Instrument Science Report ACS 2023-06

The ACS/WFC Focus-Diverse ePSF Webtool

G. S. Anand, N. A. Grogin, J. Anderson, Y. Cohen, A. Bellini

November 10, 2023

Abstract

The effects of telescope breathing cause the focus level of HST to change as a function of time during each orbit. This results in modest, but non-negligible, changes to the point-spread functions (PSFs) of stars. For optimal PSF fitting photometry, the library PSFs must be adjusted to account for the changing focus levels. Anderson & Bedin (2017) used a phylogram-based technique to show that it is possible to accurately model focus-related variations in the PSF and thus determine the focus level for a given exposure with measurements of just a handful of bright, isolated stars in the individual WFC3/UVIS images. Bellini et al. (2018) extended this analysis to ACS/WFC, with similar conclusions. Using the results of the Bellini et al. (2018) analysis and subsequent extension to the 11 most popular ACS/WFC filters, we have developed a webtool and Python API to deliver focus-diverse effective PSFs (ePSFs) to the community. The webtool is available for public access at `acspsf.stsci.edu`.

---

1 Introduction

It is now well-established that the focus level of HST instruments varies due to an effect known as "breathing" (Bely et al., 1996; Sahu et al., 2007; di Nino et al., 2008), which results from uneven heating of HST's Optical Telescope Assembly (OTA). This effect is induced mainly by HST's position relative to the Sun. The effect on the observed imaging point-spread functions (PSFs) is up to a ∼5% change in the fractional flux delivered to the central pixel (Bellini et al., 2018), along with associated changes to the surrounding pixels.

To perform more accurate photometry, users can perturb standard library PSFs (e.g. those provided in Anderson 2022) to better fit the observed PSFs. This is typically done by analyzing many bright, isolated stars in each image to then derive adjustments to the base library PSFs (e.g. as done by the PSF photometry package DOLPHOT, Dolphin 2000). However, such measurements are most accurate when there are many-tens to hundreds of bright and isolated stars available, which is not the case with many HST images. In an effort to improve the availability of accurate PSF models for WFC3/UVIS, Anderson & Bedin (2017) developed a technique based on the usage of phylograms (from evolutionary biology). They found that most PSFs lie along a single one-dimensional locus parameterized by the underlying focus of the telescope. Importantly, they were able to retrieve the underlying focus level with only ∼5 medium-bright stars in the image, allowing for much more accurate PSF determinations in images lacking the many bright stars required for typical PSF perturbation techniques. The methods of Anderson & Bedin (2017) were developed on WFC3/UVIS images, and Bellini et al. (2018) applied these methods to explore focus-related PSF variations in ACS/WFC, beginning with the F606W and F814W filters. They found more complex structures in the underlying phylograms, albeit this fact does not significantly affect the usability of this technique. Their analysis was also conducted in two distinct time domains, separating out the pre- and post-Servicing Mission 4 (SM4) images as the underlying PSFs changed after the servicing mission. The work presented in Bellini et al. (2018) was later expanded to include the 11 most used ACS/WFC filters. The Bellini et al. (2018) report ends with a concluding statement that states that future plans include the development of an interactive webtool that would provide users effective PSFs (ePSFs) that are tailored to the focus levels of any individual `flc` file. In this report, we describe precisely such a tool and make it available to the broader HST community.

2 Architecture and Usage

2.1 Architecture

Our webtool[1] has been designed and deployed through the usage of Amazon Web Services (AWS). A basic architecture diagram can be seen in Figure 1. When a user clicks onto our website, they are presented with a simple webpage whose underlying files are located within a private Simple Storage Service (S3) bucket, and hosted onto the internet via a CloudFront distribution. Upon entering an image rootname into the search box and pressing the "Retrieve ePSFs" button, the page uses a POST request to activate an application program-

---

[1] Available at `acspsf.stsci.edu`.

