

# Fourier Optics Classes 1-8: The Basics

Anand Sivaramakrishnan, STScI, 2018 Oct - Jan 2019

## Prerequisites:

astroconda with python 3 on your laptop

matrixDFT from <https://github.com/mperrin/poppy/blob/master/poppy/matrixDFT.py>  
(there is a maintained version inside STScI, but we are only using this one routine from poppy. Get it any way that's convenient.)

Get the python sources `utils.py` and `telclass.py`.

Basic familiarity with complex numbers

Familiarity with DS9, image display stretches (linear, log,...) and matplotlib's `imshow`

Time: 8 hours per week: preparatory work prior to the weekly class, and, post-class, for expanding and writing up work.

## Course organization

Pre-class work will get you acquainted with the ideas/questions and try out some exercises. It is not expected that you will "finish" the problems set for this prep work, but the prep work is essential for the class. At the start of the class you will all discuss this prep work with each other, and to air your confusion (or clarity) on the concepts and details. This is a major vehicle of learning in the class.

After the in-class discussion I will go over points that need clarification or more in-depth understanding. That, and setting the stage for the next pre-class prep work will take the remaining part of the class.

After class you should finish the exercises associated with the prep work (e.g. in an annotated Jupyter notebook). This notebook will serve as your course notes, and hopefully a valuable reference for you and a lasting legacy of the course.

## Class Schedule

| Date   | Class # | Topics   |
|--------|---------|--|
| Oct 24 | #1      | Intro, class software, EM waves, scalar field, far field, the Fraunhofer approximation behind Fourier optics. The Fourier transform definition.  |
| Oct 31 | #2      | The monochromatic plane wave, the single photon probability density function meaning of the PSF. Limits to discrete Fourier transforms: effects of sampling, and finite information input. Fourier transform properties (the main theorems), applied to telescopes/imaging (tilts, shifts). Simulating the image plane of a given telescope pupil. |
| Nov 7  | #3      | The Lyot coronagraph.  |
| Nov 14 | #4      | Aberrations in Lyot coronagraphs.  |
| Nov 21 | -       |  |
| Nov 28 | #5      | Aberrations in the BLC, Spatially-filtered wavefront sensing   |
| Dec 5  | #6      | Aberrations, BLC   |
| Jan 16 | #7      | Wavefront sensing: Gerchberg Saxton in-focus phase retrieval basics  |
| Jan 24 | #8      | Wavefront sensing: Misell-Gerchberg Saxton focus-diverse phase retrieval basics  |

## Book, articles, and links

1. The best textbook for the Fourier part is still R.N. Bracewell, "**The Fourier Transform and Its Applications**" (Third Edition, McGraw-Hill, 2000). This book is unfortunately out of print and has been so for some time. The remaining stock is expensive. The **pictorial dictionary of Fourier transforms** is on pp. 573-591.
2. "Introduction to Fourier Optics" by J. W. Goodman, (McGraw-Hill, second edition, 1996 (or any other edition))
3. "Principles of Optics" M. Born & E. Wolf, 1975 (Pergamon). A new edition of this ancient classic was announced a few years ago. Even the older editions have all the basic

material you are likely to need. This book is the bible for physical optics. I use the 7th edition.

4. ["Interferometry and Synthesis in Radio Astronomy"](#) A.R. Thompson, J.M. Moran, & G.W. Swenson, (Springer, 2017). This is the bible for the subject as applied to radio astronomy. PDF downloadable via the link above.
5. [Fourier transforms page at NRAO](#) From about 2/3 of the way down and on to the end this fairly short page the basic Fourier transform theorems are laid out well.

## Class 1: Theoretical basis, numerical Fourier machinery

This is a hands-on course, but you should have some idea of the theoretical basis for numerical manipulations are, in case you need to delve deeper in the future. We will go quickly over the basics, and outline how we get to the stage of calculating the response of many optical systems to a point source or a sky scene.

We will cover the mathematical definition of the Fourier transform, and a couple of its fundamental properties.

### Theoretical work

Look up **Fourier transforms** and **Fourier optics**, and [re]familiarize yourself with the key concepts/phrases.

