Bokeh: A Python Plotting Library for the Web Browser

Presentation at Chair of Structural Mechanics, BGU, TUM

Benjamin Rüth

January 25, 2017

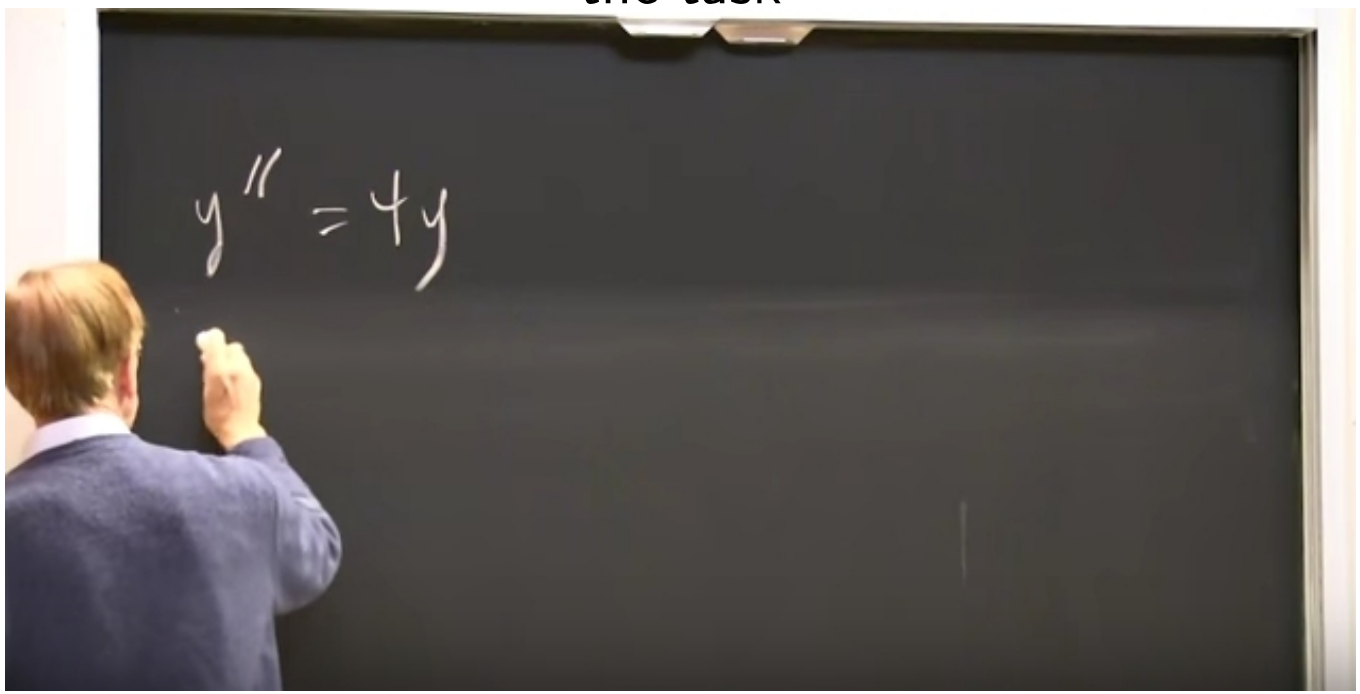
Table of Contents

1. Why do we need web visualization?
2. Interactive Web Visualization
   - What do we want?
   - Setup and Tools
3. Bokeh
   - Introduction to Bokeh
   - Example: Phase Plane Pictures
   - What do we get?
4. Collaborating

Why do we need web visualization?

Phase Plane Pictures: Source, Sink, Saddle
Gilbert Strang, MIT[1]

