

Free Resources and Curricula for Teaching Scientific Computing:

DSECOP, PICUP, PY4E



BRIDGEWATER
COLLEGE

Deva O'Neil, Bridgewater, VA
doneil@bridgewater.edu

April 2023

Scientific Computing in a Physics Curriculum

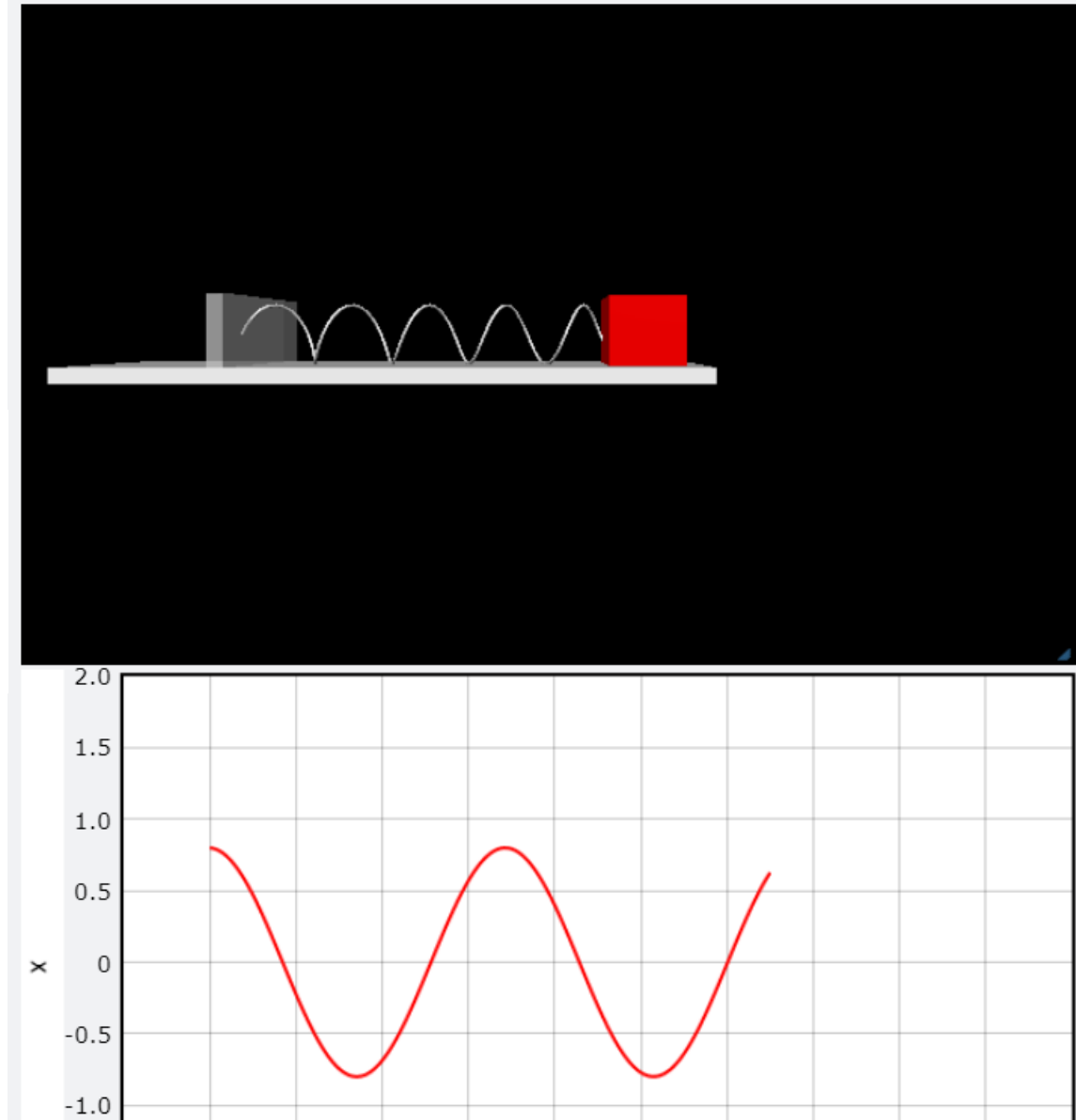
- Part of a regular physics course
- Stand-alone physics course (Computational Physics)
- Separate course requirement (Introductory Programming)
- Research projects

What does
scientific
computing
with free
software look
like?