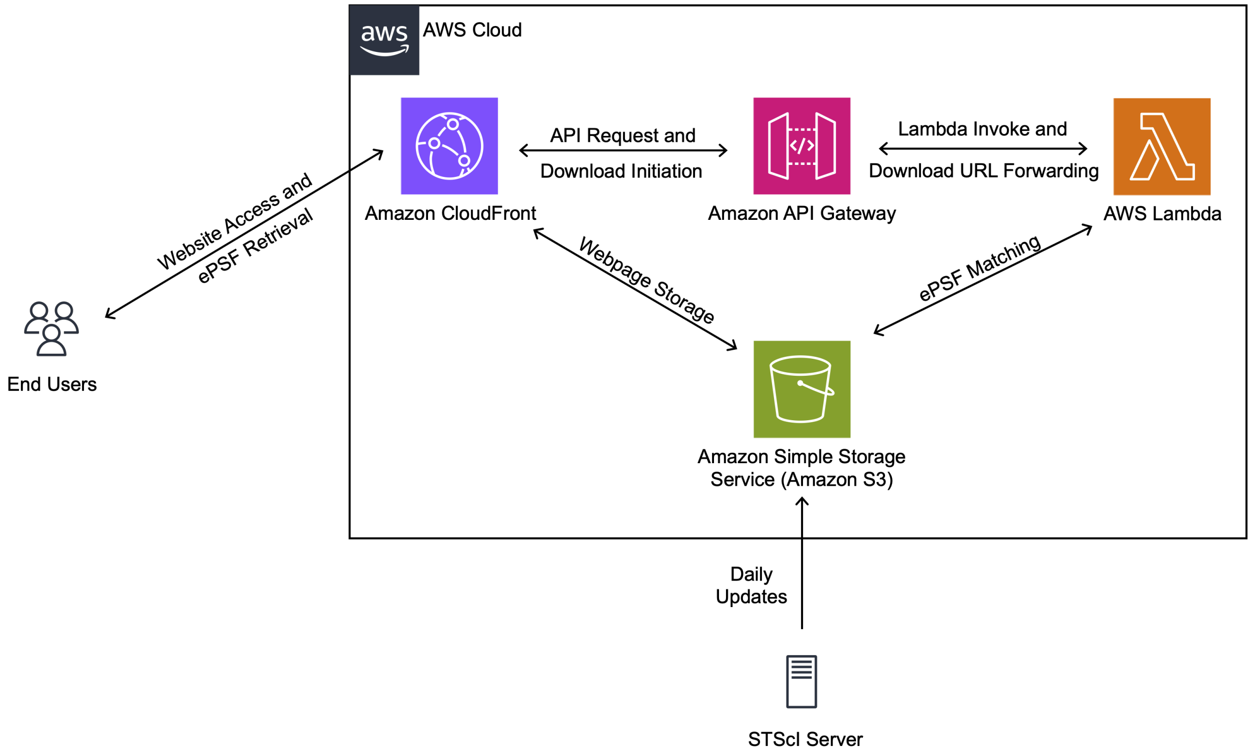
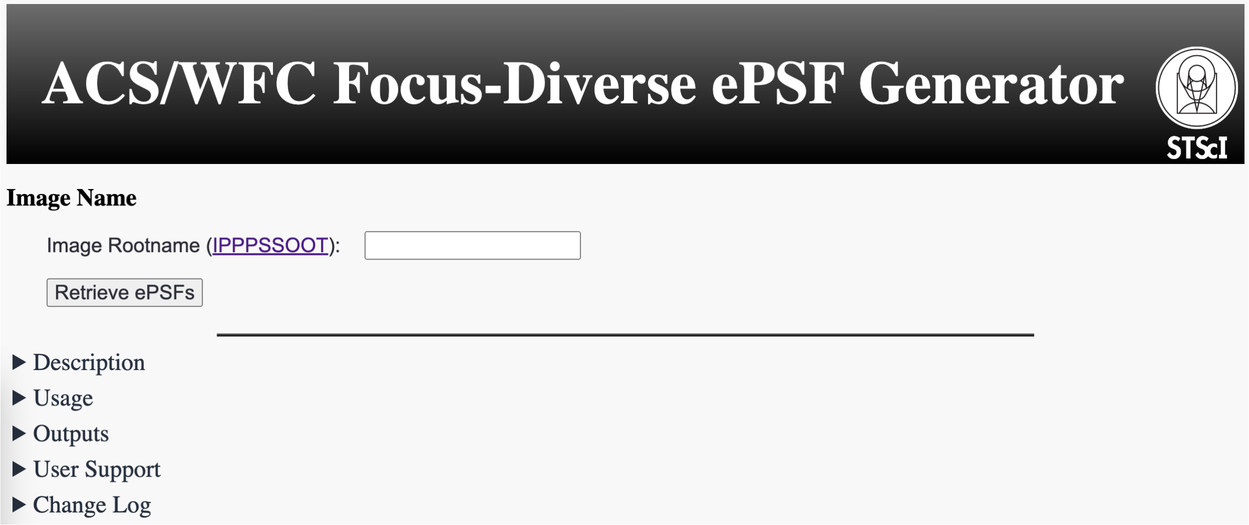