the task

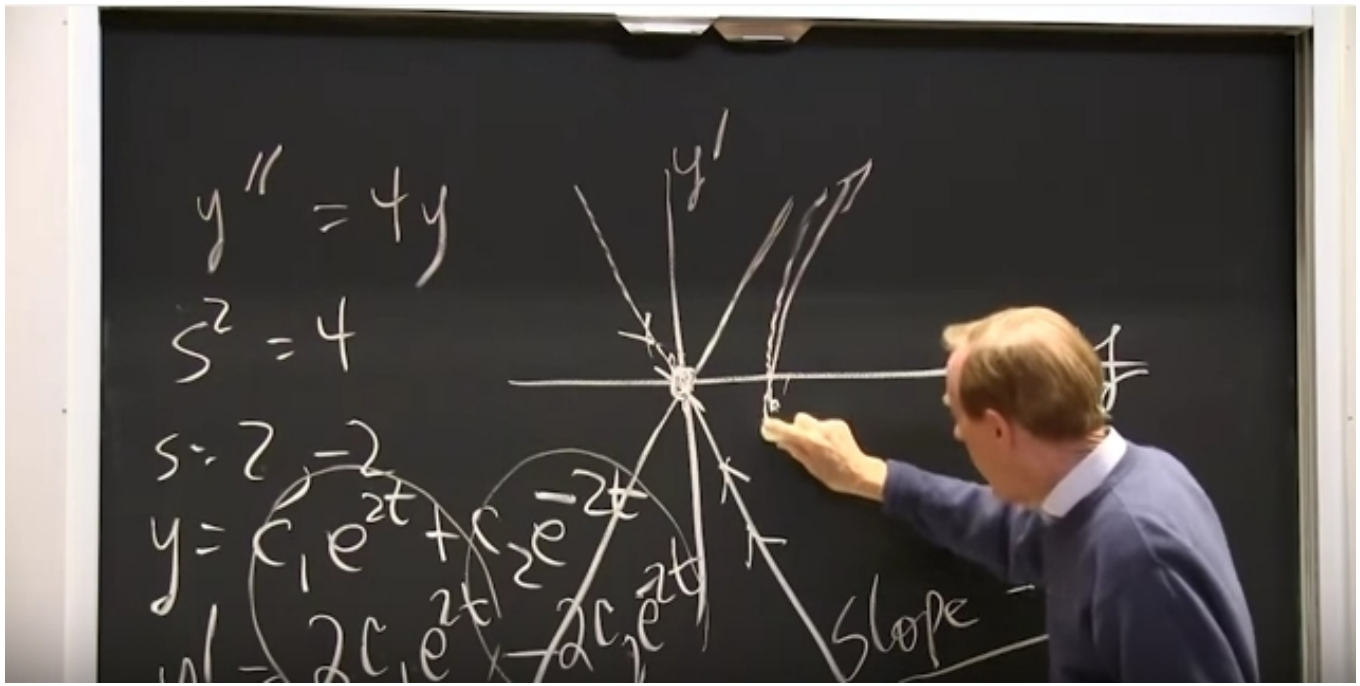


[1] https://www.youtube.com/watch?v=VqXKa11IA6A

Why do we need web visualization?

Phase Plane Pictures: Source, Sink, Saddle
Gilbert Strang, MIT[1]