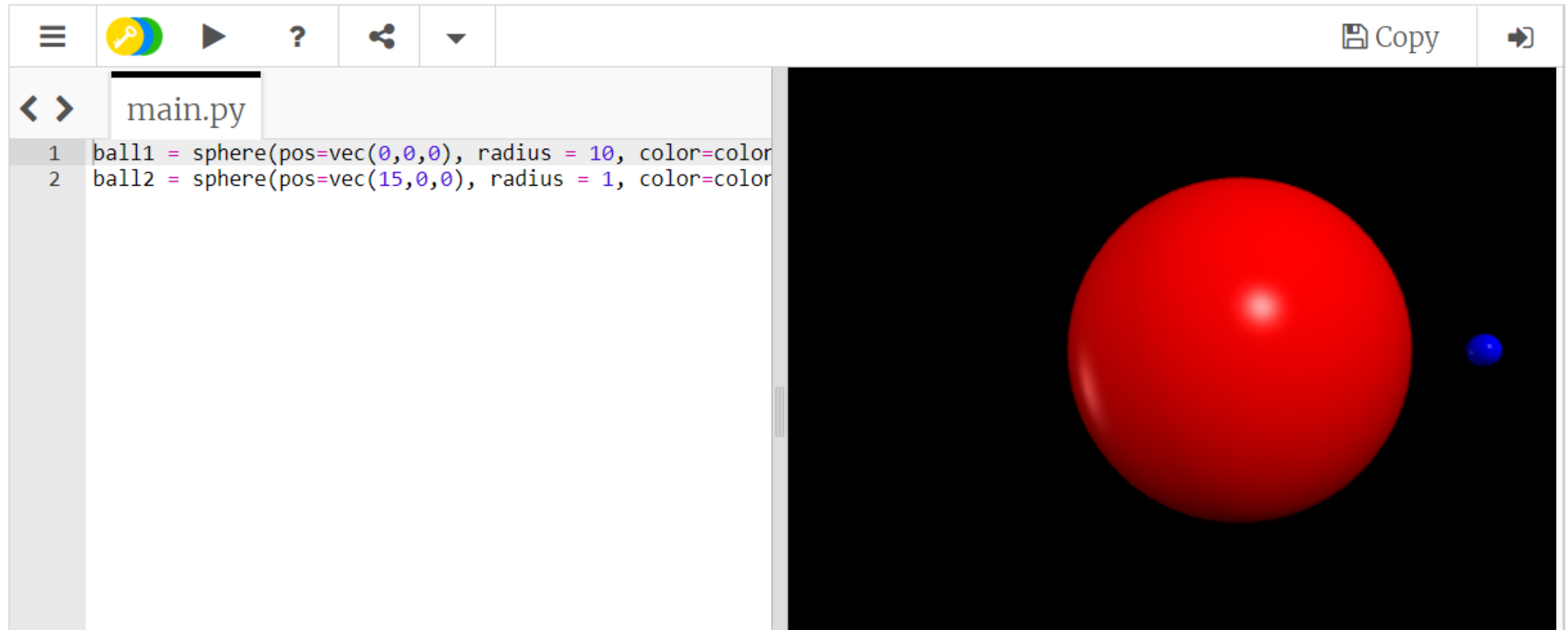
- Glowscript (now Web VPython)
- Trinket
- Jupyter Notebooks
(Python or Octave kernels)
- Octave (MATLAB workalike)

Glowscript (now Web VPython)

```
Fspring = - k*X  
Fnet = Fspring #the vertical forces cancel  
pBlock = pBlock + Fnet*delta_t  
Block.pos = Block.pos + (pBlock/mass) * delta_t  
Spring.axis = Block.pos - Wall.pos
```



