PSF on small scales with just the telescope optics. The production of a realistic diffraction pattern is essential for space-based telescopes, should one be implemented in the future.

8 Acknowledgements

Many thanks to Professor John R. Peterson at Purdue University for supervising this work, and for his many helpful conversations.

References

Fried, D. L. 1965, JOSA, 55, 1427

Hetch, E. 2016, Optics (5th Edition) (New York, NY: Pearson)

Lyons, R. G. 2010, Understanding Digital Signal Processing (3rd Edition) (Upper Saddle River, NJ: Prentice Hall)

Perren, M. D. et al. 2012, Proc. SPIE, 8442, 11

Peterson, J. R. et al. 2015, Astrophys. J, 218, 24

Peterson, J. R. et al. 2013, “PhoSim Reference Document”

Peterson, J. R. et al 2012, PhoSim Internal Document (PIN-14)

Soummer, et al. 2007, Optics Express, 15, 24