3 minutes later

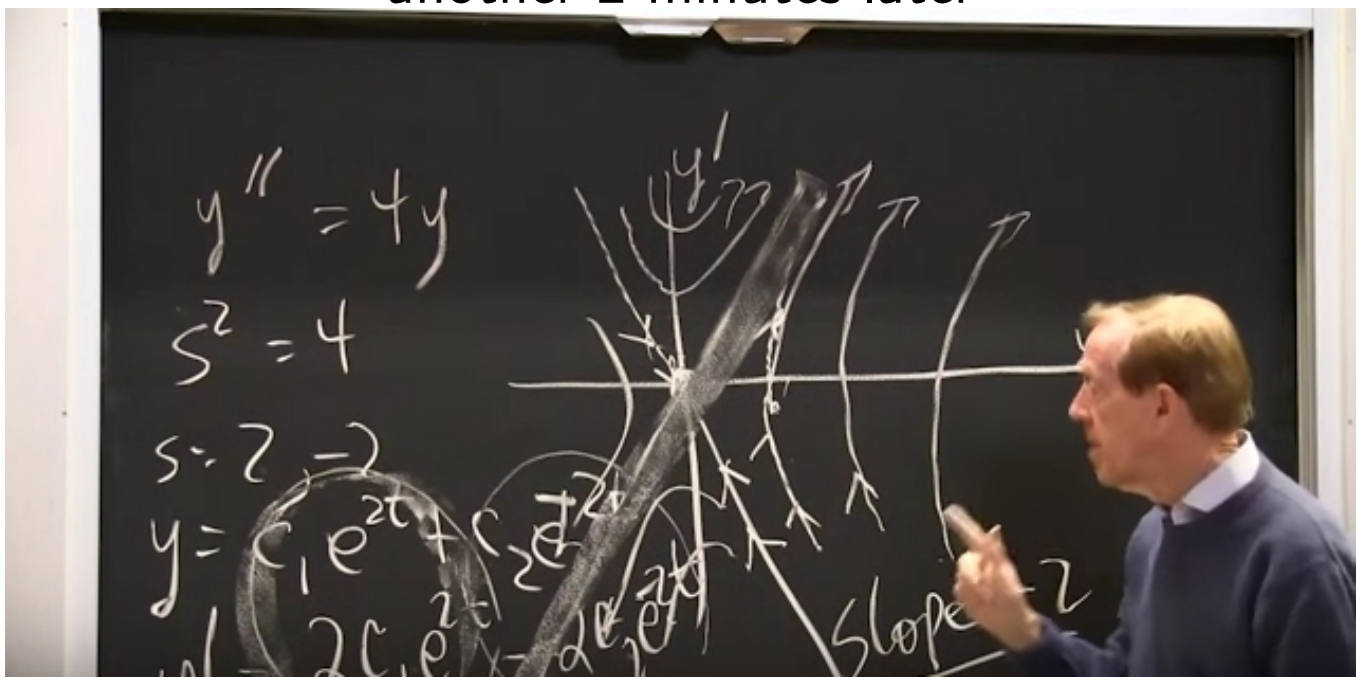


[1] https://www.youtube.com/watch?v=VqXKa11IA6A

Why do we need web visualization?

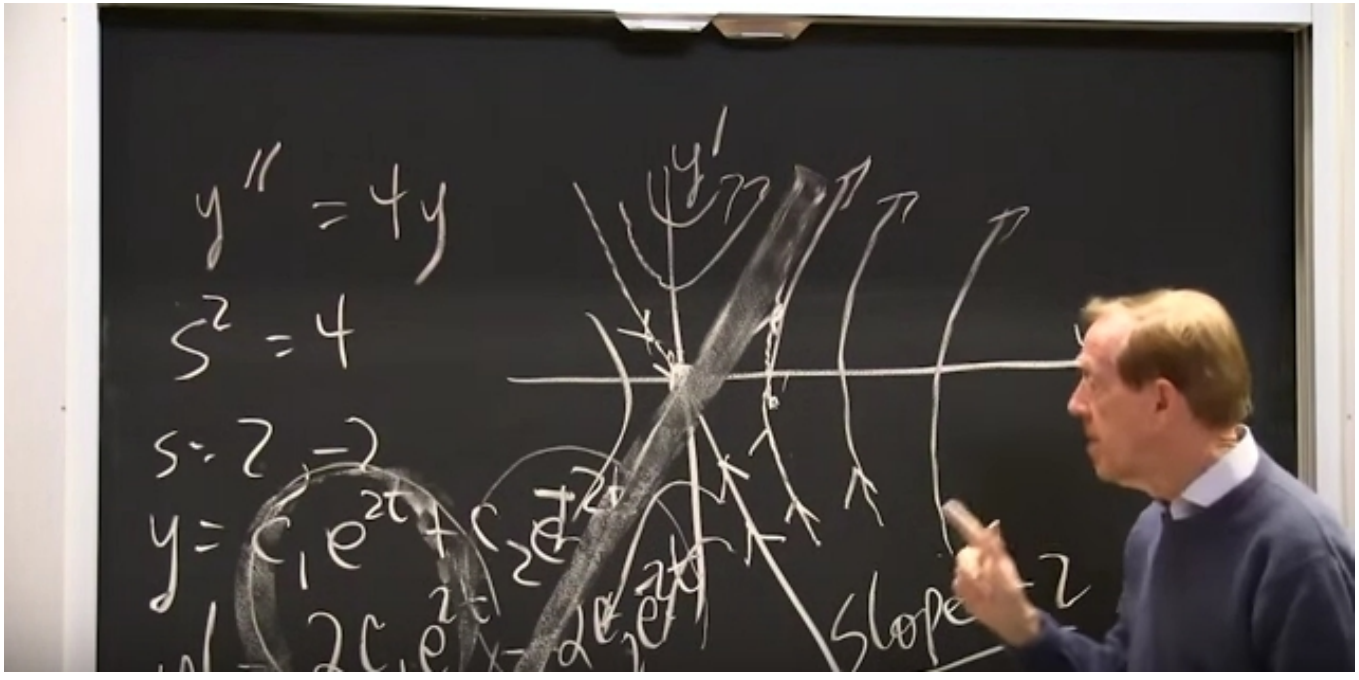
Phase Plane Pictures: Source, Sink, Saddle
Gilbert Strang, MIT[1]

another 2 minutes later



[1] https://www.youtube.com/watch?v=VqXKa11IA6A

Why do we need web visualization?



