Quark-Gluon Plasma Physics

Jupyter Notebooks

Prof. Dr. Klaus Reygers
Prof. Dr. Johanna Stachel
Heidelberg University
SS 2019

Jupyter Notebooks

- Everything in one document
  - Text (Markdown)
  - Equations (Latex)
  - Code
  - Figures
- Open source
- Very good documentation
- Large user community
- Python: universal glue language

https://jupyter.org/

We encourage you to **present** and **hand in** your **homework problem** as a jupyter notebook (particularly suited for problems which require numerical calculations and/or plotting)

TOOLBOX · 30 OCTOBER 2018

Why Jupyter is data scientists' computational notebook of choice

An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.

https://www.nature.com/articles/d41586-018-07196-1

jupyter Lorenz Differential Equations (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Code Cell Toolbar: None

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

In [7]:
```
interact(Lorenz, N=fixed(10), angle=(0.,360.),
         σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

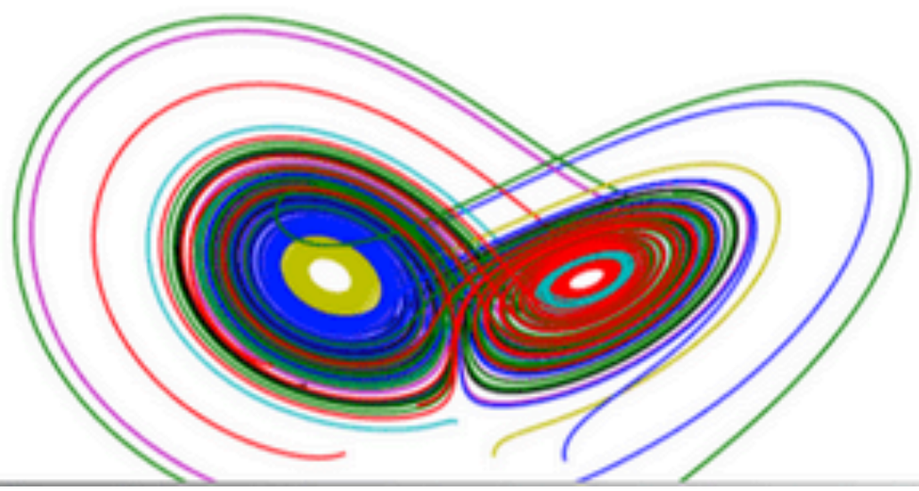
angle 308.2

max_time 12

σ 10

β 2.6

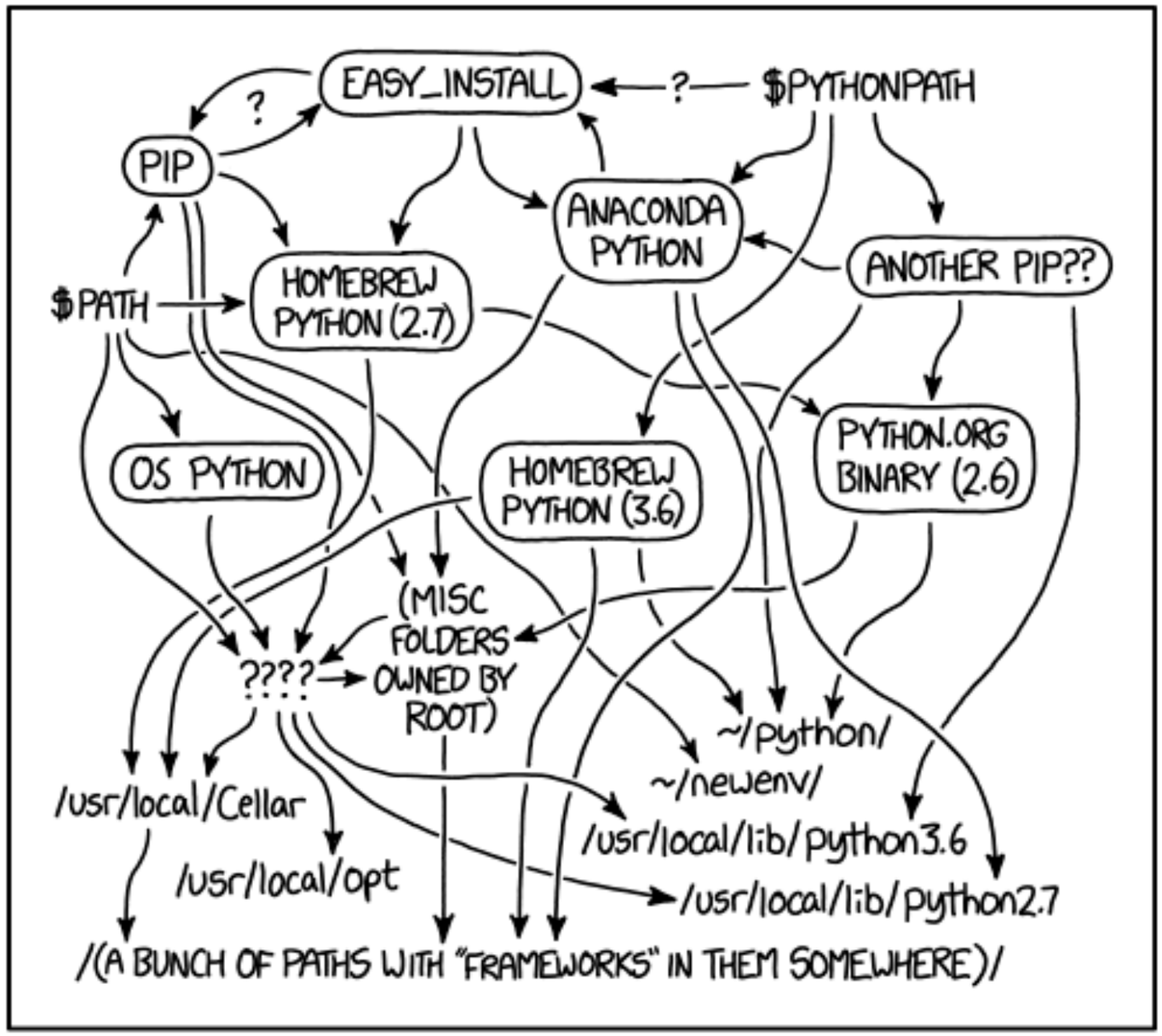
ρ 28



Installation

Python Environment:

https://xkcd.com/1987/



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

It is actually simple …

Installation

https://jupyter.readthedocs.io/en/latest/install.html

Prerequisite: Python

While Jupyter runs code in many programming languages, **Python** is a requirement (Python 3.3 or greater, or Python 2.7) for installing the Jupyter Notebook.

We recommend using the Anaconda distribution to install Python and Jupyter. We'll go through its installation in the next section.

Installing Jupyter using Anaconda and conda

For new users, we **highly recommend** installing Anaconda. Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.

Use the following installation steps:

1. Download Anaconda. We recommend downloading Anaconda's latest Python 3 version (currently Python 3.5).
2. Install the version of Anaconda which you downloaded, following the instructions on the download page.
3. Congratulations, you have installed Jupyter Notebook. To run the notebook:

