Fourier Transforms Tutorial
Narrabri Synthesis Workshop
Sep 2001

In this tutorial I am hoping to show you four essential properties of the Fourier plane, as it relates to interferometry. The treatment is not exact, but tries to develop your inituitive sense of how Fourier transforms work.

Tools

You should have an X windows session open in front of you with some terminal windows, and a `kview` window, for viewing images.

To load an image into `kview` , you need to click on the **Set 1** button at top left. Then go to the **Intensity** menu and select the **Iscale for set 1** entry. This shows a histogram of the brightness values in the image. Click the **99%** near the top to set the greyscale display to reasonable values.

You can load a second image into **Set 2** and blink between them; choose **View− >Display Mode: blink**, then click on the **Blink** button.

Make sure you are in the `fourier` directory, e.g.

```
unix% pwd
/n/synwork/fourier/group1/
```

We're going to use the MIRIAD package to manipulate the images. I have written some helper programs that will simplify things. Run the `SETUP` program to make sure you have access to them.

```
unix% source SETUP
unix% which complex_multiply
/n/synwork/fourier/group1/bin/complex_multiply
```

You will need: `forward`, `backward`, `complex_multiply`, `complex_divide` and `delimage`. We will use one MIRIAD task, `maths`, directly.

MIRIAD-format images are unix directories containing the image, some data about the image (the header) and other stuff like a history of what has happened to the image.

Working with Fourier transform images

There are two parts to the fourier transform of an image, the *real* and *imaginary* parts. This form is most useful for mathematical manipulations.

But for thinking about interferometers and visualising Fourier transforms, it is often easier to look at the *amplitude* and *phase*. These can be derived from the real and imaginary parts, and *vice versa*. In most of the following, we will be looking at the amplitude images. The central parts correspond to the largest scales, finer details are stored in the pixels out toward the edge of the image.

The `forward` transform program creates all four images. These have the suffices `.rf` (real), `.if` (imaginary), `.af` (amplitude) and `.pf` (phase). You only need to give it a real image, it will assume the imaginary part to be zero.

The `backward` transform will look for the amplitude and phase parts, and if it can't find them will try to use the real and imaginary parts. It produces a real (`.rb`) and imaginary (`.ib`) images. In many cases the latter will be zero.

For all the fourier plane images the coordinates are u, v, with units of wavelength.

Procedure

You don't have to complete this today, just get as far as you can. And you can come back to it during the workshop.

1. **Spatial Frequencies**
   The first thing to learn is the relationship between angular scale in the image plane and location in the Fourier plane. Change to the `part1` directory, and forward transform an image.

   Look at the image, and then at the amplitude part (`.af`) with `kview` .

   Describe: the difference between the transforms of the two gaussians.
   Explain: why the amplitude is larger toward the centre.

2. **Why phase matters**
   Roughly speaking, phase in the Fourier plane relates to location in the image plane.

   Change to `part2`. forward-transform two of the gaussian images. Compare the amplitudes of the transform. Now compare the phase images.

   Now zero the phase and back-transform:

   ```
   unix% maths exp="<gauss_w10_o30.pf>*0" out=zerophase.pf
   unix% maths exp="<gauss_w10_o30.af>*1" out=zerophase.af
   unix% backward zerophase
   ```

   Now look at the `.rb` image. What has changed?

3. **Convolution and Deconvolution**
   These operations can be accomplished by multiplication and division in the Fourier plane.

   Change to the `part3` directory. Take one of the real-world images and multiply it by the fourier transform of a gaussian. Transform back.

   Now take the `smooth` image. Which of the gaussians was it convolved with?

   Lastly, take the interferometer beam image and convolve it with a real-world image.

4. **Incomplete measurement of the Fourier plane**
   This is almost always the situation in radio astronomy. The interferometer measures the Fourier transform of the sky above it, but only at particular points set by the baseline lengths. It also cannot measure the centre of the plane, the *zero-spacing*. Why not? How do you measure it?

   If the object you are imaging has a simple structure, incomplete information does not matter that much. But it does make it difficult to deconvolve by division in the fourier plane. Why?

   The statement above has a corollary: it possible to ignore bits of the Fourier plane. That means you can filter out, or *edit* out some aspects of the image you find distracting.

   Change to the `part4` directory. Try applying some masks to the transforms of some images, and back-transforming.

   ```
   unix% maths exp="<object.rf>*0+1" out=mask.xaxis
   unix% immask in=mask.xaxis region="abspix,box(1,120,256,136)"

   unix% maths exp="<object.rf>*<mask.xaxis>" out=object.m.rf
   ```

```
unix% maths exp="<object.if>*<mask.xaxis>" out=object.m.if
or
unix% maths exp="<object.af>*<mask.xaxis>" out=object.m.af
unix% maths exp="<object.pf>*1" out=object.m.pf
```

5. **Advanced concepts**

   Change to the `part5` directory. Try transforming an image containing two sources. Look at the amplitudes; what caused *that*? There are two things to explain.

   Now try an image with three sources. What do you think is happening?

   Make an amplitude image with constant level and zero phase. Back-transform this combination. Think about what this means if you wanted to measure the brightness of a source, with a limited amount of telescope time, too short to image it properly.

   Ok hotshot — the interferometer makes a fairly noisy measurement of the amplitude and phase. Can you:
   simulate a phase error? How big can the error get before the image degrades?
   simulate the effect of amplitude errors on one baseline of the interferometer.
   Hint: You may need to use the `imgen` task in MIRIAD.