Figure 1: An AWS architecture diagram outlining the services used by our webtool. See the main text for an expanded description of each step.

ming interface (API) through Amazon API Gateway and forwards the image rootname to a serverless Lambda function, which performs the ePSF matching through a simple lookup table that is also located on S3. The Lambda function then generates a presigned URL for a direct download to the appropriate ePSF array, which also has the input rootname preprended for easier bookkeeping. This presigned URL (valid for 60 minutes) is then returned to the webpage, and a direct download of the appropriate ePSF takes place.

Note that we have not yet discussed the actual routine that performs the matching of an individual HST image to its corresponding focus level within our AWS architecture. This is because the determination of the ePSF focus levels happens on a local STScI server. Upon initial development of this webtool, we performed a matching of all available `flc` images via a separate Fortran routine that examines the image and looks for bright, isolated stars to determine the best-matching focus level (Bellini et al., 2018). For all future incoming data, we have set up a routine that scans MAST daily for new ACS/WFC data, runs the focus-determining routine, and appends the image rootname and focus level to the lookup table, which is then sent up to S3 via the AWS command line interface.

Our choice to run the focus-determining routine locally is based on two key factors. The first is that for images where there are very many bright stars in the field, the routine can take several minutes. Waiting several minutes for an individual request does not present a good experience for the user of a webtool. The second factor is that we are providing this service for all incoming images, including those which have a proprietary period. For security reasons, we prefer to keep the data on our local STScI machine to run the focus-



ACS/WFC Focus-Diverse ePSF Generator

Image Name

Image Rootname (IPPPSSOOT):

Figure 2: A screenshot of our new ACS/WFC ePSF webtool. Relevant details are given in each of the individual drop-down menus.

determining routine, and not within an S3 bucket that could open up another avenue for security concerns.

As per our institutional requirements, the entire AWS project is located in Infrastructure as Code (IaC) format on the STScI private GitLab page, and can be deployed via a shell script from the command line.

2.2 Usage

2.2.1 Webtool

Upon arriving at the webtool, the end-user is greeted with a simple webpage with a single box for input of the image rootname, as well as various pull-down menus describing the tool. The input format for the "Image Name" box must be in the HST IPPPSSOOT format (e.g. jds408jsq). The tool does not accept inputs in the form of association IDs or product names (e.g. jds408011), and does not support observations taken with subarrays. Upon inputting the rootname and pressing "Retrieve ePSFs", the browser will begin a download of the FITS file containing the PSFs that are best matched to the focus level of the input image, with the determined focus level given in the name of the file provided.