```
jupyter notebook
```

See Running the Notebook for more details.

Personally, I found the installation through **pip** more flexible (requires python3 installation)

Alternative for experienced Python users: Installing Jupyter with pip

Important

Jupyter installation requires Python 3.3 or greater, or Python 2.7. IPython 1.x, which included the parts that later became Jupyter, was the last version to support Python 3.2 and 2.6.

As an existing Python user, you may wish to install Jupyter using Python's package manager, pip, instead of Anaconda.

First, ensure that you have the latest pip; older versions may have trouble with some dependencies:

```
pip3 install --upgrade pip
```

Then install the Jupyter Notebook using:

```
pip3 install jupyter
```

(Use `pip` if using legacy Python 2.)

Congratulations. You have installed Jupyter Notebook. See Running the Notebook for more details.

Installation of python 3 on a mac:

1. Install homebrew (http://brew.sh):

   ```
   $ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
   ```

2. Install python 3:

   ```
   $ brew install python
   ```

Starting the Notebook Server

After you have installed the Jupyter Notebook on your computer, you are ready to run the notebook server. You can start the notebook server from the command line (using Terminal on Mac/Linux, Command Prompt on Windows) by running:

```
jupyter notebook
```

This will print some information about the notebook server in your terminal, including the URL of the web application (by default, `http://localhost:8888`):

```
$ jupyter notebook
[I 08:58:24.417 NotebookApp] Serving notebooks from local directory: /Users/catherine
[I 08:58:24.417 NotebookApp] 0 active kernels
[I 08:58:24.417 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 08:58:24.417 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to
```

How do I open a specific Notebook?

The following code should open the given notebook in the currently running notebook server, starting one if necessary.

