

Jupyter notebooks and hubs for scientific computing

Michael P. Lamoureux* and Heather K. Hardeman-Vooy

mikel@ucalgary.ca, heather.hardeman@ucalgary.ca

Abstract

The journal Nature recently published an article “Why Jupyter is data scientists’ computational notebook of choice.” With three years of use under our belts, we discuss our experience with Jupyter notebooks, provide guidelines on how to make the transition to these tools for your own research, and present several useful resources to help you make this transition. As an illustrative example, we present a notebook recording our research efforts to achieve a 30x speedup in an implementation of standard finite difference code to numerically simulate acoustic waves in a variable velocity field using WebGL on a GPU-powered video card.

Motivating Quote (Nature, reference 2)

“Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. Computational notebooks have been around for decades, but Jupyter in particular has exploded in popularity over the past couple of years. This rapid uptake has been aided by an enthusiastic community of user-developers and a redesigned architecture that allows the notebook to speak dozens of programming languages – a fact reflected in its name, which was inspired, according to co-founder Fernando Pérez, by the programming languages Julia (Ju), Python (Pyt) and R.

One analysis of the code-sharing site GitHub counted more than 2.5 million public Jupyter notebooks..., up from 200,000 or so in 2015.”

Overview of notebooks and services

A Jupyter notebook is a computer document that can host and display formatted text, graphics, data, and live code in a single computer file. Users can assemble all the key parts of their research into a single document that explains the work, does the data analysis and computation with live code, and displays the results in a professional format. Each notebook is a collection of editable cells that can contain either formatted text, graphics commands, or live code. Using Markdown, the user creates text with headers, sections, enumerated lists, and complex mathematical formulas using LaTeX. Based on modern web standards, notebooks can be as complex as any web page, with responsive controls.

Notebooks can use a wide variety of languages, including Julia, Python, R, Matlab, and more. Sample notebooks are shown in Figures 1,2,3.

Some online services available to researchers are here:

- <https://ucalgary.szygy.ca> - a free service at the U of C
- <https://pims.szygy.ca> - a free service through PIMS, Cybera
- <http://jupyter.org/try> - A service for tests and demos
- <https://notebooks.azure.com> - a Microsoft commercial service

A Jupyter Exercise: 30x Speedup for GPU FD Code

We recorded the process of implementing finite difference code for a numerical simulation of the acoustic wave equation on a Graphic Processing Unit (GPU), in a Jupyter notebook. A key requirement was to create generic code that would run on any video card that hosts a fast GPU, using the WebGL API. The final result showed a speed up by a factor of 30, compared to Matlab. Details are in the 2018 CREWES report.

The target algorithm was a finite difference simulation of the acoustic wave equation in 2D, with a heterogeneous velocity field, on a 1000x1000 grid. With access to the video card, the simulation runs in real time with a graphical animation for output.

Figure 3 shows a screen shot of the GPU computation, computing the evolution of a wave packet reflecting and refracting through a horizontal interface. Table 4 has the results of speed tests showing the 30x speedup.