Trinket (runs in browser)

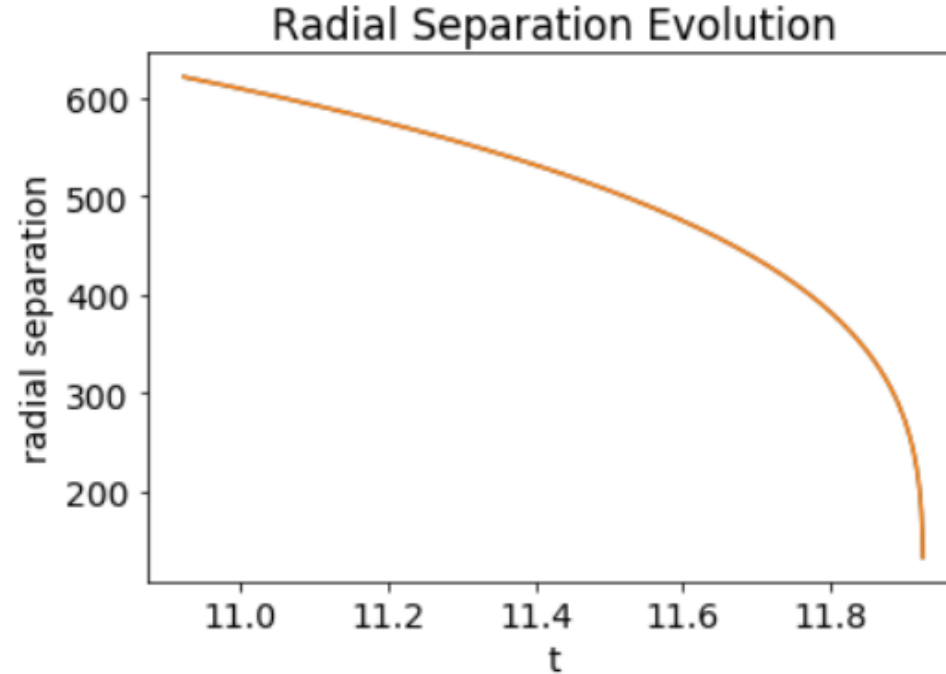


Jupyter Notebooks

Use in Google colab
or have students
install an IDE (eg,
VSCode)

```
plt.ylabel('Radial Separation')  
plt.xlabel("t")  
plt.plot(tscaled, rs)  
plt.title('Radial Separation Evolution')
```

Out[17]: Text(0.5, 1.0, 'Radial Separation Evolution')



Calculation of Inspiral Waveform

Just like electromagnetic radiation, gravitational waves have two different polarizations. However, radiation's polarizations differ by $\pi/2$ while gravitational waves have polarizations that differ by π . The cross-polarized strains given by the general formula (Huerta et al., 4' Gopakumar and Iyer, 4

$$h_+ = -\frac{M\eta}{r} \left[(\cos^2 \theta + 1) \left[\left(-\dot{r}^2 + r^2 \dot{\Phi}^2 + \frac{M}{r} \right) \cos 2\Phi \right. \right.$$

Octave
(like MATLAB,
but free)

Use in browser
(Octave Online)
or have students
install GNU
Octave

Image from:
<https://octave.org/>



Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in 2D/3D plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Microsoft Windows
- Drop-in compatible with many Matlab scripts

The Octave interpreter can be run in [GUI mode](#), as a console, or invoked as part of a shell script.

```
b = [4; 9; 2] # Column vector
A = [ 3 4 5;
      1 3 1;
      3 5 9 ]
x = A \ b      # Solve the system Ax = b
```

Curriculum Collections

PICUP (gopicup.org) Good for standard physics courses

The screenshot shows the PICUP website interface. At the top, there's a navigation bar with the PICUP logo and the tagline "Partnership for Integration of Computation into Undergraduate Physics". To the right of the logo, there are links for "My Account | Logout", "My Reviews | My Bookmarks", "Authoring Dashboard", and a "Feedback" button. Below the navigation bar is a horizontal menu with tabs: "Home", "Exercise Sets", "Faculty Commons", "Resources", "Community", "Events", and "About PICUP".