2.2.2 Python API

While the webtool interface is simple in nature, it may present some difficulty for users that need to retrieve focus-diverse ePSFs for many distinct images. To this end, we have developed a Python module (`focus_diverse_epsfs`) for `acstools`[2] that allows one to directly access the AWS API without needing to interface with the website itself. A companion Jupyter notebook[3] provides examples of ePSF retrieval using this Python module, including one case

---

[2] `https://github.com/spacetelescope/acstools`

[3] **The Jupyter notebook will be linked shortly.**

where a text file of input rootnames is provided, or one where we use `astroquery` to grab the input rootnames for an entire HST program via MAST directly. A singular example from this notebook is shown in Appendix A, however we refer the reader to the notebook for the full suite of examples. The outputs from the Python API are identical to those from the Webtool (including error handling via output of text files describing the errors).

3 Retrieved ePSFs

Until now, we have referred to the products from our new webtool as ePSFs without providing an explicit definition. As per Anderson & King (2000), the image of a star at the focal plane can be referred to as an instrumental PSF (iPSF). This iPSF is never observed because one always has a detector with a set pixel grid involved in the measurement. The relevant quantity for us is the ePSF, which gives the fraction of stellar flux falling into each pixel, taking into account the integration of the surface brightness across the face of the pixel, any pixel-response-function variation, and any offsets between the centers of the pixels and the central location of the star (which need not be directly in the middle of the central pixel).

Whether retrieved through the webtool or the API directly, the downloaded products are identical FITS files, with dimensions of 101×101×90. The first two dimensions refer to each individual ePSF, whereas the index (1-90) refers to an ePSF at a particular location on the WFC detector, as can be seen in Figure 3. The individual ePSFs are valid for the coordinate array shown in Figure 1 of Bellini et al. (2018), where the x and y coordinates are x = [0, 512, 1024, 1536, 2168, 2800, 3192, 3584, 4096] and y = [0, 512, 1024, 1536, 2048]. Note that while the ePSF definitions begin at (0,0), this position is technically off of the detector. The pixel at the lower left edge of the detector is (1,1), and this position refers to the center of the pixel. We provide an interpolation routine to output the ePSF at any given set of (x,y) coordinates (see §3.1).

The ePSFs are supersampled by a factor of 4. These ePSF arrays provided through our tools are also interpolated in focus-space, in steps of 0.1 focus levels. While Bellini et al. (2018) only described the creation of these ePSFs for the F606W and F814W filters, this work was later expanded to include the 11 most common ACS/WFC filters (a minimum number of images are required to generate useful focus levels for each filter). The number of focus levels for each filter varies (and may not be same for the pre- and post-SM4 versions), and is shown in Table 1.

It was mentioned earlier that the issue with the standard technique of perturbing library ePSFs for the purposes of better matching the observed ePSFs (that differ due to changes in focus) is difficult for images without large amounts of bright, isolated stars. Many HST users do not have this luxury in their images. A key benefit of the phylogram techniques is the requirement of just a few such stars. Indeed, we find successful focus level matches for over 96% of the `flc` images within the ACS/WFC archives. This number is likely an underestimate of the success rate, as some of the failures in our database are due to cases where there is significant drifting due to issues with guide star acquisitions. In the uncommon cases where a focus-level match was not found for a particular `flc` image, the download will instead be a text file stating that there were not enough stars for performing the focus-level matching. For cases where the image rootname does not exist in our database, the download

