A Quick Overview

This lesson sums up the discussion on Python to NumPy.

You’ve reached the end of this course. I hope you’ve learned something while reading it, I sure learned a lot writing it. Trying to explain something is a generally a good exercise to test for your knowledge of this thing. Of course, we only scratched the surface of NumPy and there are many things left to discover.

If there’s a single message to retain from this course it is “premature optimization is the root of all evil”. We’ve seen that code vectorization can drastically improve your computation, with several orders of magnitude in some cases. Still, problem vectorization is generally much more powerful. If you write code vectorization too early in your design process, you won’t be able to think out-of-the-box and you’ll certainly miss some really powerful alternatives because you won’t be able to identify your problem properly as we’ve seen in the problem vectorization chapter. This requires some experience and you have to be patient: experience is not an overnight process.

Finally, custom vectorization is an option worth considering once you’ve looked at the alternatives to NumPy. When nothing works for you, NumPy still offers you a clever framework to forge your own tools. And who knows, this can be the start of an exciting adventure for you and the community as it happened to me with the glumpy and the vispy packages.