

P2DFFT: Measuring Galactic Spiral Arm Pitch Angles

2DFFT (original) Author¹: Dr. Ivanio Puerari
Instituto Nacional de Astrofisica, Optica y Electronica
Santa Maria Tonantzintla, Puebla, Mexico

2DFFT (revised) Lead Author²: Dr. Marc Seigar
University of Minnesota Duluth
Duluth, MN, USA

2DFFT (progenitor of *P2DFFT*) Lead Author³: Dr. Benjamin Davis
Swinburne University of Technology
Centre for Astrophysics and Supercomputing
Melbourne, Victoria, Australia
<http://www.d.umn.edu/~msseigar/2DFFT/2DFFT.tar.gz>

P2DFFT By: Ian Hewitt & Dr. Patrick Treuthardt
North Carolina Museum of Natural Sciences
Astronomy & Astrophysics Research Lab
Raleigh, NC, USA
<http://github.com/treuthardt/P2DFFT>

Package Version: 5.2.2 – 20-June-2019



¹ Saraiva Schroeder, M. F., Pastoriza, M. G., Kepler, S. O., & Puerari, I. 1994, "The Distribution of Light in the Spiral Galaxy NGC 7412", A&AS 108: 41

² Seigar, M. S., et al. 2005, "Dust-Penetrated Arm Classes: Insights from Rising and Falling Rotation Curves", MNRAS 359: 1065

³ Davis, B. L., et al. 2012, "Measurement of Galactic Logarithmic Spiral Arm Pitch Angle Using Two-Dimensional Fast Fourier Transform Decomposition", ApJS 199: 33

Table of Contents

Table of Contents.....	2
LICENSE	3
GENERAL NOTES	4
INSTALLATION	4
Linux Installation Instructions	4
macOS Installation.....	4
UNDERSTANDING THE ALGORITHM	5
GALAXY ANALYSIS	6
Input Images	6
Data Flow	6
MAIN PROGRAMS	7
<i>p2dfft</i>	7
Command Line Arguments	7
Using an Input File	8
Command Line Input.....	8
Options.....	8
Output.....	9
<i>p2ifft</i> Inverse FFT Utility.....	9
Options.....	10
<i>p2spiral</i> UTILITY	10
Options.....	11
Image Properties.....	11
<i>p2pa</i> Results Program.....	12
Options.....	14
Zooniverse Project Support	15
OPTIONAL UTILITIES.....	15
Logarithmic Spiral Overlay	15
<i>p2boost</i> Contrast Booster	16
Other Utilities.....	16
MAKING CHANGES.....	17
REFERENCING THIS WORK	17
REFERENCES.....	17

LICENSE

P2DFFT Spiral Galaxy Arm Pitch Angle Analysis Suite

Copyright © 2016-2019 Ian B. Hewitt & Dr. Patrick Treuthardt

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see < <https://www.gnu.org/licenses/> >.

The authors can be contacted at:

*North Carolina Museum of Natural Sciences
Astronomy & Astrophysics Laboratory
11 West Jones Street
Raleigh, NC, 27601 USA
+1.919.707.9800*

-- or --

patrick.treuthardt@naturalsciences.org

GENERAL NOTES

As of version 5, this package is developed and tested under Ubuntu 18.10 LTS. It also includes build information for macOS Mojave, but this has had limited testing at the current time. Presently, there is no Windows build, but there are plans to test this with WSL (or it's successors) at some point in the future.

Please also note that this project is currently under active development and is subject to changes. In addition, because it is under development, there are tools and options included which are designed for testing and validation that are not normally needed for galaxy analysis.

INSTALLATION

Linux Installation Instructions

Unlike previous versions, P2DFFT version 5.x has a monolithic program for the Fourier transforms and no longer needs any linked/supporting programs (e.g. *scripter* or *get_maximum_pitch_phase*). The general installation instructions are:

1. Python 3.x and g++ need to be installed. Python does not need to be the default Python interpreter on your system.
2. Install the required libraries. On Debian/Ubuntu, this can be done by:

```
sudo apt install libfftw3-dev libcfitsio-dev libpthread-stubs0-dev  
sudo apt install libmagic-dev libcurl4-openssl-dev python3-pyqt5  
sudo apt install python3-numpy python3-matplotlib python3-astropy  
sudo apt install python3-pip  
pip3 install opencv-python
```
3. Edit the BIN_DIR variable in the makefile and set it to the directory where you would like the binaries installed. This directory should either be in your command path variable or need to be added to PATH (this assumes Bourne shell/bash).
4. Compile the main programs with the command:

```
make all (If you want to manually copy the executables to their destination)  
-- or --  
make install (Will build and install them in BIN_DIR)
```
5. The optional programs can be built with:

```
make opt
```


and then installed with:

```
make optinstall
```

macOS Installation

1. The command line tools need to be installed. These come with Xcode, so if that is already installed, you can skip to the next step. The smaller package OSX-GCC-Installer package can be used in lieu of Xcode. This can be installed from Apple. HOWEVER, do not install the OSX-GCC-Installer over an existing Xcode installation.