Problem with Videos and Lectures

- Advisor needed
- Time intensive
- Not interactive (videos)
- Not individual

Table of Contents

1. Why do we need web visualization?
2. Interactive Web Visualization
   - What do we want?
   - Setup and Tools
3. Bokeh
   - Introduction to Bokeh
   - Example: Phase Plane Pictures
   - What do we get?
4. Collaborating

What do we want?

Our goal

- Visualization of math & mechanics content
- Easy-to-use, flexible tool
- For use at home and in lectures

What do we want?

Our goal

- Visualization of math & mechanics content
- Easy-to-use, flexible tool
- For use at home and in lectures

User Constraints

- No programming experience required
- No special tools required

What do we want?

Our goal

- Visualization of math & mechanics content
- Easy-to-use, flexible tool
- For use at home and in lectures

User Constraints

- No programming experience required
- No special tools required

Development Constraints

- Easy to implement and understand
- Support for scientific applications

Setup and Tools

Installation

- **Version Control:** Git

  `https://git-scm.com/downloads`

- **Python:** Anaconda

  `https://www.continuum.io/downloads`

- **IDE:** install PyCharm

  `https://www.jetbrains.com/pycharm/`

Git: Working on software in a team

- `git clone` for downloading the source
- `git pull/push` for syncing with the repository
- `git add/commit` for contributing
- `git branch` for experiments

Git: Working on software in a team

- `git clone` for downloading the source
- `git pull/push` for syncing with the repository
- `git add/commit` for contributing
- `git branch` for experiments

Our repository:

- Sign up on `https://github.com/`
- `https://github.com/BenjaminRueth/Visualization`
- **HTTPS:**

```
git clone https://github.com/BenjaminRueth/Visualization.git
```

- **SSH:**[2]

```
git clone git@github.com:BenjaminRueth/Visualization.git
```

Setup and Tools

Anaconda: A Python distribution

- **Anaconda prompt:** `conda install bokeh`
- **Python prompt:** `ipython` or `python`

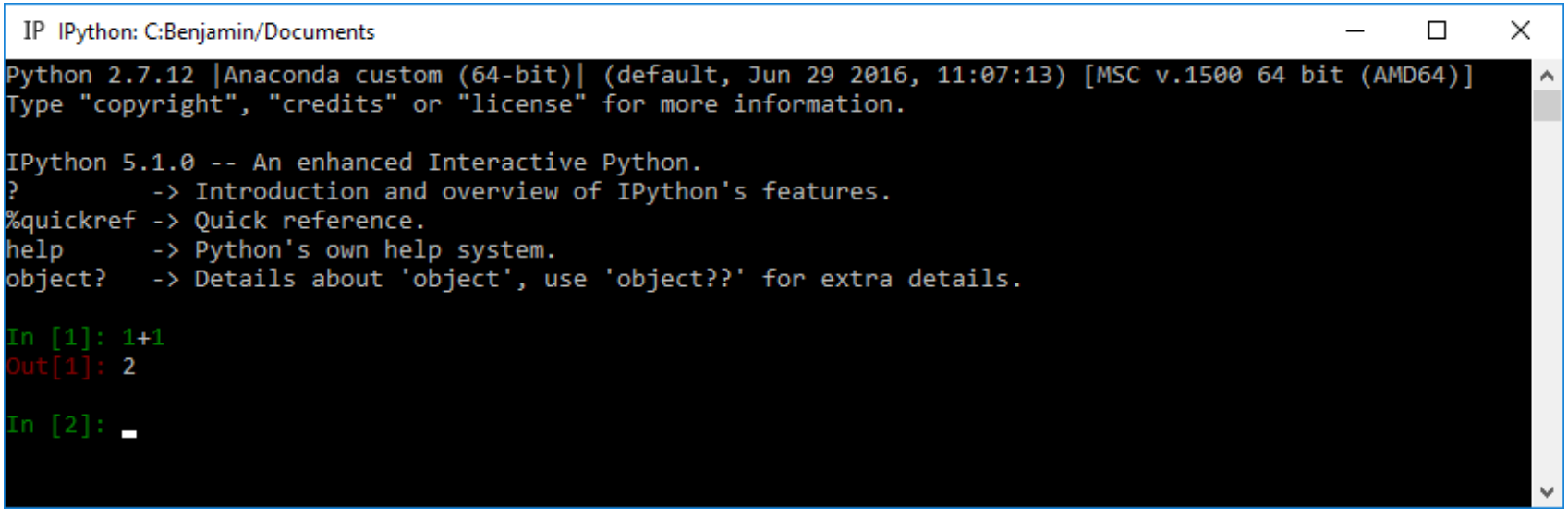


Table of Contents

1. Why do we need web visualization?
2. Interactive Web Visualization
   - What do we want?
   - Setup and Tools
3. Bokeh
   - Introduction to Bokeh
   - Example: Phase Plane Pictures
   - What do we get?
4. Collaborating