```
jupyter notebook notebook.ipynb
```

A very simple example

Rapidity vs longitudinal velocity

Used plotting libraries:

```
1 from math import *
2 import numpy as np
3 import matplotlib.pyplot as plt
```

Definition of rapidity:

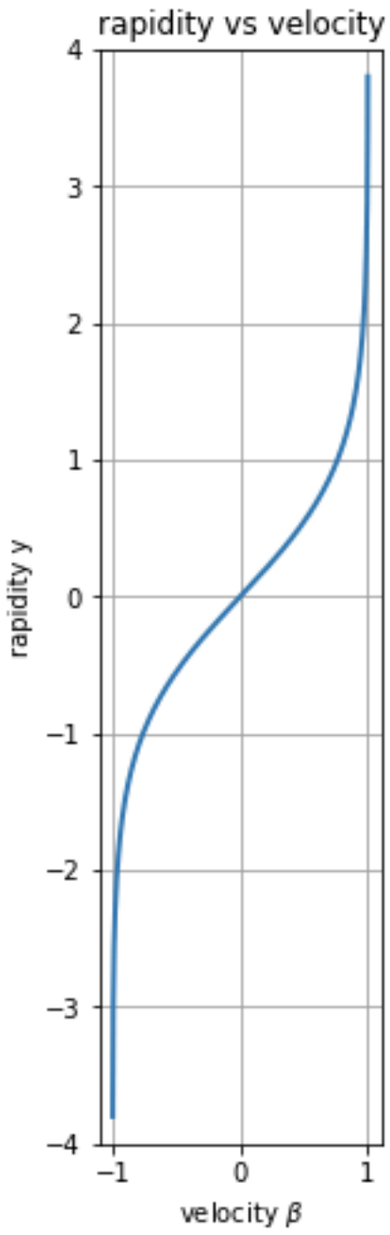
$$y = \frac{1}{2}\ln\frac{E + p_z}{E - pZ} = \frac{1}{2}\ln\frac{1 + \beta_z}{1 - \beta_z} = \text{arctanh}\beta_z$$

Define arrays for longitudinal rapidity β and rapidity y

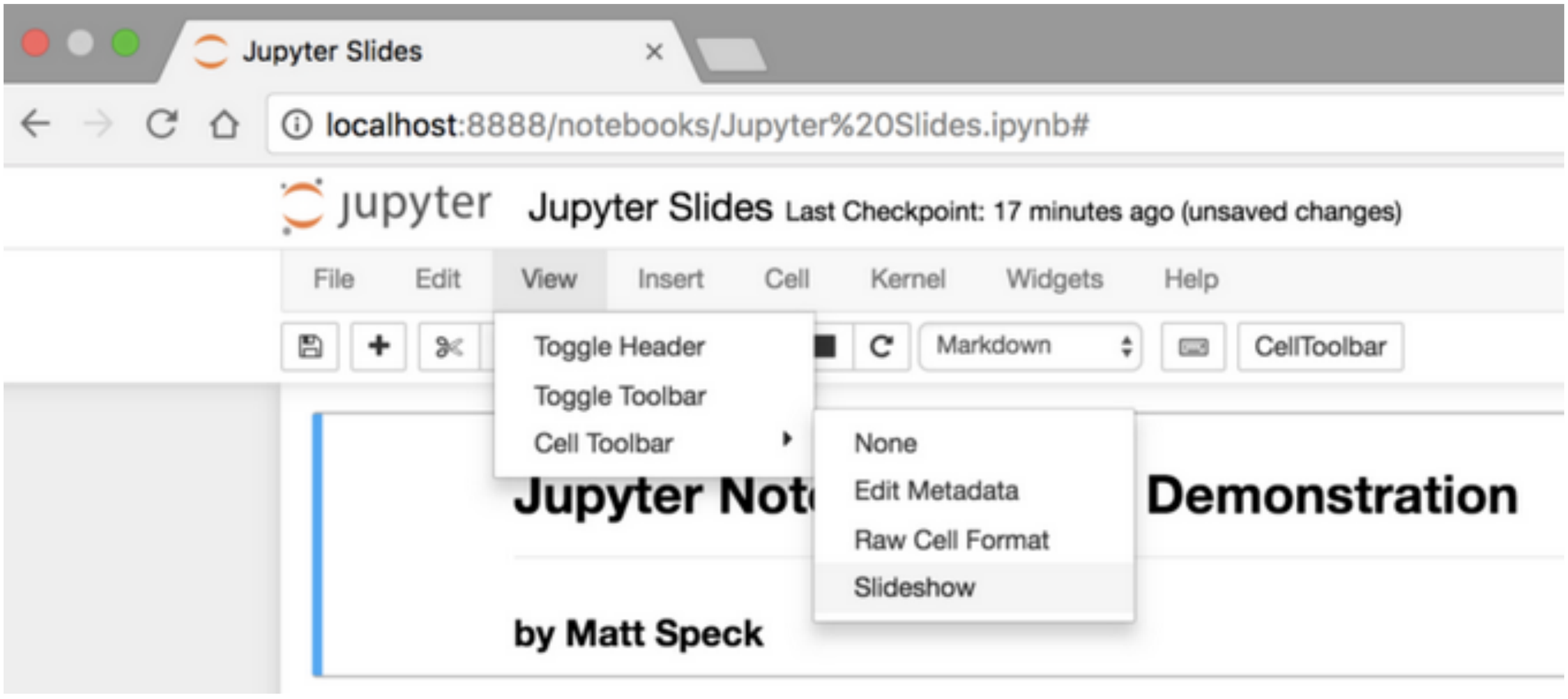
```
1 beta = np.linspace(-0.999, 0.999, 3001)
2 rap = np.arctanh(beta)
```

A very simple example