Sample Notebooks

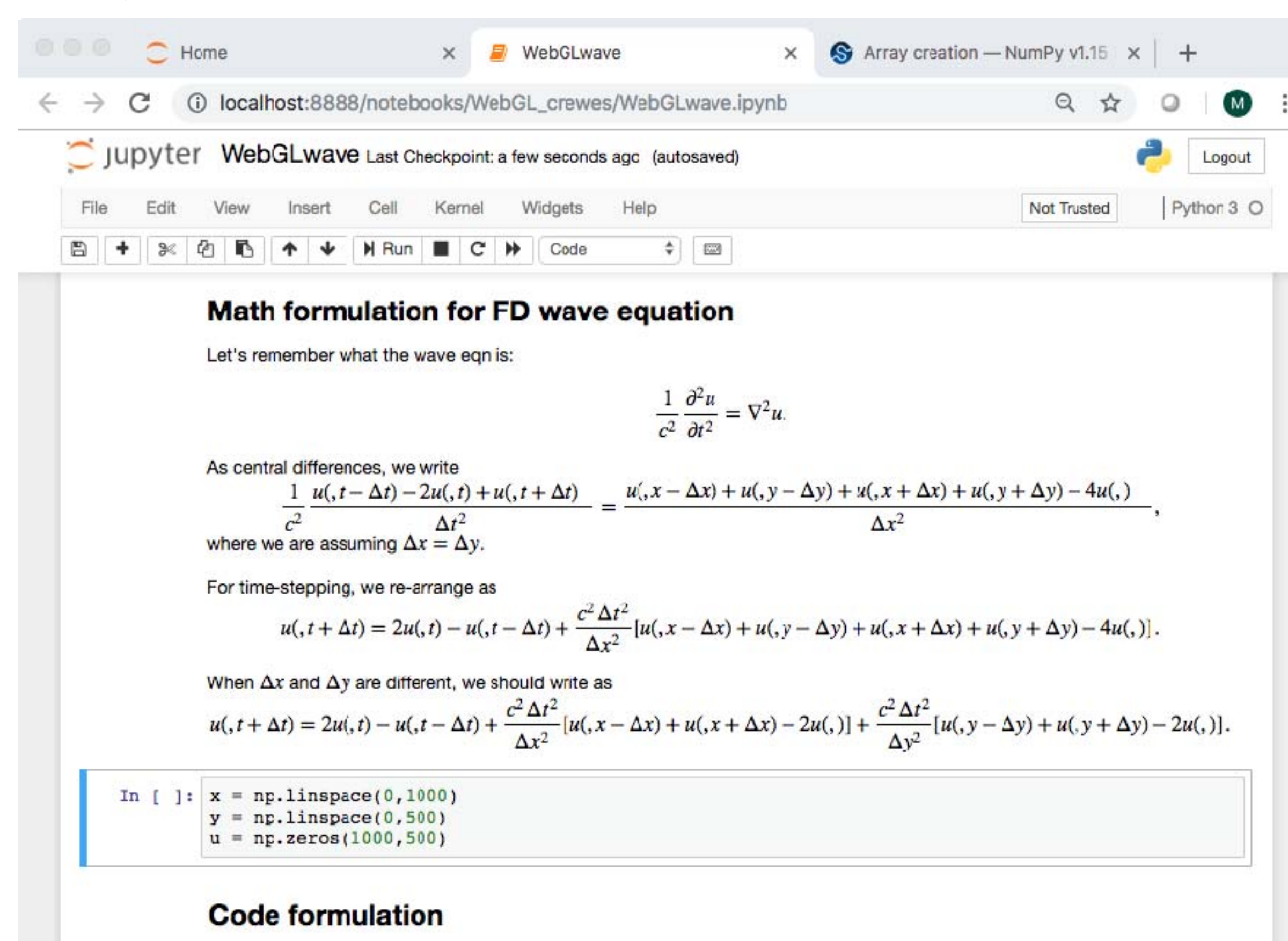


Figure 1: A Jupyter notebook with text, formulas, and code.