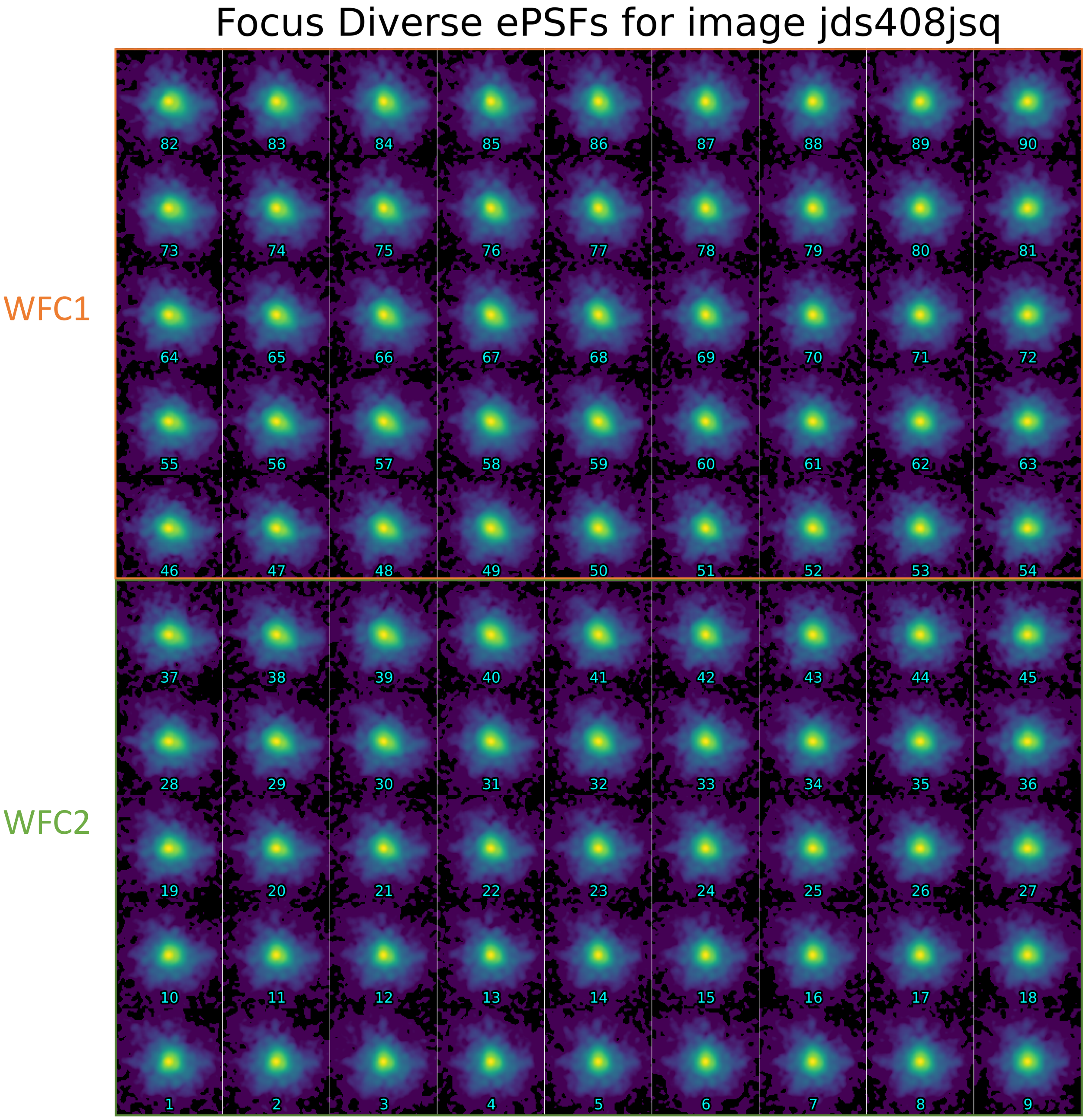


Figure 3: An example set of focus-diverse ePSFs built for image jds408jsq_flc.fits from GO-15445 (PI W. Keel). The individual ePSFs are valid for the coordinate array shown in Figure 1 of Bellini et al. (2018). The numbers (1−90) correspond to the index of each individual ePSF (shown with a log stretch) within the multi-dimensional FITS file. We also provide an interpolation routine to provide an ePSF at any particular (x,y) position.

will be a text file stating such.