**The 1-dimensional Fourier transform.** What sort of function (real, complex,. ...) is the function (e.g.  $f(x)$ ) and its transform ( $F(k)$ )? Are the domains of  $f$  and  $F$  (the spaces that  $x$  or  $k$  live in) real or complex? Are  $f$  and  $F$ , deep down, different descriptions of the same information?

Does the theoretical Fourier transform of a function lose information that was present in the original function?

**Incoherent** vs. **coherent** emission of light from distinct points an **extended object**. Does an atom on one side of a star emit spectral line radiation in phase (coherent with) another atom of the same element on the other side of the star, emitting light in the same spectral line? Do then two atoms communicate with each other to collaborate on releasing radiation coherent with each other? How is this different from a gas in a laser cavity?

**Diffraction limit** or **[angular] resolution** of a telescope,  $\lambda/D$

The Fraunhofer or **far field** situation, and the **Fresnel length** that governs when the far field limit calculations are valid. Fresnel length  $\sim D^2 / \lambda$ . Parse out the meaning of this in terms of the angular resolution.

Describing light numerically/mathematically: a plane wave of monochromatic light (in a homogenous or non-dispersive medium or vacuum) is a propagating oscillation of the electric and magnetic fields. In  $(x,y,z)$  physical space  $z$  is the direction of propagation.  $(x,y)$  is the plane

transverse to the propagation direction. The wave can be expressed as the real part of a complex number that has a (real) amplitude  $A(x,y)$  and a unit-strength "phasor"  $\exp(j(kz - \omega t + \phi))$ .

We average over many periods of the waves' oscillation in a typical optical or IR measurement. We also "follow a wave's crest", and ignore the  $(kz - \omega t)$  in the exponent. One resource is [ApJ vol. 552 pp.397-408, 2001 May 1](#)), Section 2.1 equation 1 and surrounding very short text.

What mathematical/numerical operation must you perform on a complex array describing the EM wave's "complex amplitude" to get a real array describing the **intensity** (brightness) that a CCD or IR array might detect?

The concept of the **pupil** (or aperture) plane of an optical system, and its illumination by **monochromatic light** from a distant star.

Describing the pupil quantitatively (eg. equation 2 of the above paper). If you create a numpy array that represents a circular mirror telescope without a secondary obstruction, what are the physical dimensions you assign (in your mind) to this in-memory array --- what physical thing does the numerical array span? What physical quantity is represented by a number (an element) in this numerical array?

## Pre-class 2 and class 2 work

1. Become well-acquainted with the **definition of the convolution** of two functions. How is it applicable to simulating the image of an astronomical scene?
2. Become familiar with the [Special functions](#) notations in the repo - we use these in a "cartoon representation" of Fourier theory.
3. Familiarize yourself with the [basic Fourier theorems](#) which is also in [the repo](#).
4. Consider what effect a **lateral shift of the pupil** has on its Fourier transform, i.e. the image plane **complex amplitude**, and what effect it has on the **image plane intensity**. Write down a simple 1-D theoretical Fourier argument to support your conclusions.
5. Again, on purely theoretical grounds, what can you do to the pupil complex amplitude to shift the image intensity array off the origin of the image plane?

Look at and run code/telclass.py

See how the `matrixDFT` object is instantiated, a pupil created, and an image complex amplitude ("field") and its intensity are calculated.

Look through the parameters of `exer1()` and its output in `datadir/c1/ex1*.fits`

Write up a short description of why the images appear as they do. Consider information content in the two domains, pupil and image.

Do the same with `exer2()`. How does this show the difference between a detector pixel and a pointwise value of a simulated array that is our image intensity?

After attempting or succeeding in answering questions 4 & 5 above, run `exer3()`, `exer4()`, and `exer5()` by uncommenting the call (if needed) to `class2()` in the last line of `telclass.py`. We will discuss `exer3()`, `exer4()`, and `exer5()` in class. I put png files of ds9 views of the output fits files in the repo also.

---

## Class 2 plan

- The monochromatic plane wave, the single photon probability density function meaning of the PSF.
- Limits to discrete Fourier transforms: effects of sampling, and finite information input.
- The basic Fourier theorems. The Sampling Theorem.
- Fourier transform properties applied to telescopes/imaging (tilts, shifts, equivalent widths, energy conservation).
- Simulating the image plane of a given telescope pupil. Pixel scale. Detector simulation.
- Polychromatic imaging.

