

# Bokeh: A Python Plotting Library for the Web Browser

Presentation at BGCE Research Day, TUM

Benjamin R  th

December 7, 2016

# Table of Contents

## ① Why do we need web visualization?

## ② Interactive Web Visualization

What do we want?

Bokeh

Example: Phase Plane Pictures

What do we get?

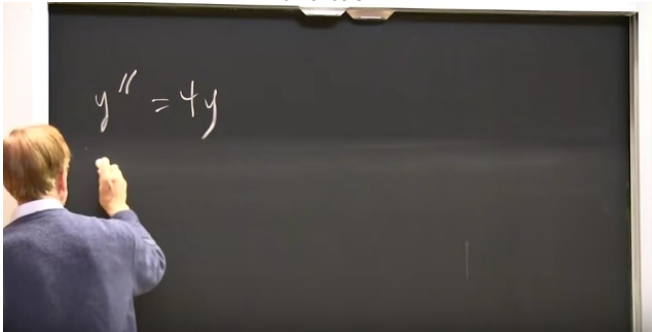
## ③ Mini Pro & Con

# Why do we need web visualization?

## Phase Plane Pictures: Source, Sink, Saddle

Gilbert Strang, MIT<sup>1</sup>

the task



---

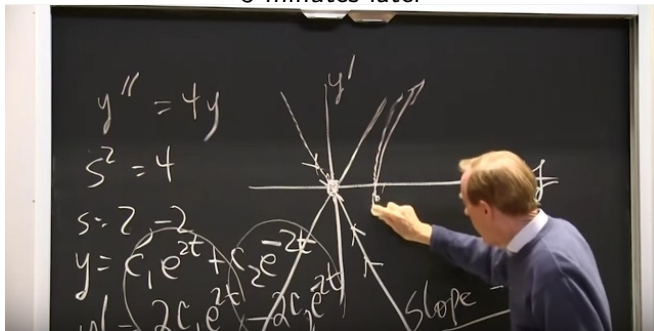
<sup>1</sup><https://www.youtube.com/watch?v=VqXKa11IA6A>

# Why do we need web visualization?

## Phase Plane Pictures: Source, Sink, Saddle

Gilbert Strang, MIT<sup>1</sup>

3 minutes later



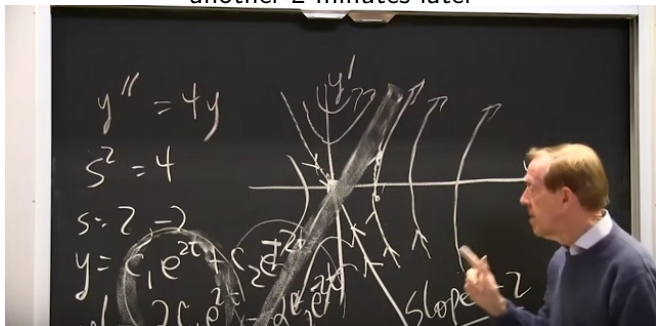
<sup>1</sup><https://www.youtube.com/watch?v=VqXKa11IA6A>

# Why do we need web visualization?

## Phase Plane Pictures: Source, Sink, Saddle

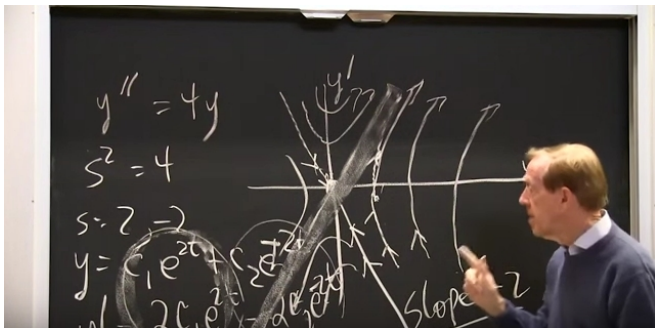
Gilbert Strang, MIT<sup>1</sup>

another 2 minutes later



<sup>1</sup><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?

- Bokeh

- Example: Phase Plane Pictures

- What do we get?