2. Homebrew needs to be installed.
ruby -e "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
3. Install Python 3
brew install python
4. P2DFFT also requires the installation of some additional libraries. Please Note: This assumes that you are running macOS 10.14. If you are running a different version, you will need different headers!

```
brew install llvm
sudo installer -pkg
/Library/Developer/CommandLineTools/Packages/macOS_SDK_headers_for_m
acOS_10.14.pkg -target /
brew install libmagic
brew install fftw
brew install cfitsio
```

5. Edit the BIN_DIR variable in *makefile.macos* and set it to the directory where you would like the binaries installed. This directory should either be in your command path variable or need to be added to PATH (this assumes Bourne shell/bash).
6. Compile the main items with the command:
make -f makefile.macos all (Need to manually copy executables to destination)
-- or --
make -f makefile.macos all install (Will build and install them in BIN_DIR)
7. The optional programs (see below) can be built with:
make -f makefile.macos all opt
and then installed with:
make -f makefile.macos all optinstall

UNDERSTANDING THE ALGORITHM

This software package is designed to estimate the pitch angle of spiral galaxy arms. It uses the Fourier decomposition technique used in Davis et. al. [2012]. This involves taking a deprojected galaxy image and mapping it into a rectangular polar projection using $\ln(\text{radius})$ and azimuthal angle coordinates. The image is mapped multiple times with the first pass mapping every radius value from the center from 1 pixel to the edge of the image, creating an annulus. Subsequent mappings repeatedly enlarge the inner annulus radius by 1 pixel each time. Each projected image is subjected to a two-dimensional fast Fourier transform and a low pass filter applied to get the results. The Fourier data is then summed and used to derive a pitch angle estimate.

The code has some comments, but to understand the algorithm further, please see the paper "Measurement of Galactic Logarithmic Spiral Arm Pitch Angle Using Two-Dimensional Fast

Fourier Transform Decomposition.⁴ [Davis et. al. 2012] and “Fourier analysis of structure in spiral galaxies⁵” [Puerari & Dottori 1992].

GALAXY ANALYSIS

This package is designed to take binary or ASCII FITS⁶ images as input and provide an estimate of spiral galaxy arm pitch angle as an output. This section covers an example of how to get the measurements.

Input Images

For good results, the following is recommended.

- Crop images so the galaxy is centered in the frame. Please note it is no longer necessary for the images to be perfectly square, only that the galaxy is centered properly. *p2dfft* will virtually crop the image to square the dimensions.
- The maximum image size is 2048 x 2048 pixels.
- The galaxy should be deprojected to a face-on orientation. This is not done in *p2dfft* but can be done with tools like SExtractor^{7,8} or IRAF/PYRAF^{9,10}.
- Any bright stars in the field should be masked for best results.
- Binary FITS images be directly input into *p2dfft*, but ASCII text versions of FITS images can also be used (like those created by IRAF *wtextimage*) as long as they have two extra bytes at the beginning. This is for compatibility with previous 2DFFT (the predecessor package) data files. These bytes are meant to hold the dimensions of the image, but can be zero, as *p2dfft* will calculate the dimensions based on the size of the data file. If these bytes are not present, the mapping of the image will be incorrect, and the results may be altered. Please also note that *p2dfft* can calculate the ASCII FITS file size if the image is square.

Data Flow

The general data flow for most users is shown in Figure 1. Please note that the steps in yellow are optional. The simplest data flow is to take deprojected, centered, binary FITS images and input them into *p2dfft*. *p2pa* can then be used to interpret the results.