The main content area features a "Welcome to PICUP!" message followed by the text "View the PICUP Collections - materials and support for integrating computation:". Below this, there are two main sections: "Exercise Sets" and "Faculty Commons". Each section has a "Browse" button and an "Author" button. The "Exercise Sets" section also has a "Contribute" button. Below these sections, there's a banner for "All Mechanics Exercise Sets" with four small images: a 3D plot of a helix, a rocket launch, a diagram of Earth with a vector, and a diagram of a projectile on an inclined plane.

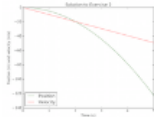
On the right side of the page, there's a "Get Started!" section. It includes a "Sign up to receive PICUP Announcements" button. Below this, there are two main items: "PICUP Virtual Conference" and "PICUP Capstone Conference 2020". The "PICUP Virtual Conference" section mentions that the conference will feature 18 presentations and 5 panel discussions to provide tips, tricks, and best practices for teaching physics online. The dates are Friday, June 26 and Wednesday, July 1, 2020, and the location is "On your computer!". The "PICUP Capstone Conference 2020" section states that the conference has been postponed until 2021 due to the ongoing crisis. The dates are July 15-18, 2020, and the location is "POSTPONED".

At the bottom of the page, there's a footer with the PICUP logo and a link to "Join the PICUP Community on Slack for in-depth discussions on computation in physics courses".

Curriculum Collections

PICUP (gopicup.org)

Good for standard physics courses

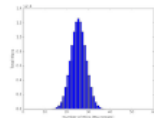


Using Python in Introductory Physics

First Year Programming Introductions

Developed by Eric Ayars

Specialized Programming Language: Python

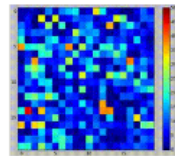


2-State Systems: Statistical Mechanics in Sports and the Story of the Purple Pandas

Beyond the First Year and Advanced Thermal & Statistical Physics

Developed by Brandon Lunk

Example Implementations: Python and Spreadsheet

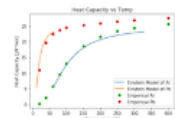


Boltzmann distribution, the statistical way

First Year and Beyond the First Year Thermal & Statistical Physics

Developed by Jay Wang

Example Implementations: Python and IPython/Jupyter Notebook



Einstein Solids: Equilibrium, Temperature and Heat Capacity

Beyond the First Year Thermal & Statistical Physics

Developed by Brandon Lunk

Example Implementations: IPython/Jupyter Notebook and Spreadsheet

Projectile Motion: Experiment and Computational Model

Developed by Todd Zimmerman - Published August 2, 2017

DOI: [10.1119/PICUP.Exercise.proj_mot](https://doi.org/10.1119/PICUP.Exercise.proj_mot)

Connecting computation to experiment is at the heart of physics. This set of exercises requires students to create a computational model for a ball launched into the air to determine the distance traveled and total time in the air. The students then must perform the experiment by launching the ball with a spring launcher. Data from the experiment must be entered into the computational model and the experimental results are compared to the computer model.



Subject Areas Mechanics and Experimental / Labs

Level First Year

Available Implementations Glowscript, IPython/Jupyter Notebook, Sage Worksheet, and Easy Java Simulations

- Learning Objectives**
- Explain that the time an object is in the air depends only on motion in the y-direction (**Exercise 4**)
 - Relate the initial velocity of a launched ball to the horizontal velocity when the ball is launched horizontally (**Exercise 2**)

Curriculum Collections

Data Science Education Community of Practice (DSECOP)
Labs and special projects (Jupyter Notebooks only)
<https://dsecop.org/>.

