

S

EXPLORE

TRACKS

MY COURSES

EDPRESSO

REFER A FRIEND

educative

From Python to Numpy

0% COMPLETED

Search Course

Temporal Vectorization

Coding Example: The Mandelbrot Set (Python approach)

Coding Example: The Mandelbrot Set (NumPy approach)

Coding Example: Minkowski-Bouligand Dimension

Spatial Vectorization

Coding Example: Implement the behavior of Boids (Python approach)

Coding Example: Implement the behavior of Boids (NumPy approach)

Conclusion

We'll cover the following

Python Implementation

Output

## Coding Example: The Mandelbrot Set (Python approach)

In this lesson, we are going to look at the Python implementation of the Mandelbrot set case study.

Python Implementation #

A pure python implementation is written as:

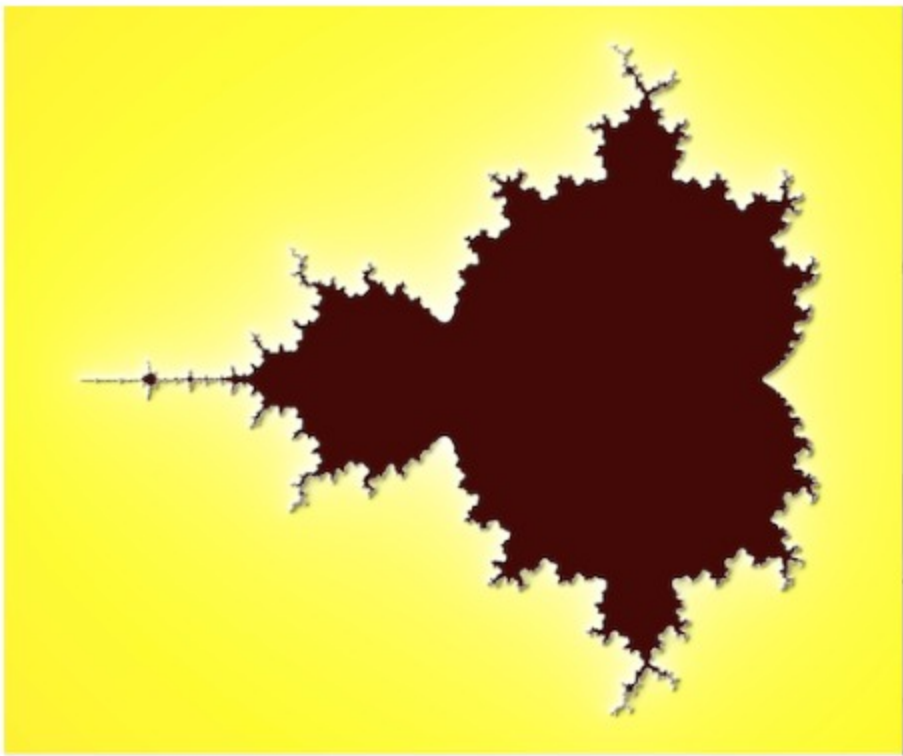
main.py

tools.py

```
1 # -----
2 # From Numpy to Python
3 # Copyright (2017) Nicolas P. Rougier - BSD license
4 # More information at https://github.com/rougier/numpy-book
5 # -----
6 import numpy as np
7
8 def mandelbrot_python(xmin, xmax, ymin, ymax, xn, yn, maxiter, horizon=2.0):
9     def mandelbrot(z, maxiter):
10         c = z
11         for n in range(maxiter):
12             if abs(z) > horizon:
13                 return n
14             z = z*z + c
15         return maxiter
16     r1 = [xmin+i*(xmax-xmin)/xn for i in range(xn)]
17     r2 = [ymin+i*(ymax-ymin)/yn for i in range(yn)]
18     return [mandelbrot(complex(r, i),maxiter) for r in r1 for i in r2]
19
20
21 def mandelbrot(xmin, xmax, ymin, ymax, xn, yn, itermax, horizon=2.0):
22     # Adapted from
23     # https://thesamovar.wordpress.com/2009/03/22/fast-fractals-with-python-and-numpy/
24     Xi, Yi = np.mgrid[0:xn, 0:yn]
25     Xi, Yi = Xi.astype(np.uint32), Yi.astype(np.uint32)
26     X = np.linspace(xmin, xmax, xn, dtype=np.float32)[Xi]
27     Y = np.linspace(ymin, ymax, yn, dtype=np.float32)[Yi]
28     C = X + Y*1j
```

Output #

The output of the above code would look like this:



The interesting (and slow) part of this code is the mandelbrot function that actually computes the sequence  $f_c(f_c(f_c...)))$ . The vectorization of such code is not totally straightforward because the internal return implies a differential processing of the element. Once it has diverged, we don't need to iterate any more and we can safely return the iteration count at divergence. Now the problem is to do the same in NumPy.

In the next lesson, we'll solve this example using the numpy approach.

← Back

Mark as Completed

Next →

Temporal Vectorization

Coding Example: The Mandelbrot Set (...)