⁴ Available at: <https://arxiv.org/abs/1202.4780>

⁵ Available at: http://articles.adsabs.harvard.edu/cgi-bin/nph-article_query?1992A%26AS...93..469P&data_type=PDF_HIGH&whole_paper=YES&type=PRINTER&filetype=.pdf

⁶ See NASA’s FITS Support Office <https://fits.gsfc.nasa.gov/>

⁷ SExtractor can be found at Astromatic.net <https://www.astromatic.net/software/sextactor>

⁸ A Guide to SExtractor can be found at http://astroa.physics.metu.edu.tr/MANUALS/sextactor/Guide2source_extractor.pdf

⁹ Part of Astroconda <https://astroconda.readthedocs.io/en/latest/index.html>

¹⁰ Specific information on pyraf can be found at http://www.stsci.edu/institute/software_hardware/pyraf

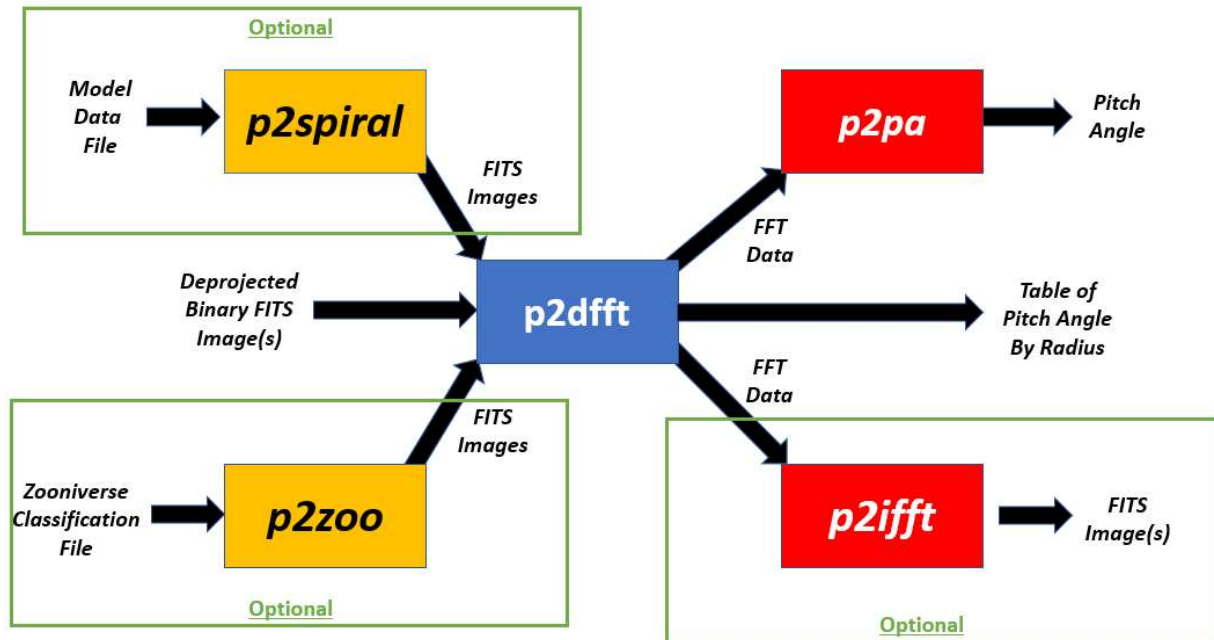


Figure 1. Data flow graph for galaxy analysis. The green boxes represent optional steps that are only used when specific inputs or output images are needed.

MAIN PROGRAMS

p2dfft

p2dfft will process a list of files and perform two dimensional FFT on each image in a set of annuli. The input files can come from standard input, the command line, or an input file. Files can be either binary FITS files or ASCII text FITS files or a combination of both. *p2dfft* will use multiple threads (one per core) to process the Fourier transforms in parallel to improve execution time (see the *PA_Notes.pdf* file for details and comparison to previous versions).

By default, *p2dfft* will determine the maximum outer annulus radius from the image size and the output file prefix will be the input filename minus the .fits extension. If different values are needed for either one of these, it can be done via command line options, or using an input file (see below).

Command Line Arguments

This is the easiest method to specify an arbitrary number of input files. Wildcard arguments can also be used.

```
p2dfft <args>
```

By default, there will be minimal information printed to the console, with only error messages being displayed. All the files on the command line will be processed with a default maximum radius from the center to the edge of the image and an output file prefix the same as the name of the image file (minus the .fits suffix).

Using an Input File

Using an input file allows you to specify not only the FITS images to be processed, but also optionally specify the output file prefix and the outer radius to be used for each file. The input file has the form:

```
Image_file_1,result_file_1,outer_radius_1
Image_file_2,result_file_2,outer_radius_2
Image_file_3,result_file_3,outer_radius_3
...
Image_file_n,result_file_n,outer_radius_n
```

You can specify the input file via a command line option:

```
p2dfft -i <file>
- or -
p2dfft --input <file>
```

The file can include blank lines and these will be ignored, but unlike previous versions, leading and trailing blank lines are no longer required. In addition, the output prefix and radius fields are optional. You can leave off the outer radius field and *p2dfft* will calculate it based on the image size. You can also leave off both the result file and outer radius field and these will default to the image file name (minus the .fits prefix) and the maximum outer radius based on image size, respectively.