The screenshot shows the GitHub interface for the repository 'GDS-Education-Community-of-Practice / DSECOP'. The repository is public and has 10 watchers. The navigation bar includes links for Code, Issues (1), Pull requests, Actions, Projects (1), Wiki, and Settings. Below the navigation bar, there are buttons for 'main' (3 branches), '0 tags', 'Go to file', 'Add file', and a green 'Code' button. The repository content shows a commit by 'rmastand' edited 'notebook 2' (db1b1d8, 2 weeks ago) with 341 commits. The file list includes 'Exploratory_Data_Anal...' (Update README.md, 2 months ago), 'Intro_to_Data_Processi...' (edited notebook 2, 2 weeks ago), and 'Intro_to_Deep_Learning' (Update 12_Using_Deep_Learning_to_Find_Hot_J..., 2 months ago).

GDS-Education-Community-of-Practice / DSECOP Public Watch 10

<> Code Issues 1 Pull requests Actions Projects 1 Wiki S

main 3 branches 0 tags Go to file Add file <> Code

rmastand edited notebook 2 db1b1d8 2 weeks ago 341 commits

Exploratory_Data_Anal...	Update README.md	2 months ago
Intro_to_Data_Processi...	edited notebook 2	2 weeks ago
Intro_to_Deep_Learning	Update 12_Using_Deep_Learning_to_Find_Hot_J...	2 months ago

Curriculum Collections

DSECOP

Table of Contents

- [Intro to Data Processing](#) by Radha Mastandrea
- [Intro to Deep Learning](#) by Fatima Bagheri
- [Learning the Schrodinger Equation](#) by Karan Shah
- [NMR Deep Learning](#) by Sebastian Atalla
- [Solving Differential Equations with NNs](#) by Julie Butler
- [Spectral Clustering](#) by Cunwei Fan
- [Exploratory Data Analysis](#) by Radha Mastandrea
- [Intro to Random Forest](#) by Fatima Bagheri
- [Singular Value Decomposition](#) by Sebastian Atalla
- [Machine Learning Workflow](#) by Julie Butler

Curriculum Collections

PY4E - free curriculum for
python programming

You can embed videos and
autograded homework into
LMS

If you don't know python, you
can use this to teach yourself.



Curriculum Collections

PY4E - free curriculum for
python programming

You can embed videos and
autograded homework into
LMS

HW for Ch 4

In the past week, we learned built-in functions (Wednesday of week 2) and how to define your own functions (Friday of week 2). The following 2 videos corresponds to those lectures:

<https://www.youtube.com/watch?v=5Kzw-0-DQAk> ➞



<https://www.youtube.com/watch?v=AJVNYRqn8kM> ➞



Autograded Homework

4.6 Write a program to prompt the user for hours and rate per hour using input to compute gross pay. Pay should be the normal rate for hours up to 40 and time-and-a-half for the hourly rate for all hours worked above 40 hours. Put the logic to do the computation of pay in a function called **compute_pay()** and use the function to do the computation. The function should return a value. Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75). You should use **input** to read a string and **float()** to convert the string to a number. Do not worry about error checking the user input unless you want to - you can assume the user types numbers properly. Do not name your variable sum or use the sum() function.

Check Code

Reset Code



```
1 def compute_pay(h, r):
2     return 42.37
3
4 hrs = input("Enter Hours:")
5 p = compute_pay(10, 20)
6 print("Pay", p)
```

Your Output

Desired Output

Pay 498.75

A blue decorative graphic in the top-left corner, consisting of a solid blue rectangle with a curved, semi-circular cutout on its right side.

Interactive Textbook
has Trinkets embedded

Python for Everybody

by Charles Severance

Python for Everybody is an introduction to the basics of coding in Python 3 with an emphasis on practical usage, intended as a foundation for students who are looking to apply Python within other academic subjects as well as preparation for the serious study of computer science.