## Class 3 plan

How did I take the photograph in `class3/Tilt_in_pupil.jpg`? It is unprocessed.

Understand the details of the Convolution and Similarity theorems in `class1/theorms6.pdf`. They will be utilized to understand coronagraphy. Look at e.g. the effects of convolving one function with another - top hat with a Gaussian curve that is much narrower than the top hat, for example - sketch the result.

Diffraction-limited stellar coronagraphy [SKMBK2001](#) especially Section 2.1, Fig. 1 and Fig 2.

Application: Earliest system sketch for GPI in Fig. 9.

## Class 4 prep and in-class work

Look at these questions individually, think about them and/or attempt them. It could be very helpful for you to get together in groups or subgroups before class 4 and discuss the details amongst yourselves. Some of you have more knowledge of coronagraphs than others, so please share your understanding.

Find the definition of a Band Limited function. This is used in the perfect [Band-Limited coronagraph](#).

Understand how the two graphical derivations in Fig. 2 of the paper [Aberration leak](#) work. One is a classical Lyot coronagraph, the other is a band-limited Lyot coronagraph.

- In particular, why is it that to first order, a tilt of the incoming wave (placing the star that needs to be suppressed off-center on the occulting focal plane mask) has no light leak through the coronagraph?
- Why is it that a quadratic phase error in pupil - a focus phase error - creates what is in effect a faint (and slightly wider) version of the original non-coronagraphic PSF in the final coronagraphic image plane? Look at Eq. 13, and Fig. 3 in the [Aberration leak](#) paper.

## Class 5 prep and in-class work

### Before class

Low pass and high pass filters in imaging:

- Pupil (stop) filters out **high angular frequency** information in sky image (a.k.a. resolution limit). A finite-sized pupil is a **low pass filter**. A point source is a delta function source in 2D sky angle space (eg RA, Dec). We only see lower angular frequency components through a finite diameter telescope.
- Field stop in an image plane filters out **high spatial frequency** information in pupil. Simulate Lyot plane intensity with a square hole in the preceding image plane in two cases: send through only three of the Airy rings, or 10 Airy rings). The field stop is a complement of a Lyot coronagraph's Focal Plane Mask, which lets all high spatial frequency pupil information through but removes the lowest frequencies.

Band-limited coronagraph aberrations:

- Create your own BLC, with a 250x250 pupil array, 100 pixel diameter (not radius) circular pupil inside it, an FPM that is  $1 - \text{jinc}^2$  where the  $\text{jinc}^2$  has its first zero at about the 5th Airy ring of the PSF in the image plane. Make sure the FPM array is zero transmission at its center! FT the  $\text{FPM} \times \text{imagefield}$  to the Lyot pupil plane array (another `mft.perform()` call), and look at the intensity in that Lyot plane **after** you multiply it with a sensible undersized Lyot stop, cutting out the light at the edges of the pupil.

- Also get the final coronagraphic image intensity after one last `mft.perform()`
- Next, put in a small phase aberration into the incoming wavefront at the entrance pupil. Put in a tilt, then a parabola centered in the pupil array (i.e. focus), astigmatism, coma, and see what happens in each separate case to the intensity of light in the Lyot pupil. Make your rms aberration over the active entrance pupil about 0.2 radians (so the first image's Strehl ratio is ~96%), to stay in the small aberration regime. I put a **Zernike fitting script** that generates a cube of zernikes in the repository - adjust the pupil size in that code to 100 across (match your active pupil), and you can use those slices for your phase. Check you have your desired phase rms (multiply them by the appropriate constant to ensure that).

Can you reproduce the sort of behavior you see in the [Aberration leak](#) paper?

This is your coronagraphic train - you can make it a Lyot coronagraph by using a 1 - top-hat FPM, with the occulting spot about 10 Airy rings in diameter. You'll have to play with your Lyot stop undersizing to get the Lyot stop effective in this case.

## In class

The Sampling theorem (Nyquist/Shannon, original stronger theorem by Laplace) ([class6/SamplingTheorem.pdf](#)).