Filter	# (pre-SM4)	# (post-SM4)
F435W	10	8
F475W	12	11
F502N	5	5
F555W	9	6
F606W	18	20
F625W	5	3
F658N	7	7
F660N	3	3
F775W	14	6
F814W	13	9
F850LP	5	5

Table 1: A table that lists the number of individual focus levels for each of the 11 ACS/WFC filters for which focus-diverse PSFs are available through our webtool, separated by epoch (pre- and post-SM4).

3.1 Further Spatial Interpolations

While the 9×10 grid of ePSFs we provide may be sufficient for some use cases, some users may wish to further interpolate to a particular point on the detector grid. The `focus_diverse_epsfs` module in `acstools` also contains a function (`interp_epsf`) that allows one to take the output FITS file from either our webtool or Python API and further interpolate it to any given position on the ACS/WFC detector. The function uses bilinear interpolation between the nearest four ePSFs to output a new ePSF at any given set of integer coordinates. Further, the function also supports the output of ePSFs in pixel space (as opposed to their native 4× supersampling), as well as sub-pixel shifts via bicubic interpolation. Details on usage of this function are provided in the same companion Jupyter notebook that describes the AWS API. We show brief code snippets and outputs in Appendix A, however we refer the reader to the full notebook for a complete description.

4 Summary

We present a webtool and API that allows end-users to retrieve ePSFs for ACS/WFC that are tailored to the particular focus level at the level of individual images using the phylogram technique introduced in Anderson & Bedin (2017). The webtool can be found at `ACSePSF.stsci.edu`, whereas the Python API can be accessed via the `acstools` software package. Use of these ePSFs have been shown to provide photometry and astrometry that is of a higher quality than that obtained with the standard library ePSFs (Bellini et al., 2018). The webtool and API are both built with AWS infrastructure, while the actual focus-matching is performed daily on an STScI server (mainly due to security concerns with the handling of proprietary data). We provide example Jupyter notebooks for the usage of both the Python API, as well as for further spatial interpolation of the ePSF arrays to a precise position within each image. We encourage feedback from the community on this webtool through the HST Help Desk.

5 Acknowledgements

We thank David Stark and Jenna Ryon for useful comments on an earlier version of this report. We also thank Anupinder Rai for assistance with the deployment of this AWS application.

References

Anderson, J. 2022, One-Pass HST Photometry with hst1pass, Instrument Science Report ACS 2022-02

Anderson, J., & Bedin, L. R. 2017, MNRAS, 470, 948

Anderson, J., & King, I. R. 2000, 112, 1360

Bellini, A., Anderson, J., & Grogin, N. A. 2018, Focus-diverse, empirical PSF models for the ACS/WFC, Instrument Science Report ACS 2018-8

Bely, A., Anderson, J., & Grogin, N. A. 1996, Orbital Focus Variations in the Hubble Space Telescope

di Nino, D., Makidon, R. B., Lallo, M., et al. 2008, HST Focus Variations with Temperature, Instrument Science Report ACS 2008-03, 29 pages

Dolphin, A. E. 2000, PASP, 112, 1383

Sahu, K. C., Lallo, M., & Makidon, R. 2007, ACS PSF Variations with Temperatures, Instrument Science Report ACS 2007-12, 15 pages

Appendix A

Here we show code snippets from an abridged version of the corresponding Jupyter notebook (**to be provided shortly**). Please see the notebook for the full code and context.

5.1 ePSF Retrieval

Here is a simple example of how to retrieve a focus-diverse ePSF array corresponding to a single input rootname.

```python
import matplotlib.pyplot as plt
from astropy.io import fits
from focus_diverse_epsfs import interp_epsf, psf_retriever

# provide an existing file location for download
download_location  = "/Users/ganand/Desktop/ePSFtool/batch-focus-work/single/"

# call the psf_retriever function with observation rootname
retrieved_download = psf_retriever("jds408jsq", download_location)

# open the file with astropy.io and grab the image data
retrieved_filepath = (retrieved_download)
hdu = fits.open(retrieved_filepath)
ePSFs = fits.getdata(retrieved_filepath, ext=0)
```

Listing 1: Code to grab the corresponding focus-diverse ePSF array for a singular image.