Command Line Input

This feature is designed to support legacy 2DFFT input files, which means you can use files generated by the *scripter* program and they will correctly execute, as long as you create a link to *p2dfft* with the name *2dfft* somewhere in your command path. Using *p2dfft* this way will make it slower as it will operate in a single threaded mode.

Options

There are multiple non-exclusive command line options to modify the behavior of the program. Please note that some of these options are experimental and have not been validated to return results superior or equivalent to running *p2dfft* in default mode. The options are:

-v|--verbose: Prints multiple status messages during the processing.

-w|--warn: Causes warnings to be printed when individual values are not rational or when non-fatal processing problems are discovered. In most images, there will be some cases (especially in thin annuli) where these occur, and good results will still be obtained. This option will slow performance and is mainly useful if unexpected results are seen.

-r|--reverse: (Experimental) By default, the pitch angle is calculated using different annuli with increasing inner radius. This option performs the calculations using annuli with decreasing outer radius.

-f|--fixed: (Experimental) Use annuli with a fixed radius (specified by the integer argument to fixed). The inner radius of the annuli will still start at 1 and increase, but the end will always be that value plus the argument to fixed.

-p|--polar: Generate a FITS file with the logarithmic polar mapping used by P2DFFT (with an inner annulus of 1 and an outer annulus of the outer radius).

-z|--zero: (Experimental) Generate a zero filled padding around the polar projection to simulate an FFT window.

-m|--mask: (Experimental) Option to mask bright values (e.g. the core and bulge). A value of 0 will mask out any values \geq the value at the center of the image. A mask value of 1 will calculate the largest radius of continuous values that have a value \geq the center value and will eliminate (zero) all values in the polar plot below that radius. This eliminates structural artifacts that could effect the data when a mask value of 0 is used. This is most useful with the p2boost tool (see below).

Output

When *p2dfft* has completed it will create seven files for each image:

```
<result_file>_m0  
<result_file>_m1  
<result_file>_m2  
<result_file>_m3  
<result_file>_m4  
<result_file>_m5  
<result_file>_m6
```

These files contain the data for each harmonic mode (number of arms). The structure of the columns in these files from left to right is:

*Column 1 = The harmonic mode number (0-6)
Column 2 = "outiX_m2" where "X" is the starting inner radius in pixels
Column 3 = The highest amplitude Fourier frequency for this annulus and mode
Column 4 = The highest Fourier amplitude value for this annulus and mode
Column 5 = Pitch angle from the highest amplitude frequency (degrees)
Column 6 = Phase angle from the highest amplitude frequency (degrees)*

In addition, a directory called *<result_file>* will be created for each image. *p2dfft* will copy all the output Fourier data files (two for each radius) to this directory. These files are used in *p2pa* and *p2ifft*.

p2ifft Inverse FFT Utility

The *p2ifft* utility can be used to make a map of the filtered and summed FFT signal over all annuli from *p2dfft*. This mapping will be output as a floating-point binary FITS file and (optionally) an

ASCII text FITS file. *p2ifft* takes a series of names as input. Each argument is a name of a directory containing all the FFT data files output from *p2dfft* (i.e. *.dat* and *.rip* files for each radius). The names can also be entered via an input file which contains a list of files (and options). The *p2ifft* utility will also accept the original file name (e.g. *image1.fits*) and it will strip the *.fits* extension and look for the corresponding directory and data files.

By default, *p2ifft* will plot all modes (number of arms), but there are command line options to only print certain modes. Please note that in no case will mode 0 be plotted. The output files will have a prefix of *I_* (for the default all mode output) or *I_<mode>_* (if the *-m* option is used) so the output will not collide with any original image files. A sample output is shown in Figure 2.