Spatially-filtered wavefront sensing, speckle theory ([class6/rjaspeckle.pdf](#))

## Class 6

We covered BLC 2nd order, Strehl ratio, filter coherence length, something on Zernikes & effect on PSF, Maréchal approximation for SR, JWST Level 2 80% SR requirement in F200M NIRCAM,

Requests of WFS next few classes: Gerchberg Saxton, Missell Gerchberg Saxton

## Class 7 Gerchberg-Saxton in-focus phase retrieval

### Basic principles, the ideal case.

#### Background

I did not find any simple textbook explanations for the GS algorithm. The ones I saw in books describe the algorithm rather abstractly, very heavy on the mathematics. I wrote up a few latex pdf summary slides on the GS algorithm, **class7\_Gerschberg-Saxton\_Slides2019.pdf** in our repository. Please go over them before you do the pre-class exercises. I point to a single short section of **Thesis\_2012\_Osherovich.pdf** (in the class 7 materials) in these slides. Skim that briefly first (you might want to read that section after you have done the homework). After you read the slides you can look through the **GerchbergSaxton** class in my **gerchbergsaxton.py** code for the class.

I put a few other papers and files in the repository, but they are not mandatory reading.

The GS algorithm starts with image data, and a pupil geometry. We seed it with an initial guess at the phase over pupil. It then iterates between pupil and image planes, enforcing the known pupil, and the known image information each time.

The case I coded here is for monochromatic images, and better-than-Nyquist sampled images.

The bottom line for us is that the GS algorithm iterates to convergence. It converges to a pupil phase. It does not always converge to the correct answer. Reasons for this are beyond this course. It is important to start 'close enough' to the correct phase for GS to converge nicely to the correct solution.

### GS Exercises to do before class:

#### gerchbergsaxton.py file nomenclature explanation

| File name for Example 1 | Description                       |
|-------------------------|-----------------------------------|
| gs1__input_img.fits     | Image intensity file (unit power) |
| gs1__input_pup.fits     | Pupil constraint file             |
| gs1__input_truepha.fits | Phase used to create image        |
| gs1__errorint.fits      | Input image - final image         |
| gs1__errorpha.fits      | Input phase - measured phase      |
| gs1__imgint.fits        | Cube of image intensities         |
| gs1__pupint.fits        | Cube of pupil intensities         |
| gs1__puppha.fits        | Cube of pupil phases              |

Run the python script gerchbergsaxton.py from the code directory. It takes no input parameters. It creates examples of input data, tagged 0-3, and tries to retrieve the input phase. It uses a **main** driver to exercise the **GerchbergSaxton** class in the same file. You can use this object to explore other cases if you need to in the future. It is written for clarity rather than speed, but it runs quickly enough on the examples I created.

**The Gerchberg-Saxton.ipynb notebook also runs these examples** (thanks to Keira Brooks for putting the script into ipynb form! I modified the notebook slightly for more focussed displays with appropriate scales and limits as described below, before uploading it)

**Look at example 0 (which fails) briefly, but focus your attention on example 3 (which succeeds).**

The same phase aberration is used to create the images, with different pupils. The GS algorithm then tries to find the phase aberrations in each case. It fails in example 0 but succeeds in the



others. I change the pupil geometry, destroying the symmetry the **gs0\_\_input\_pup.fits** circular pupil.

**Open question:** Why is that so? It may be that if one tries several random pupil phase arrays (between  $\pm 1.5$  radians, i.e. about  $\pm \pi/2$ ) you might get some or many of them to converge to the correct phase. I have not investigated this particular case.

There's some degeneracy in a symmetric pupil that allows two phase maps to create the same image intensity pattern. Can you figure out what that is? Here's the [answer](#) (in the abstract, and early in the introduction. Can you demonstrate this to yourself mathematically? Also look at Figure 3 of this paper.)

The `gerchbergsaxton.py` script creates noiseless images on an exactly known pixel scale (two things that don't hold for real data!). We also know the exact pupil geometry (another idealization. Look at the **gsN\_\_input\*.fits** files: the pupil shapes, the image data, and the (supposedly unknown) pupil phase that we want to measure from the single in-focus image.

Examine the way phase slices in **gsN\_puppha.fits** evolve with iterations (I use a ds9 'movie' of the slices).