- [Chapter 1: Introduction](#)
- [Chapter 2: Variables](#)
- [Chapter 3: Conditionals](#)
- [Chapter 4: Functions](#)
- [Chapter 5: Iterations](#)

Summary: Curriculum Collections

- PICUP – everything physics
- DSECOP – data science
- PY4E - python

Common environments for python programming (free)

Anaconda – ecosystem with multiple environments (eg, Pycharm, Jupyter Notebooks)















Stand-alone installations:

PyCharm – free version does not support Jupyter notebooks

VSCode – free version supports Jupyter

IDLE – built-in to python upon install

Anaconda

 <p>CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated</p> <p>Launch</p>	 <p>Datalore Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.</p> <p>Launch</p>	 <p>IBM Watson Studio Cloud IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.</p> <p>Launch</p>	 <p>JupyterLab 3.3.2 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.</p> <p>Launch</p>	 <p>Jupyter Notebook 6.4.8 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.</p> <p>Launch</p>
 <p>PyCharm Community 2022.3.1 An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.</p> <p>Launch</p>	 <p>Qt Console 5.3.0 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.</p> <p>Launch</p>	 <p>Spyder 5.1.5 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features</p> <p>Launch</p>	 <p>VS Code 1.76.2 Streamlined code editor with support for development operations like debugging, task running and version control.</p> <p>Launch</p>	 <p>console_shortcut_miniconda 0.1.1</p> <p>Install</p>
 <p>Orange 3 3.32.0 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.</p>	 <p>powershell_shortcut_miniconda 0.0.1</p>	 <p>PyCharm Professional A Full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.</p>	 <p>RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.</p>	

Common
environments
for python
programming
(free)

PyCharm (makes version control easy)

The screenshot displays the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, Git, Window, and Help. The toolbar contains icons for file operations, navigation, and running tests (pytest). The main editor window shows the file `test_root_insert.py` with the following Python code:

```
1 import ...
7 def test_root_insert():
8     # Test insertion at the root node
9     M = 25 #interior node keys limit
10    L = 7 #leaf data limit
11    btree = BTree(M, L)
12    for i in range(8):
13        btree.insert(i, str(i))
14
15    root = DISK.read(btree.root_addr)
16    assert not root.is_leaf
17
18    assert root.get_child(0).data == ["0", "1", "2", "3"]
19    assert root.get_child(1).data == ["4", "5", "6", "7"]
20    assert btree.find(3) == "3"
21    for i in range(8):
22        print(btree.find(i))
```

The left sidebar shows the Project view with a tree structure of files: `btree.py`, `btree_node.py`, and `test_btrees.py`. The Commit view shows a commit message "Commit to main" and a list of changes. The Pull Requests view shows a list of pull requests. The Terminal view shows the output of the test run:

```
leaf key: [1]
leaf key: [2]
interior child key: []
children:
leaf key: [3]

===== short test summary info =====
FAILED tests/test_btrees.py::test_btree_properties_small_both - assert 0 == 1
===== 1 failed, 8 passed in 4.02s =====
```

IDLE

oodles_web_scraping_template.py - C:/Users/doneil/Box/doneil-migrated/Classes/PYTHON/scripts/oodles_web_scraping_template.py (3

File Edit Format Run Options Window Help

```
import re
import urllib.request
#visit the webpage:
hand = urllib.request.urlopen('https://www.gutenberg.org/ca

oodles = []
for line in hand:
    line = line.decode().strip()
    oodle_result = re.findall( #YOUR REGEX CODE HERE, line)
    if oodle_result != []: #remember that you get an empty
        #Add the new list (oodle_result) to the main list (ood

print("Found from the web: ")
print(oodles)
```


A few other
free resources

The Data Science Handbook

Free on github. Jupyter notebooks.



The screenshot shows the GitHub interface for the repository 'jakevdp / PythonDataScienceHandbook'. The repository is marked as 'Public'. Below the repository name, there are navigation links: 'Code' (highlighted with an orange underline), 'Issues' (with a count of 100), 'Pull requests' (with a count of 89), and 'Actions'. At the bottom, there is a dropdown menu showing 'master' and a breadcrumb trail: 'PythonDataScienceHandbook / notebooks /'.

jakevdp / PythonDataScienceHandbook Public


<> Code Issues 100 Pull requests 89 Actions


master PythonDataScienceHandbook / notebooks /


A few other
free resources


The Data Science Handbook


Includes numpy, pandas,
machine learning, etc.

 04.12-Three-Dimensional-Plotting.ipynb


 04.13-Geographic-Data-With-Basemap.ip...