More complex examples of ePSF file retrievals (including batch retrievals) are available in the Jupyter notebook.

5.2 ePSF Interpolation

Here we will show examples of how to use our interpolation routines. First we begin by obtaining the ePSF valid at a specified integer (x,y) position.

```
# grab and plot interpolated ePSF in supersampled space
x = 2000
y = 2000
chip = "WFC1"

fig = plt.figure()
ax  = fig.add_subplot(111)

P = interp_epsf(ePSFs, x, y, chip)

im = ax.imshow(P, cmap = cmap, norm = colors.LogNorm(vmin = 1e-4), origin = "lower")

ax.set_title("jds408jsq at x,y = (2000,2000) on WFC1")
fig.colorbar(im)
```

Listing 2: Code to obtain the ePSF at a specified integer position.

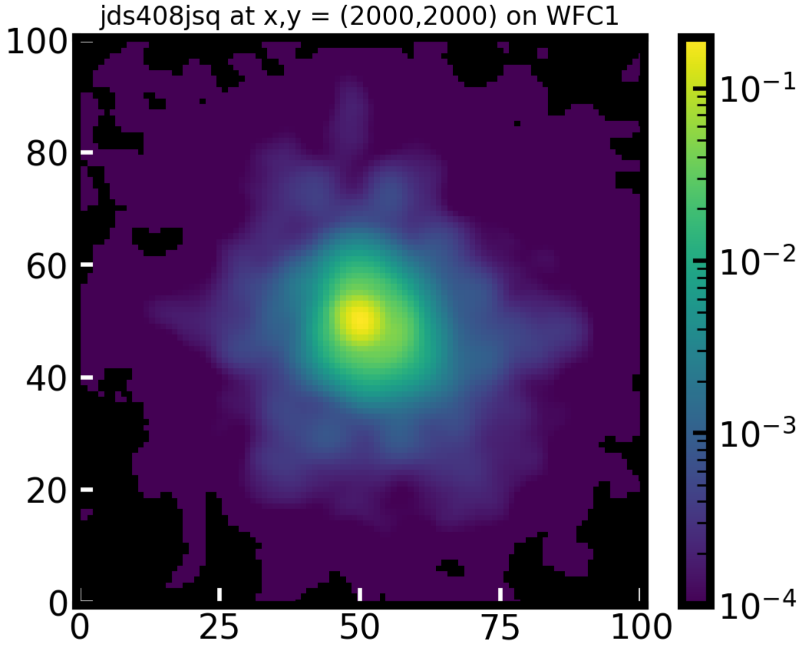


Figure 4: The focus-diverse ePSF for jds408jsq valid at x,y = (2000,4048).

Next, we will show the same ePSF as above, except outputting it in detector space.

```
# grab and plot interpolated ePSF in detector space
fig = plt.figure()
ax  = fig.add_subplot(111)

P = interp_epsf(ePSFs, x, y, chip,
                pixel_space = True)
im = plt.imshow(P, cmap = cmap, norm = colors.LogNorm(vmin = 1e-4), origin
    = "lower")

ax.set_title("(2000,2000) on WFC1 (Detector Space)")
fig.colorbar(im)
```

Listing 3: Code to obtain the ePSF at a specified integer position and output the ePSF in detector space.