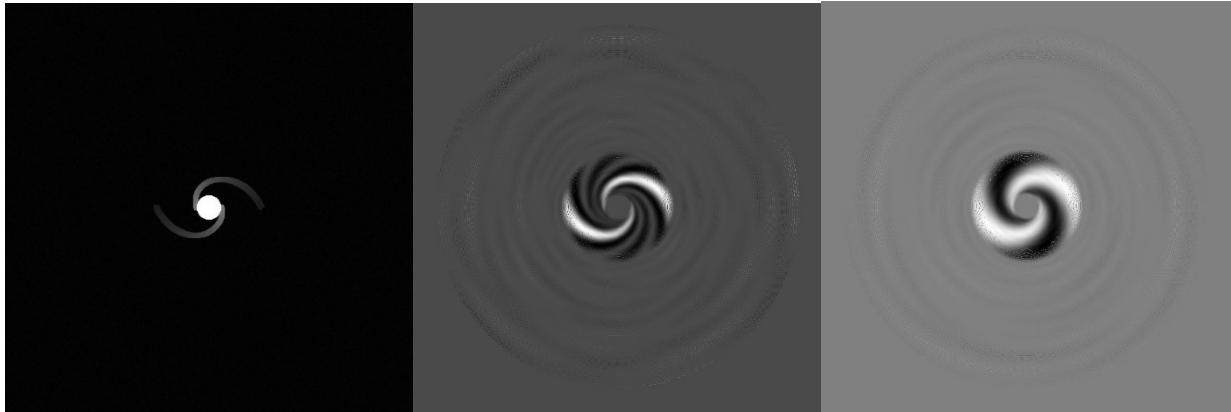


Figure 2. Sample *p2ifft* output. (Left) Model image input originally into *p2dfft*. (Center) *p2ifft* output for all modes. (Right) *p2ifft* output image for mode 2 only.

Options

```
ifft [-i|--input <file>] [-v|--verbose] [-m|--mode <n>[,<n>...]] [-s|--start <arg>] [-e|--end <arg>] [<name>[ <name>...]]
```

-i|--input: Will read file names and data file prefixes from the file specified with this option instead of from the command line.

-v|--verbose: Prints status messages during the processing.

-m|--mode: Specifies the modes to be used in the output image. By default, all modes (1-6) are used. The *-m* option specifies comma separated list of modes to be used in the plot, e.g. *-m 1,3,5,6*.

-s|--start: Specify a starting inner radius (default is 1).

-e|--end: Specify an ending inner radius (default is file size - 10% per Davis et.al., [2012]).

p2spiral UTILITY

The *p2spiral* program can be used to generate one or more images that contain simple galaxy models with logarithmic spirals (See Figure 2 (Left) for an example). The input data file contains a line for each image to be created. Each line has several fields separated by a comma, space, or

tab (see below for a description of the fields). The software package includes an example input file *sp_input.txt*. If *p2spiral* is called with no command line arguments, the user will be prompted for all the parameters (see below). Entering <ctrl>-D will stop the input and start the generation of the files.

All the output images will be 32 bit binary floating point FITS files. The FITS header will have some extra keywords which can provide program information about the models. The added keywords are:

- *COLORSPC* – Color space value which is always set to grayscale.
- *ARMS* – Set to the number of ARMS generated.
- *AVGPITCH* – Floating point value of the average pitch angle for the arms.
- *MINPITCH* – Floating-point value of the minimum pitch value for the arms.
- *MAXPITCH* – Floating point value of the maximum pitch value for the arms.

Please note that these extra keywords may be read by programs like *p2pa* and affect the output results.

Options

The complete list of options for *spiral* is:

-i|--input: Will read file names and keywords from the file specified with this option instead of standard input.

-v|--verbose: Causes status messages to be printed.

-t|--text: Generate ASCII FITS files as well as binary.

-p|--print: Print a listing of pitch angle by radius to stdout. This is useful for variable pitch angle spiral arms.

Image Properties

The following properties can be set for each image:

- *<name>* - The file name to be generated (not including the .fits suffix).
- *<pa>* - Floating point value for pitch angle of the arms from -75° to +75°. Positive values have S-wise chirality and negative values have Z-wise chirality.
- *<arms>* - Integer number of spiral arms in the image from 1 to 6.
- *<hsize>* - Integer height of the image in pixels. NOTE: P2DFFT returns better results with odd sized images.
- *<vsiz>* - Integer height of the image in pixels.
- *<feath>* - Integer value from 0 to 10 setting the added thickness of arm points in pixels from the center line.
- *<sweep>* - Floating point value from 0.0° to 720.0° which indicates how far in polar coordinates to map the arms in degrees (e.g. 720.0° is two complete circles).

