

---

# **FractalPy**

*Release 0.1.0*

**Fergus O'Hanlon**

**Jan 23, 2023**



**CONTENTS:**

- 1 src** **1**
- 1.1 fractalpy package . . . . . 1
- 2 Indices and tables** **7**
- Python Module Index** **9**
- Index** **11**



## 1.1 fractalpy package

### 1.1.1 Subpackages

`fractalpy.cli` package

**Subpackages**

`fractalpy.cli.commands` package

**Submodules**

`fractalpy.cli.commands.cmd_fractal_base` module

`fractalpy.cli.commands.cmd_fractal_base.needs_options(f)`

`fractalpy.cli.commands.cmd_julia` module

`fractalpy.cli.commands.cmd_mandelbrot` module

**Module contents**

**Submodules**

`fractalpy.cli.cli_main` module

This module provides the Command Line Interface for this package.

`fractalpy.cli.cli_main.main()`

## fractalpy.cli.helper module

fractalpy.cli.helper.get\_default\_args(*func*)

## Module contents

### fractalpy.fractals package

#### Submodules

### fractalpy.fractals.fractal\_base module

This module provides the base fractal class for this package.

```
class fractalpy.fractals.fractal_base.FractalBase(limits: tuple[float, float, float, float], n_pts: int = 1000, threshold: int = 1000, color_map: str = 'hsv', c_set: str = 'black', pallet_len: int = 250, color_map_shift: int = 0)
```

Bases: `ABC`

An abstract class to represent a base class for fractals.

Initialises Fractal with either the Mandelbrot set or Julia set along with default attributes.

#### Parameters

- **limits** (*tuple, optional*) – Tuple of minimum and maximum values along x-axis and y-axis.
- **n\_pts** (*int, optional*) – Number of points along y-axis. Defaults to 1000.
- **threshold** (*int, optional*) – Number of iterations before point determined to be in the set. Defaults to 1000.
- **color\_map** (*str, optional*) – Color map for plots. Defaults to 'hsv'.
- **pallet\_len** (*int, optional*) – Length of periodicity for color pallet. Defaults to 250.
- **color\_map\_shift** (*int, optional*) – Length to shift color pallet. Defaults to 0.

#### abstract property color\_chart

The color chart of the calculated set

#### property color\_map

```
get_target_ranges(m: float, target: tuple[float, float])
```

Gets the x range and y range for the target point at corresponding zoom magnitude

```
abstract plot(fig_size: float = 5, axis: Union[str, bool, tuple[float, float, float, float]] = 'off', title: str = "", n_ticks: int = 5)
```

Plots the calculated set.

#### Parameters

- **fig\_size** (*int, optional*) – Size of figure in inches of x-axis for plot. Defaults to 4.
- **axis** (*tuple[float, float, float, float] or bool or string, optional*) – Axis parameters for plot. Defaults to 'off'.

- **title** (*str*, *optional*) – title of plot. Defaults to ‘’.
- **n\_ticks** (*int*, *optional*) – Number of ticks for axes. Defaults to 5.

#### property ratio

A property for the aspect ratio

**abstract save**(*filename: str = ''*, *extension: str = 'png'*)

Saves an image of the calculated set in the ‘images’ directory.

#### Parameters

- **filename** (*\_type\_*) – The filename of the saved image. Defaults to None.
- **frame\_iter** (*#*) – The frame iteration number. Defaults to ‘’.
- **extension** (*str*, *optional*) – The extension to save the image as. Defaults to ‘png’.

#### property x\_y\_ranges

A property for the ranges of the x-points and the y-points

**abstract zoom**(*m: float*, *target: tuple[float, float]*, *filename: str = ''*, *extension: str = 'gif'*, *frame\_subdir: str = 'frames'*, *n\_frames: int = 120*, *fps: int = 60*, *n\_jobs: int = 2*)

Compiles a video after generating a sequence of images, beginning with the object’s current co-ordinate frame and finishing at the target location frame.