 04.14-Visualization-With-Seaborn.ipynb

 04.15-Further-Resources.ipynb

 05.00-Machine-Learning.ipynb

 05.01-What-Is-Machine-Learning.ipynb

 05.02-Introducing-Scikit-Learn.ipynb

 05.03-Hyperparameters-and-Model-Valid...

A few other free resources

Daniel Schroeder's Website

Scientific Computing Resources

<https://physics.weber.edu/schroeder/>

Computational Physics e-text
(Python/Jupyter)

Quantum Mechanics e-text (Mathematica)

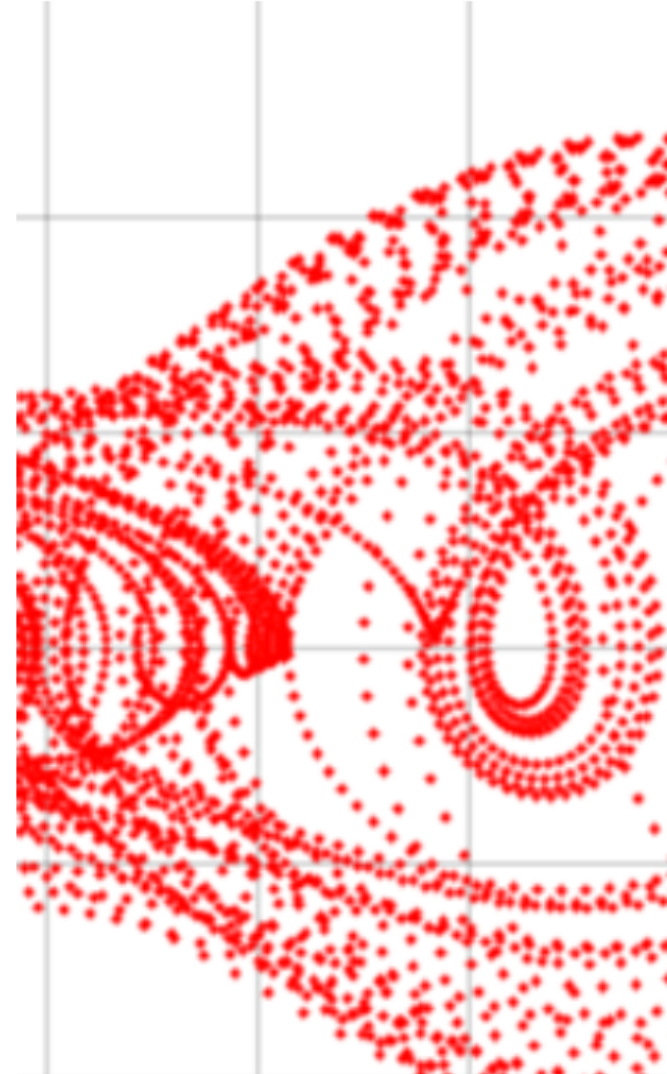
Thermal Physics book

(not free, but has free computational tutorials
on website)

Scientific Computing (Physics 2300)

[Daniel V. Schroeder](#), [Department of Physics](#), [Weber State University](#).

- [Syllabus](#)
- [Physics Simulations in Python](#) (lab manual)
- [LaTeX source and figure files for lab manual](#)
- [GlowScript home page](#)
- [VPython documentation](#)
- [Quick reference sheet](#) and [pdf version](#)
- [Download Anaconda](#)
- [Matplotlib.pyplot documentation](#)
- [Python resources for beginners](#)



Contact for more resources :
doneil@bridgewater.edu (can find on BC website)

Data Science Education Community of Practice (DSECOP)

<https://github.com/GDS-Education-Community-of-Practice/DSECOP>

<https://dsecop.org/>

Partnership for Integration of
Computation into Undergraduate Physics (PICUP):

gopicup.org

Python for Everybody (PY4E.com)

Scientific Computing Resources

<https://physics.weber.edu/schroeder/>

Dr. Deva O'Neil



Associate Professor of Physics

Department Chair of Engineering and Physics