- `<rot>` - Rotation of the arm starting position, -90° to $+90^\circ$. By default (0.0) the first arm will start at “noon” and the others evenly spaced around that. Positive values move the start of the arms in a clockwise direction.
- `<r0>` - Initial integer radius of the core/start of arms from 0 to `<size>-1`. NOTE: This is only for the circular core and does not include any bar features (see below).
- `<core>` - Set the core feature appearance:
 - A value of 0 means the core is not filled in at all.
 - A value of 1 fills core with the starting luminosity (inner) value of the arms.
 - A value of 2 fills the central core with values twice the starting luminosity (inner) value of the arms.
- `<bara>` - Semi-major axis of bar (0=no bar), must be an integer. This value will not overwrite the `<r0>` (core radius) value if it is less than `<r0>`.
- `<barb>` - Semi-minor axis of bar (integer). Ignored if `<bara>=0`.
- `<mar>` - Integer value for outside margin in pixels. Nothing will be mapped closer to the edge of the image than this, no matter what other settings are used.
- `<fg>` - Starting foreground luminosity value for arms.
- `<bg>` - Background value in the image. This is equivalent to the bias and any `<noise>` will be added to this value.
- `<delta>` - Change in pitch angle (negative or positive) in degrees of the arms over their length. This value is in degrees from -60.0° to 60.0° . NOTE: The pitch angle formula becomes unstable with decreasing pitch angle, so *p2spiral* uses a modified algorithm that won't ever change the pitch angle in the wrong direction. This means if the pitch angle will not increase at any point along the arm.
- `<lum>` - Change in luminosity (negative or positive) of an arm over its length. This is a percentage value between -0.99 and 0.99.
- `<log>` - Rate of change in arm luminosity: (1) is logarithmic and (0) linear.
- `<arm_lum>` - Apply change in luminosity over the width of an arm as well as its length. A value of 0 means constant width luminosity and 1 means decreasing luminosity.
- `<noise>` - This is a floating-point value controlling the simulated Poisson (shot) noise in the image. This value sets the largest possible value of the noise in any pixel and the actual noise values used in any given pixel will be random up to this amount. Please note that noise will not be added to the core or arms and those values should be set above the noise value.

p2pa Results Program

The *p2pa* program will analyze and plot the data output from the *p2dfft* program. It will read the filename prefixes specified on the command line and look for a directory with the FFT data files. For convenience, you can also specify the original .fits filename and *p2pa* will strip the extension and look for the proper output files. This program will take the FFT output data from *p2dfft* and sum it together to create integrated frequency/amplitude curves.

As of version 5.0 the default behavior of the analysis has changed. By default, the program will check the FITS file to see if there are header keyword values for ARMS and BAR. Since the

mode selection has a significant effect on the final pitch angle reported, it is important to understand how *p2pa* determines which mode to use. It uses the following rules:

- If the ARMS keyword is found, that will be the mode selected for the pitch angle despite the Fourier component amplitudes of other modes. This is the default behavior.
- If the *-a* option is specified, this will override the ARMS value and *p2pa* will use the mode specified for the *-a* option in pitch angle calculations.
- If the *-m* option is specified, the ARMS value will be ignored (NOTE: The *-a* argument has precedence over the *-m*) and the mode with highest Fourier amplitude will be used. If there is no ARMS keyword and *-a* is not specified, this will be the default behavior.

The BAR keyword value will control the starting inner radius used in the calculations:

- If the BAR keyword is found, that will be the starting inner annulus value.
- If the *-s* option is specified, this will override the BAR value.
- If the *-n* option is specified, the BAR value will be ignored and 1 will be used as the inner starting annulus value (The *-s* argument has precedence over the *-n*). If there is no *-s* option specified and no BAR keyword, this will be the default.

p2pa will print out a confirmation of which arm and starting radius are used for each image in the output file. The output file will be in CSV format and have the following columns (in order):

- File - File name (minus .fits extension).
- Start - Smallest inner radius of the annuli.
- End - Largest inner radius of the annuli.
- Mode - The mode used to calculate the PA.
- PA - Pitch angle value.
- Error - Error calculated based on Davis et. al. [2012]. Please note that this is the \pm error amount (not the total error amount).
- Mode Method – The mode determination method (*Amp* – for max amplitude, *ARMS* – for ARMS keyword, *Man* – Manual choice via *-a*).
- Bar Method – The method to determine the bar (*BAR* – for BAR keyword, *Man* – Manual via the *-s* option, *None* – No bar using the *-n* option).

In addition, if plots are generated, a PDF file will be created with the plots. All plots will have a black line indicating the 0 Hz frequency component of the curve. A sample plot is shown in Figure 3.