#### Parameters

- **filename** (*str*, *optional*) – Name of output file. Defaults to None.
- **extension** (*str*, *optional*) – Extension of the output file format. Defaults to ‘gif’.
- **frame\_subdir** (*str*, *optional*) – Directory to save image frames. Defaults to ‘frames’.
- **m** (*float*, *optional*) – the zoom magnitude.
- **target** (*tuple[float, float]*, *optional*) – Target location for the end of zoom. The first tuple entry is location with x, y coordinates. Defaults to (6e+4, -1.186592e+0, -1.901211e-1). # TODO fix the defaults here. it depends on which class
- **n\_frames** (*int*, *optional*) – Number of total frames compiled. n\_frames-2 frames are compiled intermediately between the initial frame and target frame. Defaults to 120.
- **fps** (*int*, *optional*) – Number of frames per second for output video. This determines the smoothness between frames and affects the length of the output video. Defaults to 60.
- **n\_jobs** (*int*, *optional*) – Number of workers for parallel processing. Defaults to os.cpu\_count().

## fractalpy.fractals.fractals module

This module provides the usage classes for this package.

**class** fractalpy.fractals.fractals.**Julia**(*c: complex = -0.79 + 0.15j*, *limits: tuple[float, float, float, float] = (-1.5, 1.5, -1.5, 1.5)*, *\*\*kwargs*)

Bases: *FractalBase*

A class to represent the Julia set.

Initialises Fractal with either the Mandelbrot set or Julia set along with default attributes.

#### Parameters

- **limits** (*tuple*, *optional*) – Tuple of minimum and maximum values along x-axis and y-axis.
- **n\_pts** (*int*, *optional*) – Number of points along y-axis. Defaults to 1000.
- **threshold** (*int*, *optional*) – Number of iterations before point determined to be in the set. Defaults to 1000.
- **color\_map** (*str*, *optional*) – Color map for plots. Defaults to 'hsv'.
- **pallet\_len** (*int*, *optional*) – Length of periodicity for color pallet. Defaults to 250.
- **color\_map\_shift** (*int*, *optional*) – Length to shift color pallet. Defaults to 0.

**property color\_chart**

The color chart of the calculated set

**plot(\*\*kwargs)**

Plots the calculated set.

**Parameters**

- **fig\_size** (*int*, *optional*) – Size of figure in inches of x-axis for plot. Defaults to 4.
- **axis** (*tuple*[*float*, *float*, *float*, *float*] or *bool* or *string*, *optional*) – Axis parameters for plot. Defaults to 'off'.
- **title** (*str*, *optional*) – title of plot. Defaults to ''.
- **n\_ticks** (*int*, *optional*) – Number of ticks for axes. Defaults to 5.

**save(filename="", \*\*kwargs)**

Saves an image of the calculated set in the 'images' directory.

**Parameters**

- **filename** (*\_type\_*) – The filename of the saved image. Defaults to None.
- **frame\_iter** (*#*) – The frame iteration number. Defaults to ''.
- **extension** (*str*, *optional*) – The extension to save the image as. Defaults to 'png'.

**spin(filename: Optional[str] = None, extension: str = 'gif', n\_frames: int = 60, fps: int = 60, n\_jobs: int = 2)**

**Creates a sequence of images beginning with the objects current co-ordinate frame and finishing at the target location.**

**Parameters**

- **filename** (*\_type\_*, *optional*) – *\_description\_*. Defaults to None.
- **extension** (*str*, *optional*) – *\_description\_*. Defaults to 'gif'.
- **n\_frames** (*int*, *optional*) – *\_description\_*. Defaults to 60.
- **fps** (*int*, *optional*) – *\_description\_*. Defaults to 60.
- **n\_jobs** (*\_type\_*, *optional*) – *\_description\_*. Defaults to `os.cpu_count()`.

**zoom(m=5, target=(0.5, 0.5), \*\*kwargs)**

Compiles a video after generating a sequence of images, beginning with the object's current co-ordinate frame and finishing at the target location frame.