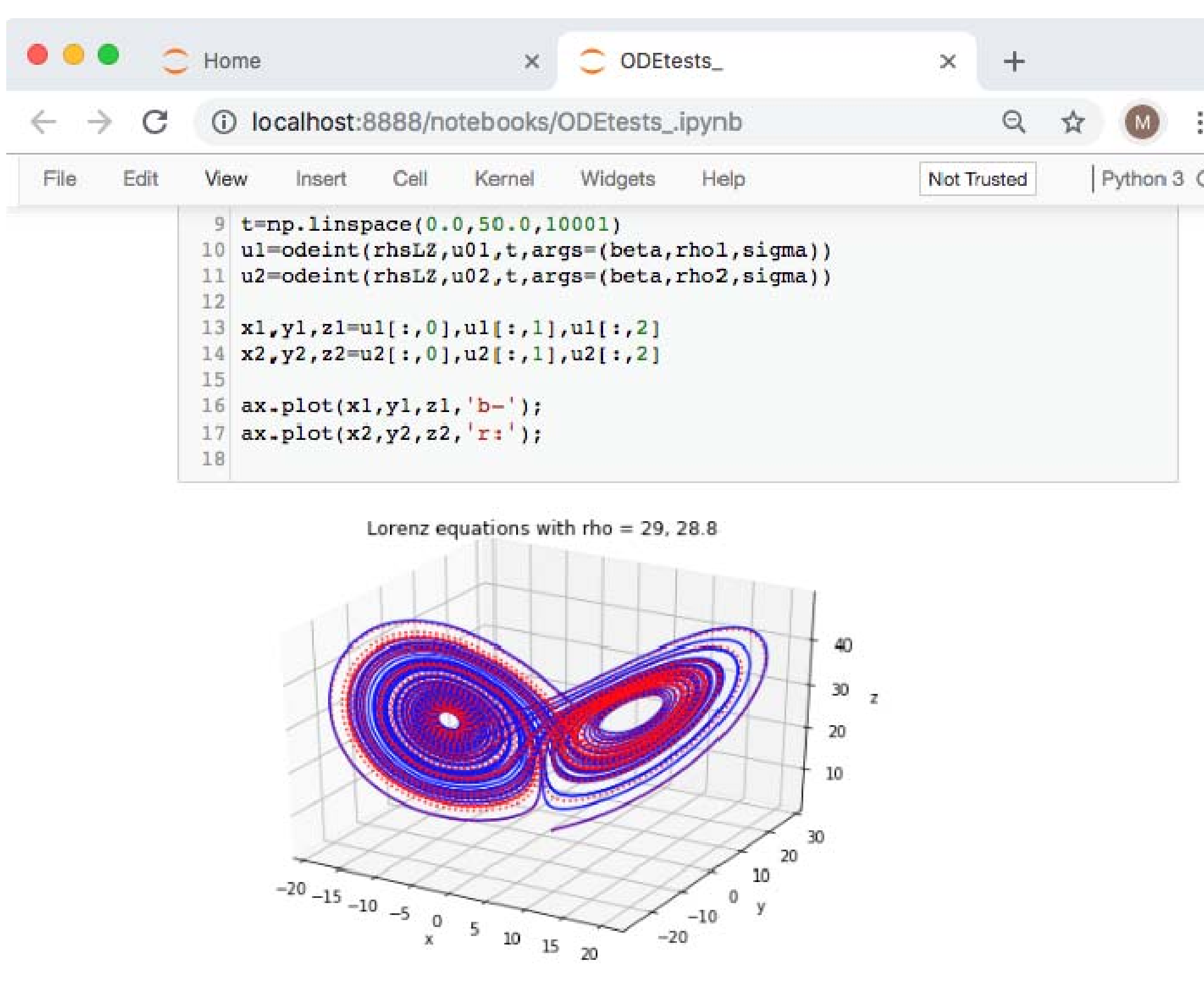


Figure 2: A notebook with live ODE simulation and complex graphics.