Look at the **gsN\_pupint.fits** intensity slices' evolution, log scale between approx.  $1e-9$  and  $0.1$  to see what's happening **outside the pupil support area**. Next, use a log scale between  $0.9$  and  $1.1$  to see how the **inside of the pupil support** behaves.

If you capture the output of the `gerchbergsaxton.py` script it gives you an iteration-by-iteration account of the way the phase updates evolve.

I stay well below phase-wrapping over  $2\pi$  (a 'technical' problem if it occurs), and avoid asking too much of the GS algorithm (my aberrations are within about quarter of a wave).

Further reading: John Krist on Hubble (focus on comments on non-parametric phase retrieval, GS especially, p4954 in **Krist & Burrows 1995 Applied Optics** in the class 7 materials directory).

[Fienup](#)'s papers for background and further details, practical cases, for when you might need them.

## **Class 8 MGS loop: uses same repo directory as class 7. Phase-diverse Phase Retrieval (PDPR).**

I added the MGS algorithm class notes to class 7's notes, because MGS is a natural extension of GS (so it did not make sense to set up the preamble in an identical way in a separate file).

**Your task before class is to flip through the MGS description in `class7_Gerchberg-Saxton_Slides2019.pdf`. Follow the step-by-step process of doing one full MGS loop through two defocussed images.** The new MGS class 8 material follows after last week's

class 7 material on GS. I will go over the details in these slides in class.

I did not write an MGS module. The step-by-step MGS recipe for phase retrieval using two defocussed images is in my addition to my class7 notes on GS. That MGS HOW-TO, along with my GS module as an iterative PR code example, should enable you to write an MGS module if you need to at some later date. I did create an "mgsdataset" in the code which is my way of organizing sets of input files for a full-up MGS phase retrieval. I did not run the data through an MGS PR scenario in this misellgerchbergsaxton.py code.

I felt it was more important to give you a clear how-to in my notes, and, in addition **develop your focus-diversity PR intuition** in class, since this is JWST's baseline fine phasing PR technique. With this in mind I copied my GS code to misellgerchbergsaxton.py, but only use the **create\_input\_datafiles\_bumps()** and **create\_input\_datafiles()** routines in it. The rest of the code can be ignored. You do not need to run the code or look at it: it is not a clean deliverable.

What is relevant is the two figures I made with the code illustrating the signal present in **pairs of equal-defocus images taken on either side of focus**. I will go through some of Dean & Burrows' figures in class after I explain the MGS theory while going through my step-by-step instructions for an MGS loop. The

**class8\_MGS\_DiversityChoice\_2003\_JOSA\_DeanBowers.pdf** paper is in the class materials for reference after the class, if you want to look deeper into these topics.

**1995\_KristAppliedOpticsHSTPhaseRetrieval.pdf** is the standard reference on HST phase retrieval, and contains a wealth of detail and distilled experience. If you are involved in PDPR and pupil modeling you should go over this article once you understand the basics.

**I will go over these in class:** The figures

**class8\_DeanBowers2003\_signal\_vs\_defocus\_ripple.png** and **class8\_DeanBowers2003\_signal\_vs\_defocus\_bump.png** show many panels. Each panel is made from a given wavefront aberration and a pair of defocussed images taken with the same amount of defocus, either side of focus. I flip and subtract one defocussed image from the other, just for a 'quicklook' at the sort of signal we get with that set-up. Dean and Bowers discuss the phase ripple case in their paper. I also did a 'segment piston' case labelled "bump" - what does a sticking-up segment look like at different defocusses??

In order to examine the information in these images I used a fixed small standard deviation of phase error across the pupil, so the Strehl ratio hit is the same for all images. I also scaled each difference image by the square of the defocus, a quick and dirty proxy for exposing until each image reached the same brightest pixel value. Therefore if a pane shows faint structure it does not contain much information on the associated aberration. If you see clear structure then those defocus values are useful for pinning down the aberration that I put into the wavefront. The quicklook processing I did here is akin to curvature wavefront sensing.

Finally, a high level very short overview of JWST WFS&C stages is given in **class8\_Acton-Scott\_WAVEFRONT SENSING AND CONTROL FOR THE JAMES WEBB SPACE**

**TELESCOPE.pdf.** After the course you might want to flip through it if you are not familiar with the stages of JWST's establishing an aligned co-phased telescope after launch.