Introduction to Bokeh

- Python plotting library (similar to matplotlib)
- uses the webbrowser for displaying graphics
- uses `D3.js`
- open source
- visit `http://bokeh.pydata.org`

Necessary preparations

- **Clone the repository:**
  use Git.
- **Our working directory:**
  `...\Visualization\Presentation\Examples`
- **Start a shell:**
  - **Windows:** `start+r`, then type cmd
  - **Linux:** `ctrl+alt+T`
  - with `cd <path>` navigate to the working directory

Static example: Plotting in the browser

- Fire up a shell in your working directory
- Start with `python staticPlotting.py`
- The browser should show the plot

Introduction to Bokeh

Static example: Plotting in the browser

```
import numpy as np
from bokeh.plotting import figure, show, output_file

# create data
x = np.linspace(0, 4*np.pi, 100)
y = np.sin(x)
# define tools
TOOLS = "pan,wheel_zoom,box_zoom,reset,save,box_select"
# plot data
p1 = figure(title="Legend Example", tools=TOOLS)
p1.circle(x,    y, legend="sin(x)")
p1.circle(x, 2*y, legend="2*sin(x)", color="orange")
p1.circle(x, 3*y, legend="3*sin(x)", color="green")
# save and open plot
output_file("legend.html", title="legend.py example")
show(p1)
```

Server example: An interactive function plotting tool

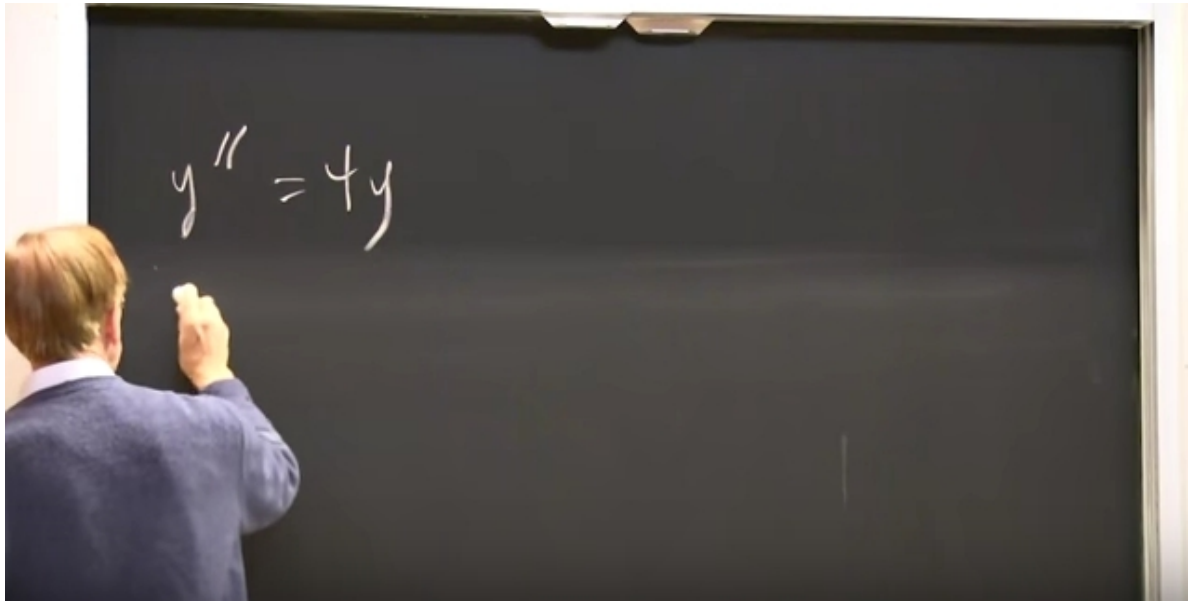
- Fire up a shell in your working directory
- Start with `bokeh serve functionPlotter.py`
- Visit `http://localhost:5006/functionPlotter`