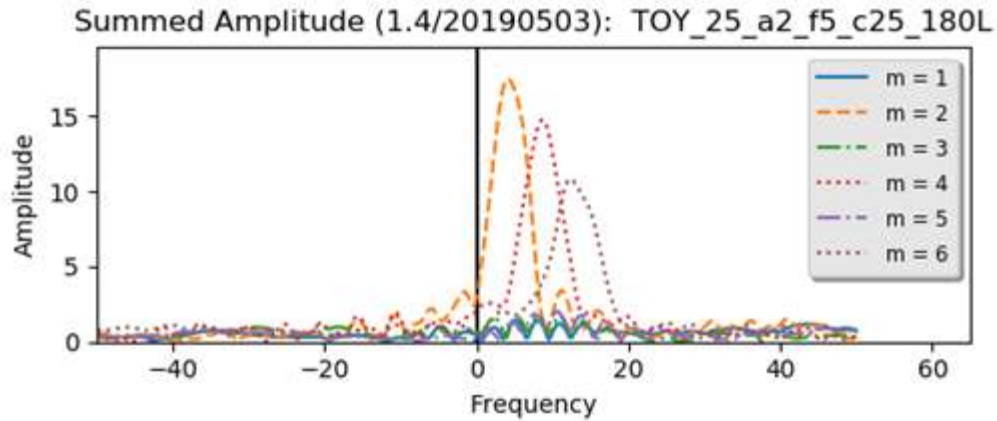


Figure 3. Sample plot of Fourier amplitude vs. frequency for a simple toy model galaxy. The model has 2 arms winding 180 degrees with a pitch angle of 25 degrees, 11 pixel arm width, and a 25 pixel bulge radius.

Options

The entire list of options for the program is:

`p2pa [-p|--plot] [-s|--start <num>] [-e|--end <num>] [-o|--output <name>] [-a|--arms <1-6>] [-w|--warning] [-n|--nobar] [-m|--magnitude] <file> [<file>...]`

<files> (Required) The name of the files to be processed. Can be multiple/wildcard values and can include the .fits/.fts extension (which will be stripped) or just the prefix.

-p|--plot (Optional) Produce a PDF plot of the frequency amplitude curve for each input file.

-s|--start <arg> (Optional) Initial annulus inner radius for calculations.

-e|--end <arg> (Optional) Ending annulus inner radius for calculations.

-o|--output (Optional) The name of the output text file (default is Results.csv).

-n|--nobar (Optional) Ignore the BAR FITS header value (if given). Will be default if no BAR header value is found (starting radius is 1) and -s is not specified.

-m|--magnitude (Optional) Ignore the ARM FITS header value (if given) and will use the largest Fourier magnitude value to determine the mode. Will be default if no ARM header value is found or -a is not specified.

-a|--arms (Optional) Force the preferred mode number. This option will override the use of max amplitude to select the mode.

-w|--warning (Optional) Print warnings for invalid data, etc.

There are also some constants that can be changed in the script. The most important is the EXCLUDE constant. If this is set to True, the last 10% of the annulus inner radius values will not be used. This is recommended as these values have been shown to be less reliable [Davis et. al. 2012].

Zooniverse Project Support

The *p2zoo* utility can be used to interpret the data from the *Spiral Graph*¹¹ project on Zooniverse. It will read the Zooniverse classifications CSV data file and build 32-bit binary FITS images from the measured points. It uses the following rules in generating the files:

- The images will be sized based on the outermost mapping of the points (with a 20% margin beyond the last point - but this can be set by changing the PAD variable in *p2zoo*).
- The images will have the following additional keywords:
 - ARMS will indicate the number of arms (since this is known when the object is traced).
 - BAR which will contain the estimated bar/core feature radius (in pixels) based on the minimum distance from the center to the innermost traced points.
- Images will have the name of the subject ID from the Zooniverse dataset, unless the *-n* option is specified in which case the filename specified in the CSV file will be used.
- By default, the core is not filled and only the arms are drawn. This can be changed using the *-c* option.
- Each arm segment from the data file will have the points plotted and then connected with a line. The generated arms will have a padding of three pixels in all directions (for a width of 7 pixels). This can be changed with the *-f* option.
- Multiple attempts on the same file will sum all the tracings into a single file with the subject name.
 - The tracing will be additive with overlapping areas being brighter.
 - The ARMS variable will be calculated as the median of all attempts rounded to the nearest integer.

A second utility that supports the *Spiral Graph* project is *p2zname*. This utility will change the file names generated by *p2zoo* from subject id's to the original filename (contained in the data file).