**Parameters**

- **filename** (*str*, *optional*) – Name of output file. Defaults to None.
- **extension** (*str*, *optional*) – Extension of the output file format. Defaults to ‘gif’.
- **frame\_subdir** (*str*, *optional*) – Directory to save image frames. Defaults to ‘frames’.
- **m** (*float*, *optional*) – the zoom magnitude.
- **target** (*tuple*[*float*, *float*], *optional*) – Target location for the end of zoom. The first tuple entry is location with x, y coordinates. Defaults to (6e+4, -1.186592e+0, -1.901211e-1). # TODO fix the defaults here. it depends on which class
- **n\_frames** (*int*, *optional*) – Number of total frames compiled. n\_frames-2 frames are compiled intermediately between the initial frame and target frame. Defaults to 120.
- **fps** (*int*, *optional*) – Number of frames per second for output video. This determines the smoothness between frames and affects the length of the output video. Defaults to 60.
- **n\_jobs** (*int*, *optional*) – Number of workers for parallel processing. Defaults to os.cpu\_count().

```
class fractalpy.fractals.fractals.Mandelbrot(limits: tuple[float, float, float, float] = (-2, 1, -1.5, 1.5),
                                             **kwargs)
```

Bases: *FractalBase*

A class to represent the Mandelbrot set.

Initialises Fractal with either the Mandelbrot set or Julia set along with default attributes.

#### Parameters

- **limits** (*tuple*, *optional*) – Tuple of minimum and maximum values along x-axis and y-axis.
- **n\_pts** (*int*, *optional*) – Number of points along y-axis. Defaults to 1000.
- **threshold** (*int*, *optional*) – Number of iterations before point determined to be in the set. Defaults to 1000.
- **color\_map** (*str*, *optional*) – Color map for plots. Defaults to ‘hsv’.
- **pallet\_len** (*int*, *optional*) – Length of periodicity for color pallet. Defaults to 250.
- **color\_map\_shift** (*int*, *optional*) – Length to shift color pallet. Defaults to 0.

#### property color\_chart

The color chart of the calculated set

#### plot(\*\*kwargs)

Plots the calculated set.

#### Parameters

- **fig\_size** (*int*, *optional*) – Size of figure in inches of x-axis for plot. Defaults to 4.
- **axis** (*tuple*[*float*, *float*, *float*, *float*] or *bool* or *string*, *optional*) – Axis parameters for plot. Defaults to ‘off’.
- **title** (*str*, *optional*) – title of plot. Defaults to ‘.’.
- **n\_ticks** (*int*, *optional*) – Number of ticks for axes. Defaults to 5.

#### save(filename="", \*\*kwargs)

Saves an image of the calculated set in the ‘images’ directory.

#### Parameters

- **filename** (*\_type\_*) – The filename of the saved image. Defaults to None.
- **frame\_iter** (#) – The frame iteration number. Defaults to ‘’.
- **extension** (*str*, *optional*) – The extension to save the image as. Defaults to ‘png’.

**zoom**(*m=60000.0*, *target=(-1.186592, -0.1901211)*, *\*\*kwargs*)

Compiles a video after generating a sequence of images, beginning with the object’s current co-ordinate frame and finishing at the target location frame.

#### Parameters

- **filename** (*str*, *optional*) – Name of output file. Defaults to None.
- **extension** (*str*, *optional*) – Extension of the output file format. Defaults to ‘gif’.
- **frame\_subdir** (*str*, *optional*) – Directory to save image frames. Defaults to ‘frames’.
- **m** (*float*, *optional*) – the zoom magnitude.
- **target** (*tuple[float, float]*, *optional*) – Target location for the end of zoom. The first tuple entry is location with x, y coordinates. Defaults to (6e+4, -1.186592e+0, -1.901211e-1). # TODO fix the defaults here. it depends on which class
- **n\_frames** (*int*, *optional*) – Number of total frames compiled. n\_frames-2 frames are compiled intermediately between the initial frame and target frame. Defaults to 120.
- **fps** (*int*, *optional*) – Number of frames per second for output video. This determines the smoothness between frames and affects the length of the output video. Defaults to 60.
- **n\_jobs** (*int*, *optional*) – Number of workers for parallel processing. Defaults to `os.cpu_count()`.