Server example: An interactive function plotting tool

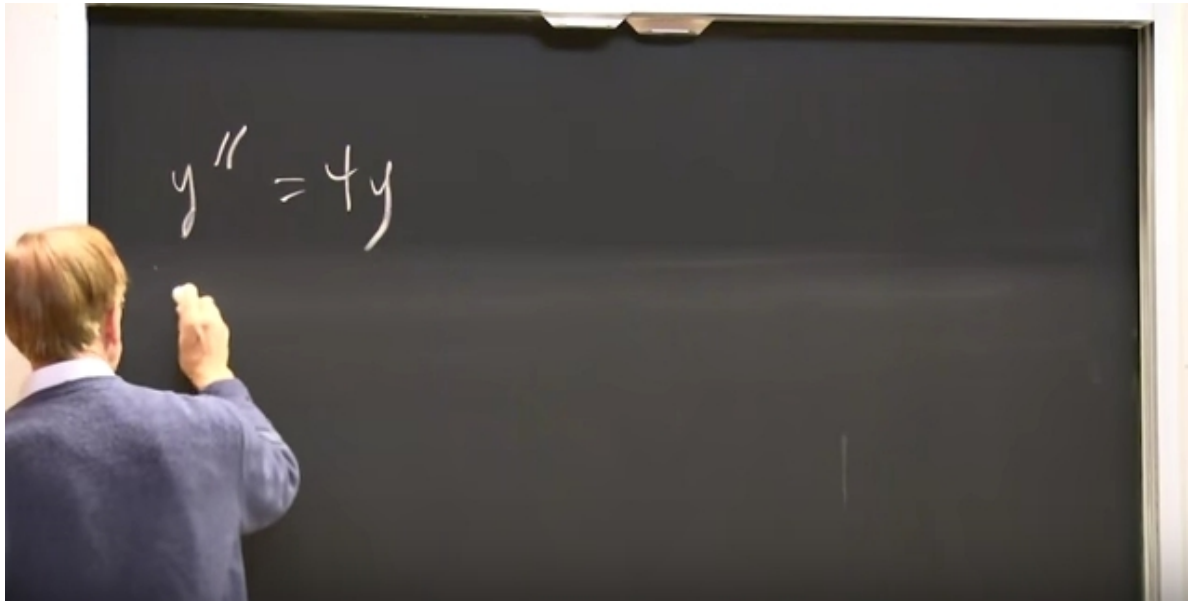
- Fire up a shell in your working directory
- Start with `bokeh serve functionPlotter.py`
- Visit `http://localhost:5006/functionPlotter`
- only 60 LoC!
- uses `numpy` and `scipy`

```
from bokeh import ...
import numpy as np
from sympy import sympify, lambdify, diff
...
# Data
line_source = ColumnDataSource(data=dict(x=[], y=[]))

# Controls
f_input = TextInput(value="sin(x)", title="f(x):")
derivative_input = Slider(title="n", value=1.0, start=0.0, en
...
# Plotting
plot = Figure(...)
plot.line(x='x', y='y', source=line_source, ...)
plot.line(x='x', y='dy', source=line_source, ...)

#Callback
f_input.on_change('value', fun_change)
derivative_input.on_change('value', fun_change)

#Layout
curdoc().add_root(row(plot,column(f_input,derivative_input)))
```