## ③ Mini Pro & Con

# What do we want?

## Our goal

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



# What do we want?

## Our goal

- Visualization of math 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 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

- ics

# Bokeh

## static 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

- Start with `bokeh serve functionPlotter.py`
- Visit `http://localhost:5006/functionPlotter`

## Server example: An interactive function plotting tool

- Start with `bokeh serve functionPlotter.py`
- Visit `http://localhost:5006/functionPlotter`
- only 60 LoC!
- uses `numpy` and `scipy`

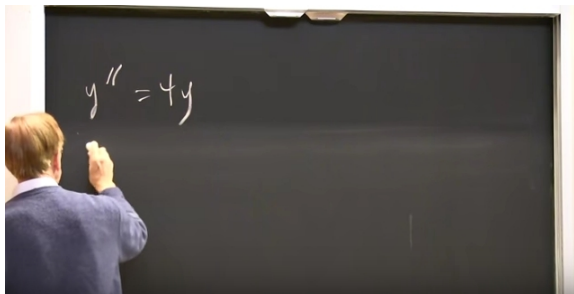
```
from bokeh import ...
import numpy as np
from sympy import sympify, lambdify, diff
...
# Data
line_source = ColumnDataSource(data=dict(x=[], y=[]))

# Controls
f_input = TextInput(value="sin(x)", title="f(x):")
derivative_input = Slider(title="n", value=1.0, start=0.0, end=5.0)
...
# 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)

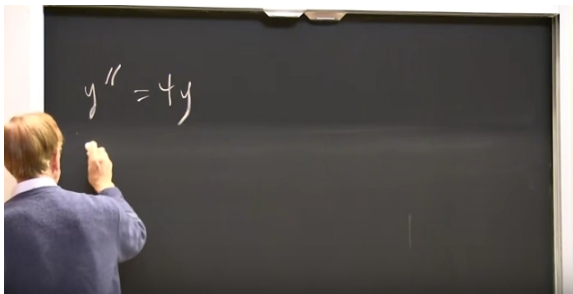
#Layout
curdoc().add_root(row(plot, column(f_input, derivative_input)))
```

## Example: Phase Plane Pictures





## Example: Phase Plane Pictures



2D ODE system

$$\begin{aligned}y'' &= u(x, y) = 4y \\ y' &= v(x, y) = x\end{aligned}$$

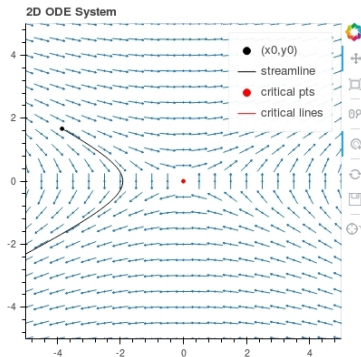
Visualization on [http://localhost:5006/odesystem\\_app](http://localhost:5006/odesystem_app)

# What do we get?

choose a sample function pair or enter one below ▾

$u(x,y):$

$v(x,y):$



## Our goal

- ✓ Visualization of math 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?

Bokeh

Example: Phase Plane Pictures

What do we get?

## ③ Mini Pro & Con

## Mini Pro & Con

Go to lectures or use fancy tools?

## Mini Pro & Con

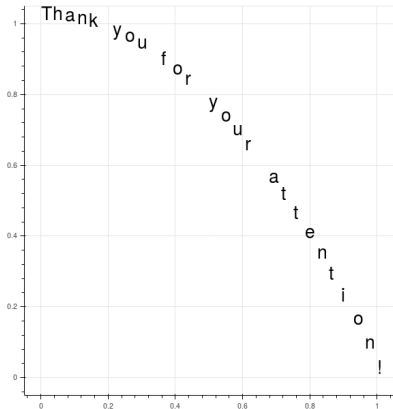
Go to lectures or use fancy tools?

### Lecture

- ✓ expert knowledge
- ✓ ask questions
- ✓ interpretation of results
- ✗ sit & relax
- ✗ asking questions...

### Fancy tools

- ✓ individual use
- ✓ check results
- ✓ explore algorithms
- ✗ just pictures
- ✗ no feedback
- ✗ costly



```
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")
```