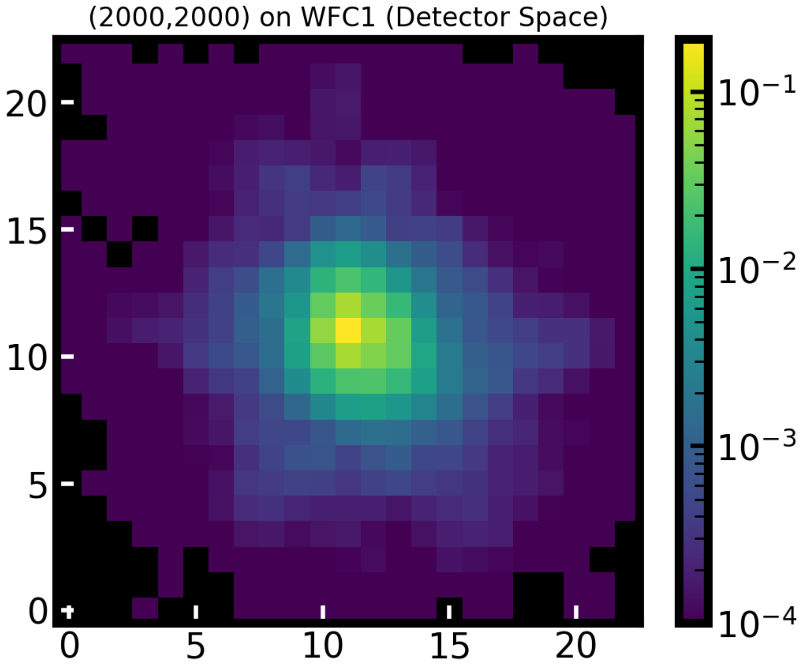


Figure 5: The focus-diverse ePSF for jds408jsq valid at x,y = (2000,4048), now output in detector space.

Lastly, we show how to obtain the previous ePSF, with additional subpixel phase offsets.

```python
# grab and plot interpolated ePSF in detector space with specified sub-pixel shifts
fig = plt.figure()
ax  = fig.add_subplot(111)

P = interp_epsf(ePSFs, x, y, chip,
                pixel_space = True,
                subpixel_x = 0.77, subpixel_y = 0.33)

im = plt.imshow(P, cmap = cmap, norm = colors.LogNorm(vmin = 1e-4), origin = "lower")
ax.set_title("(2000.77,2000.33) on WFC1 (Detector Space)")
fig.colorbar(im)
```

Listing 4: Code to obtain the ePSF at a specified subpixel phase position.

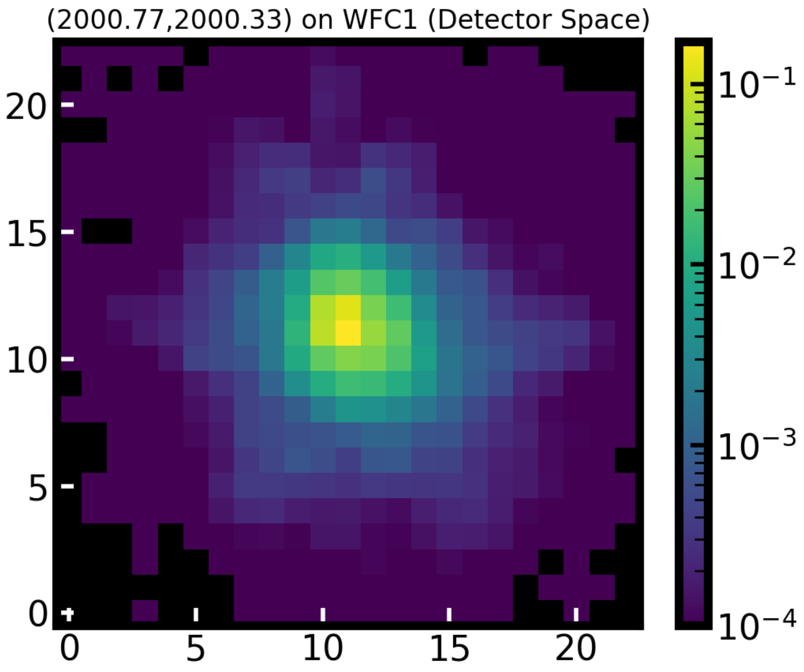


Figure 6: The focus-diverse ePSF for jds408jsq valid at x,y = (2000,4048), with additional subpixel phase shifts of x,y = (0.77, 0.33).