Example: Phase Plane Pictures



Example: Phase Plane Pictures

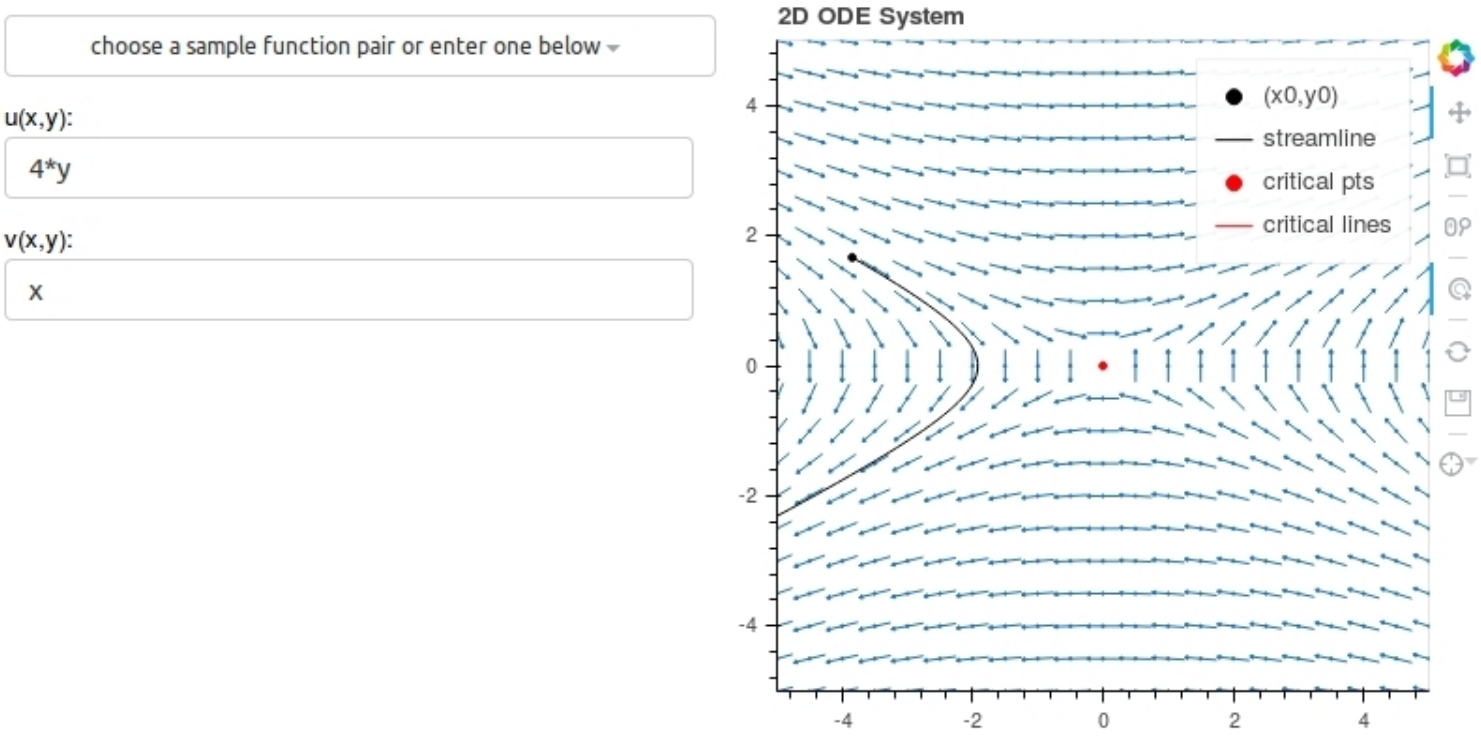


2D ODE system

$$y'' = u(x, y) = 4y$$
$$y' = v(x, y) = x$$

Visualization on `http://localhost:5006/odesystem_app`

What do we get?



Our goal

- ✓ Visualization of math (& mechanics?) content
- ✓ Easy-to-use, flexible tool
- ✓ For use at home and in lectures

What do we get?

User Constraints

- ✓ No programming experience required
- ✓ No special tools required

Development Constraints

- ✓ Easy to implement and understand
- ✓ Support for scientific applications

BUT

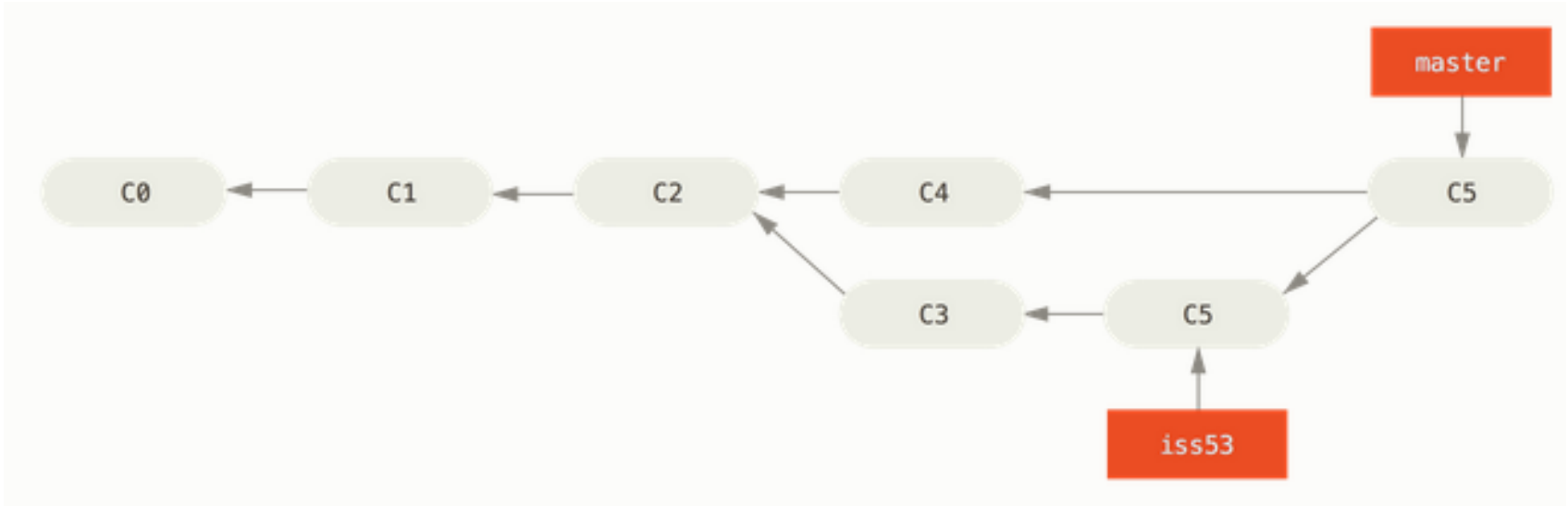
- we need a server
- the user needs an internet connection
- we have to transfer data
- sometimes slow