OPTIONAL UTILITIES

Logarithmic Spiral Overlay

The *p2logsp* program can be used to overlay a logarithmic spiral pattern over a FITS image to manually estimate the pitch angle of the spiral arms. The program takes only a single argument, which is the image to be measured. The program will bring up a simple graphical user interface with buttons for the spiral parameters. As the parameters are changed, the overlaid spiral pattern will be changed accordingly. The recommended process to get a reasonable fit is:

¹¹ <https://www.zooniverse.org/projects/astro-lab-ncmns/spiral-graph>

- Adjust the $R0$ value so that it matches the end of the bar/core feature.
- Adjust the rotation of the start of the arms using the *Rot* buttons.
- Set the chirality to match the image using the *S-wise* and *Z-wise* buttons.
- Set the number of arms in the spiral.
- Adjust the pitch angle using the *PA* buttons until the desired fit is obtained.

Please note that the only parameter that cannot be changed in the GUI is the azimuthal extent of the overlay arms. This is set to 180° by default but can be changed by editing the SWEEP variable in the program.

p2boost Contrast Booster

This is an experimental program and has not been tested to see if it returns improved or consistent results. It is designed to boost the contrast on a FITS image for better arm discrimination using image sharpening. It will generate a FITS image with the prefix C_ as the boosted image. Since the thresholds for the image sharpening routines have many parameters, the *-i* option can will write a FITS image at each image processing step. These interim files will have the prefix of B1_, B2_, etc.

The program uses the OpenCV2 libraries and the overall algorithm is:

- Adjust lowermost 5% and uppermost 5% of brightness values to the 5% and 95% threshold, respectively (interim file prefix B1_).
- Find the brightness at center and set that as the new upper threshold (interim file prefix B2_).
- Rescale the image over an 8-bit range (0.0-255.0) (interim file prefix B3_).
- Execute a bilateral filter¹² on the image (interim file prefix B4_).
- Perform a simple background neutralization by taking the minimum pixel value and subtracting that from all pixels (no interim file).
- Execute a Local Contrast Enhancement (final file prefix C_).

The *-p* option will produce histograms of the images at various steps. The histograms produced are (in order):

- Raw histogram of original image.
- Histogram after the bilateral filter is applied.
- Histogram after the local contrast enhancement is applied.

Other Utilities

These utilities are useful for testing changes to the programs or to understand how a particular result is obtained.

¹² See https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf for more details on bilateral filters

p2chart_freq - This utility will print the mapping between the FFT frequency values and their calculated pitch angles for all modes.

p2filter - This is a useful utility for tuning the parameters of *p2boost*. It will apply a set of transforms with different parameters to an input image and generate the resulting images.

p2map - This utility does a $\ln(\text{radius})$ /polar projection of a FITS image into the same 1024x2048 FITS image format used by *p2dfft* for the 2D FFT analysis.

p2txt2fits - utility can be used to convert ASCII FITS files to 32-bit floating point binary FITS files. This duplicates the functionality in IRAF.

- The program will take any number of ASCII FITS file names as arguments, convert them, and write them out with .fits extensions.
- There are two command line options:
 - By default, *p2txt2fits* will calculate the size of the FITS image in the text file. If you would rather this size be read from the first two bytes, use the *-r|--readsize* option.
 - The *-v|--verbose* option will print status messages during the conversion.

MAKING CHANGES

If you want to make changes, the *-v|--verbose* option may not provide enough output for debugging. If you set `DEBUG` to 1 in *globals.h*, the main programs (*p2dfft*, *p2ifft*, etc.) will print out copious amounts of output data. If you need to check the input/output matrices, you can set `DEBUG_DAT` and/or `DEBUG_MAT` to 1 in *p2dfft.cpp*. This will print out the contents of the matrices but will produce a lot (> 4M lines) of output data.

If you do make changes in the code, there is a separate package to test these changes (*p2dfft-test*). The test code package is large, so it is not included in this package. Contact the authors for access to this package.

REFERENCING THIS WORK

If you want to reference or publish results using this work, please cite Davis et. al. 2012 and Mutlu-Pakdial et. al. 2018.

REFERENCES

Davis, B. L., Berrier, J. C., Shields, D. W., Kennefick, J., Kennefick, D., Seigar, M. S., Lacy, C. H.S., & Puerari, I., 2012, ApJS, 199, 33

Mutlu-Pakdil, B., Seigar, M. S., Hewitt, I. B., Treuthardt, P., Berrier J. C., & Koval, L. E., 2018, MNRAS, 474, 2594

Puerari, I. & Dottori, H. A., 1992, A&AS, 93, 469