## Module contents

### 1.1.2 Module contents

Top-level script for this package.

## INDICES AND TABLES

- [genindex](#)
- [modindex](#)
- [search](#)

Please note that this webpage is currently under heavy development.



## PYTHON MODULE INDEX

### f

- fractalpy, 6
- fractalpy.cli, 2
  - fractalpy.cli.cli\_main, 1
  - fractalpy.cli.commands, 1
    - fractalpy.cli.commands.cmd\_fractal\_base, 1
    - fractalpy.cli.commands.cmd\_julia, 1
    - fractalpy.cli.commands.cmd\_mandelbrot, 1
  - fractalpy.cli.helper, 2
- fractalpy.fractals, 6
  - fractalpy.fractals.fractal\_base, 2
  - fractalpy.fractals.fractals, 3



## C

color\_chart (*fractalpy.fractals.fractal\_base.FractalBase property*), 2  
 color\_chart (*fractalpy.fractals.fractals.Julia property*), 4  
 color\_chart (*fractalpy.fractals.fractals.Mandelbrot property*), 5  
 color\_map (*fractalpy.fractals.fractal\_base.FractalBase property*), 2

## F

FractalBase (*class in fractalpy.fractals.fractal\_base*), 2  
 fractalpy  
   module, 6  
 fractalpy.cli  
   module, 2  
 fractalpy.cli.cli\_main  
   module, 1  
 fractalpy.cli.commands  
   module, 1  
 fractalpy.cli.commands.cmd\_fractal\_base  
   module, 1  
 fractalpy.cli.commands.cmd\_julia  
   module, 1  
 fractalpy.cli.commands.cmd\_mandelbrot  
   module, 1  
 fractalpy.cli.helper  
   module, 2  
 fractalpy.fractals  
   module, 6  
 fractalpy.fractals.fractal\_base  
   module, 2  
 fractalpy.fractals.fractals  
   module, 3

## G

get\_default\_args() (*in module fractalpy.cli.helper*), 2  
 get\_target\_ranges() (*fractalpy.fractals.fractal\_base.FractalBase method*), 2

## J

Julia (*class in fractalpy.fractals.fractals*), 3

## M

main() (*in module fractalpy.cli.cli\_main*), 1  
 Mandelbrot (*class in fractalpy.fractals.fractals*), 5  
 module  
   fractalpy, 6  
   fractalpy.cli, 2  
   fractalpy.cli.cli\_main, 1  
   fractalpy.cli.commands, 1  
   fractalpy.cli.commands.cmd\_fractal\_base, 1  
   fractalpy.cli.commands.cmd\_julia, 1  
   fractalpy.cli.commands.cmd\_mandelbrot, 1  
   fractalpy.cli.helper, 2  
   fractalpy.fractals, 6  
   fractalpy.fractals.fractal\_base, 2  
   fractalpy.fractals.fractals, 3

## N

needs\_options() (*in module fractalpy.cli.commands.cmd\_fractal\_base*), 1

## P

plot() (*fractalpy.fractals.fractal\_base.FractalBase method*), 2  
 plot() (*fractalpy.fractals.fractals.Julia method*), 4  
 plot() (*fractalpy.fractals.fractals.Mandelbrot method*), 5

## R

ratio (*fractalpy.fractals.fractal\_base.FractalBase property*), 3

## S

save() (*fractalpy.fractals.fractal\_base.FractalBase method*), 3  
 save() (*fractalpy.fractals.fractals.Julia method*), 4  
 save() (*fractalpy.fractals.fractals.Mandelbrot method*), 5

`spin()` (*fractalpy.fractals.fractals.Julia method*), 4

## X

`x_y_ranges` (*fractalpy.fractals.fractal\_base.FractalBase property*), 3

## Z

`zoom()` (*fractalpy.fractals.fractal\_base.FractalBase method*), 3

`zoom()` (*fractalpy.fractals.fractals.Julia method*), 4

`zoom()` (*fractalpy.fractals.fractals.Mandelbrot method*), 6