

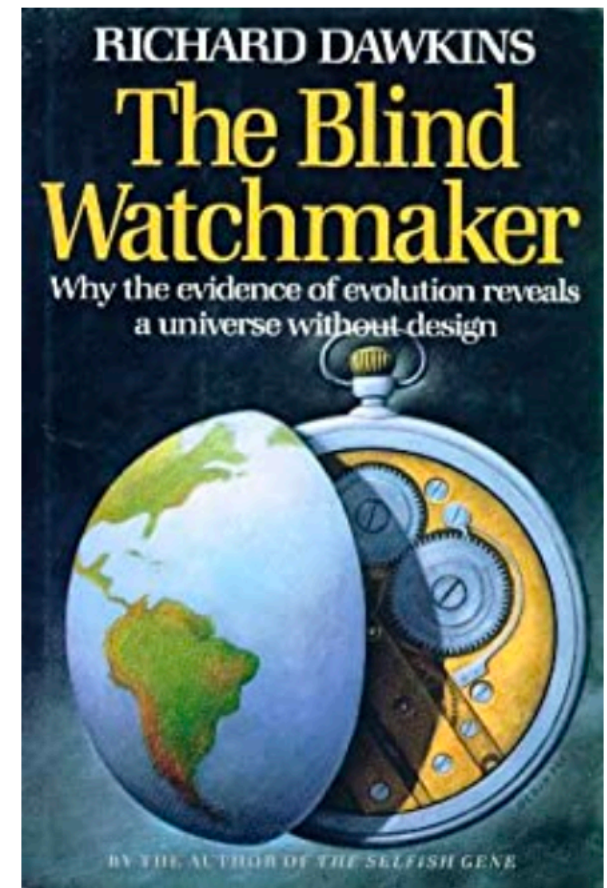
# CSE/ECE 848 Introduction to Evolutionary Computation

## Module 2 - Lecture 5 - Part 3 Octave Code for “The Blind Watchmaker”

Wolfgang Banzhaf, CSE  
John R. Koza Chair in Genetic Programming

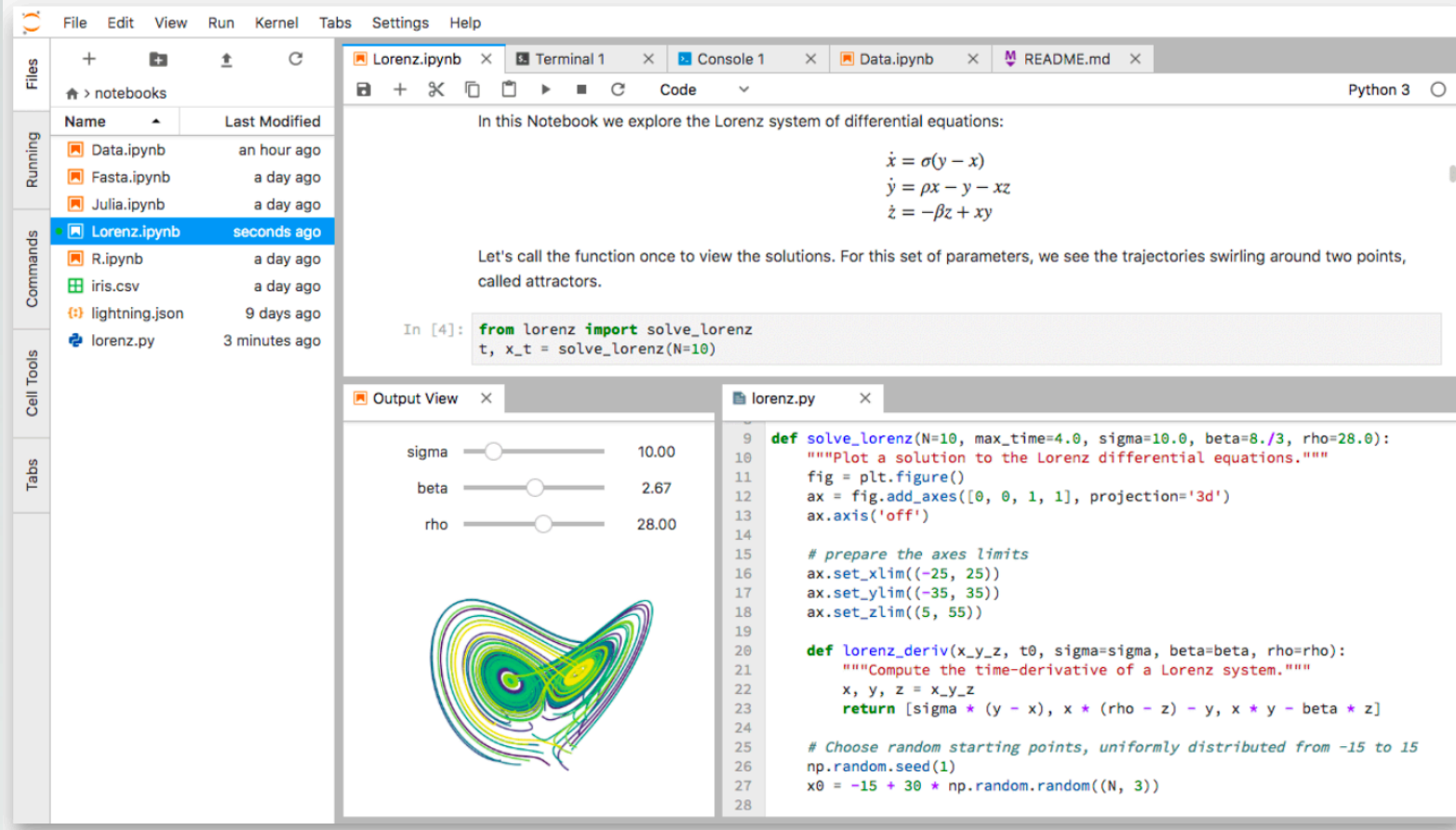
# The Blind Watchmaker

- Richard Dawkins' book with the same title has an interesting demonstration that argues for the way evolution works.
- This goes back to the argument between proponents of natural evolution and proponents of “intelligent design”.
- It asks whether anything as complex as Shakespearean sonnets (standing in for a complex natural being) could have been created by randomness.
- The answer is “yes”, by cumulative selection.



# JupyterLab Documentation

JupyterLab is the next-generation web-based user interface for Project Jupyter. [Try it on Binder](#).  
JupyterLab follows the Jupyter [Community Guides](#).



The screenshot displays the JupyterLab web interface. On the left, a sidebar shows a file browser with a list of notebooks and files, including 'Data.ipynb', 'Fasta.ipynb', 'Julia.ipynb', 'Lorenz.ipynb' (selected), 'R.ipynb', 'iris.csv', 'lightning.json', and 'lorenz.py'. The main area is divided into three panes. The top pane shows the 'Lorenz.ipynb' notebook with text explaining the Lorenz system and its equations:  $\dot{x} = \sigma(y - x)$ ,  $\dot{y} = \rho x - y - xz$ , and  $\dot{z} = -\beta z + xy$ . Below this, a code cell is shown with the following code: 

```
In [4]: from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
```

. The bottom-left pane, labeled 'Output View', displays a 3D plot of the Lorenz attractor with sliders for parameters sigma (10.00), beta (2.67), and rho (28.00). The bottom-right pane, labeled 'lorenz.py', shows the Python code for solving the Lorenz system, including a function 'solve\_lorenz' and a function 'lorenz\_deriv'.