Table of Contents

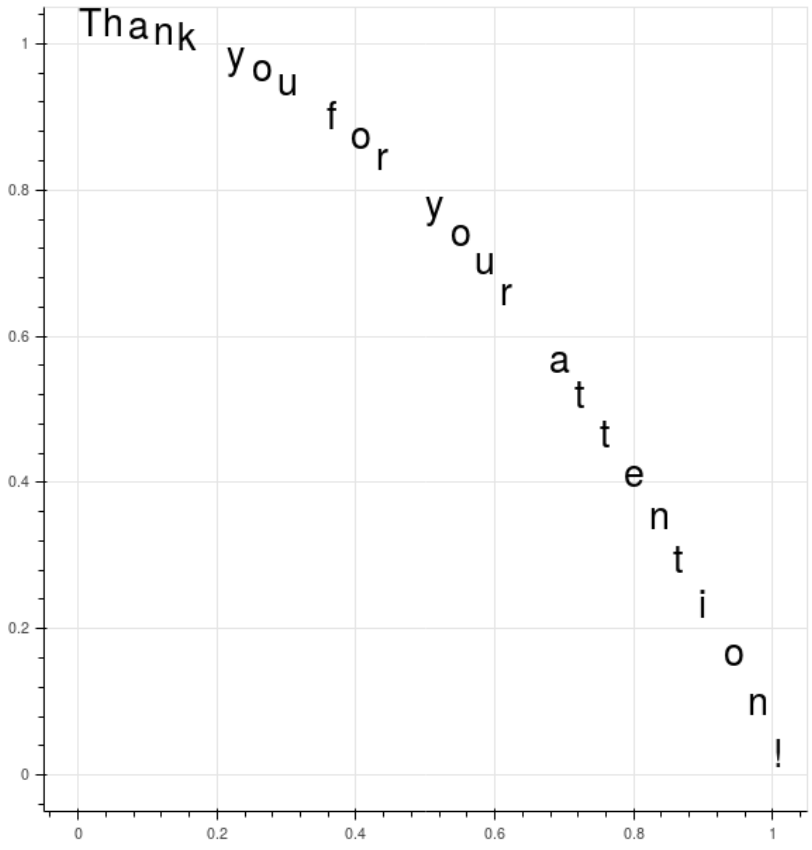
1. Why do we need web visualization?
2. Interactive Web Visualization
   - What do we want?
   - Setup and Tools
3. Bokeh
   - Introduction to Bokeh
   - Example: Phase Plane Pictures
   - What do we get?
4. Collaborating

Use Git

- **Create a branch:** `git checkout -b <myName>`
- **Develop:** e.g. change something in `Visualization/Example/staticPlotting.py`
- **Finalize:** `git add`, `git commit -m "..."`
- **Synchronize:** `git push`[3]
- **Merge:** `git checkout master`, `git merge <myName>`



[3] probably you get the hint to type:
`git push --set-upstream origin <myName>` just do it.




```
import numpy as np
from bokeh.plotting import \
    figure, \
    output_file

thanks = "Thank you for your attention!"
# create data
t = list(thanks)
x = np.linspace(0, 1, t.__len__())
y = 1-x**2

# plot data
p1 = figure()
p1.text(x,y,t, text_font_size="20pt")
# save and open plot
output_file("thanks.html")
```