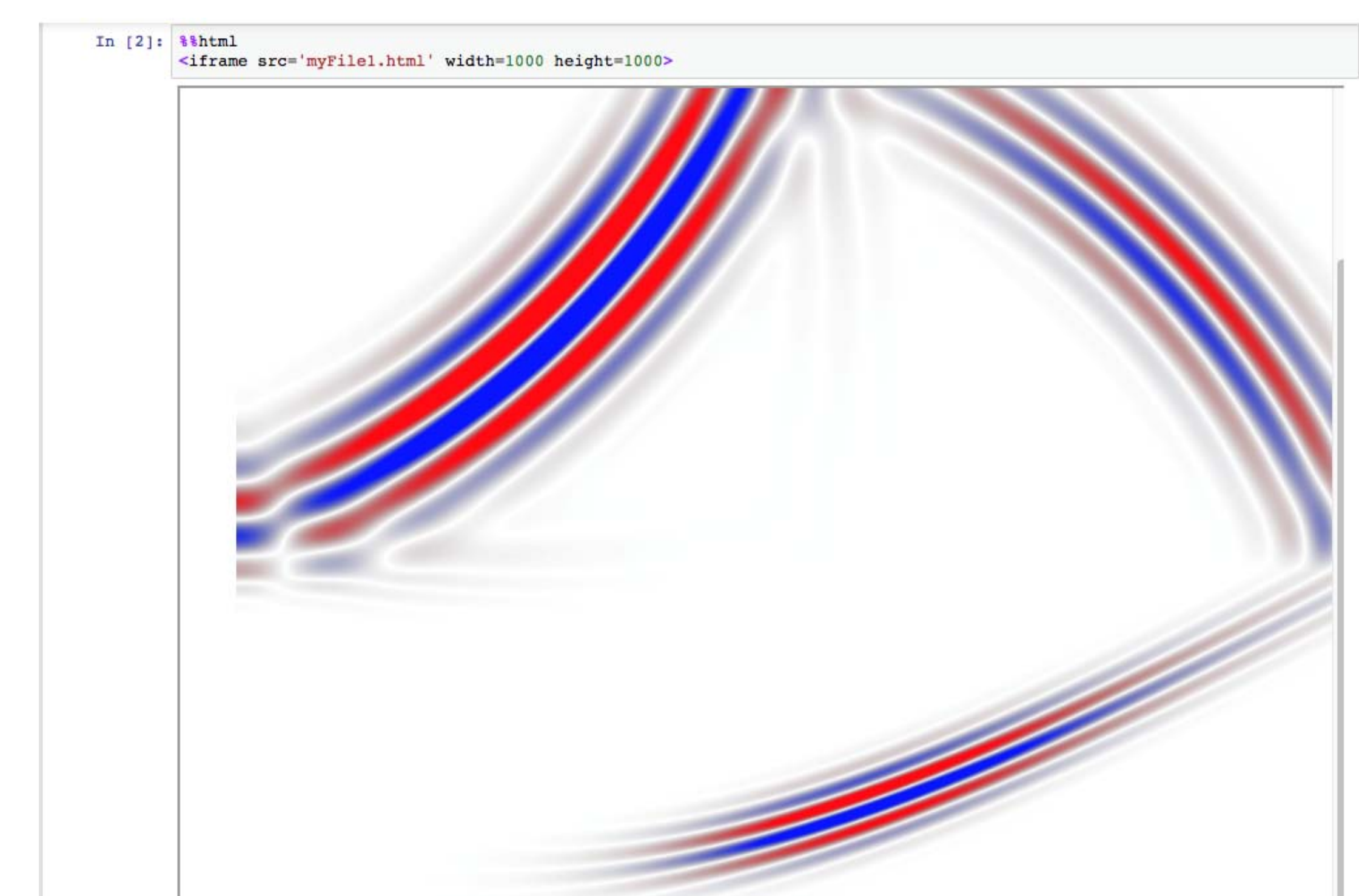


Figure 3: GPU implementation of wave propagation, in a Jupyter notebook.

Loop size	Matlab - R2018a	C - optimized	AMD	Nvidia	Speed up
20000	68 s	42.69 s	6.259 s	2.263 s	30x
40000	137 s	85.37 s	11.919 s	4.285 s	31x
60000	206 s	129.57 s	17.957 s	6.725 s	31x
80000	275 s	180.82 s	23.844 s	8.873 s	31x
100000	340 s	229.22 s	29.514 s	11.012 s	31x
slope	3.41 ms	2.34 ms	.306 ms	.109 ms	31x

Table 4: Timing tests comparing Matlab, C, and two GPU implementations.

In Table 4, we observe that 100,000 FD updates of the full one million point grid takes only 11 seconds on the Nvidia graphics card via the WebGL API, a speed up by a factor of 31 over optimized Matlab code.

Open source software use

There is a large community of scientists, researchers, and programmers who contribute algorithms, methods, and codes for free use by anyone. Unlike classic references such as “Numerical Recipes in C,” the open source community provides unrestricted licenses for use of their code by any purpose, by anyone. Some interesting examples of open source efforts are the GNU Scientific Libraries, Numpy, SciPy, and the text “Python for Scientists.” The CREWES toolbox at UCalgary, Seismic Julia from SAIG at UAlberta, and the SLIM tools at UBC are all seismic utilities accessible through Jupyter notebooks.

Future Work

We intend to extend our use of Jupyter notebooks to all our research and teaching. For the FD code on GPU, the next goals are to:

- Implement absorbing boundary conditions, perfectly matched layers for simulations suitable to seismic imaging.
- Implement in three dimensions with elastic wave equations, taking advantage of the GPU speedup.
- Develop APIs to allow the seismic researcher to access any GPU resources available on a system’s video card, bypassing major technical details. Test these APIs in real seismic data processing.

Conclusions

A Jupyter notebook is a valuable tool for computation, communication, and collaboration. In this article, we discussed our experience in using the notebooks with a particular example developing a Jupyter notebook to access a video card GPU for a 30x speedup of an acoustic wave simulation. We presented information on resources available to the reader to begin making the transition to Jupyter notebooks for their own research.

Acknowledgements

We thank the sponsors of CREWES for their support. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used in this research. We also acknowledge support from NSERC through grant CRDPJ 461179-13, Discovery Grant for the 1st author.

References

1. Lamoureux, M. P., 2016, Introduction to Syzygy: PIMS.
2. Perkel, J. M., 2018, Why Jupyter is data scientists’ computational notebook of choice: Nature, 563, 145–146.
3. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 2007, Numerical Recipes. Cambridge University Press.
4. Shreiner, D., Woo, M., Neider, J., and Davis, T., 2004, OpenGL Programming Guide, Fourth Edition: Addison-Wesley.
5. Stewart, J. M., 2014, Python for Scientists: Cambridge University Press.