```
plt.figure(figsize=(2,8))
plt.xlabel(r'velocity $\beta$')
plt.ylabel('rapidity y')
plt.title('rapidity vs velocity')
plt.plot(beta, rap, linewidth=2.0)
plt.axis([-1.1,1.1,-4,4])
plt.grid(b=True, which='both', color='0.65',linestyle='-')
plt.show()
```

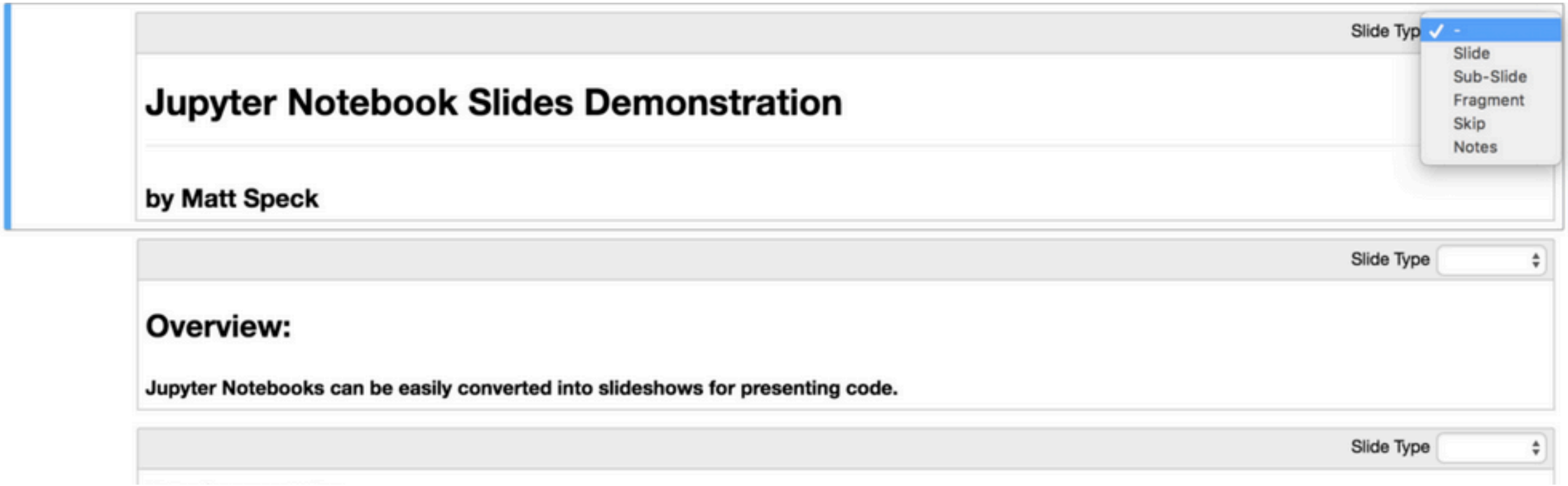


Making a slideshow presentation from your notebook



https://medium.com/@mjspeck/presenting-code-using-jupyter-notebook-slides-a8a3c3b59d67

Making a slideshow presentation from your notebook



Converting a notebook to a html presentation:

```
jupyter nbconvert rapidity_vs_beta.ipynb --to slides --post serve
```