UCLA Anderson Math Bootcamp

Charles Rambo

UCLA Anderson

2025

Table of Contents

1. About Me

2. How the Course Works

3. Python

About Me

About Me

- charles.tutoring@gmail.com
- github.com/charlesrambo
- linkedin.com/in/charlesrambo

Please email me your questions at my gmail account. Also, please feel free to connect with me on LinkedIn!

Education

- MFE from UCLA in 2020
- BA in Mathematics from UC Berkeley in 2009

Experience

State Street

Quantitative Analyst, Assistant Vice President (4 months)

- I work in the model risk magement team within enterprise risk management
- I use quantitiative methods to verify that State Street's credit, liqudity, and macroeconomic models work properly

SSI Investment Management

Portfolio Research Analyst (4 years)

- Firm specializes in convertible securities
- I did valuation, credit, equity, and portfolio construction

Experience

Rambo Tutoring (Self-Employed)

Math Tutor and Author (10 years)

- Did tutoring for precalculus, calculus, linear algebra, differential equations, statistics, probability, and more!
- Wrote study material as well! Check out my stuff on Amazon.com!

How the Course Works

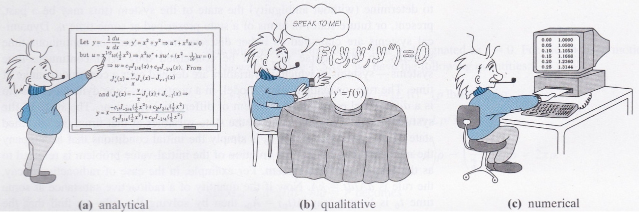
How the Course Works

- Links for videos avalable on my GitHub and emailed out
- Notes and homework posted on my GitHub
- Grade of 65% or better to pass; 25% credit given for timely submissions
- No exams

Homework

- Can work in teams of up to four people
- Make sure everybody's name is on the homework when you submit
- Email homework solutions to me at charles.tutoring@gmail.com
- Homework will be math problems which require some programming
- Submit as an html or pdf file; no screenshots, ipynb, or py files
- I look over the code but I don't run it so more important to make things readable than runable
- Due dates posted on homework
- Three assignments and one make-up assignment if you missed one or did badly

Approaches to Solve Math Problems


Let $y = -\frac{1}{u}\frac{du}{dx} \Rightarrow y' = x^2 + y^2 \Rightarrow u'' + x^2 u = 0$
but $u = x^{1/2} w(\frac{1}{2}x^2) \Rightarrow x^2 w'' + x w' + (x^2 - \frac{1}{16})w = 0$
$\Rightarrow w = c_1 J_{1/4}(x) + c_2 J_{-1/4}(x)$. From
$J_\nu'(x) = \frac{\nu}{x} J_\nu(x) - J_{\nu+1}(x)$
and $J_\nu'(x) = -\frac{\nu}{x} J_\nu(x) + J_{\nu-1}(x) \Rightarrow$
$y = x \frac{-c_1 J_{-3/4}(\frac{1}{2}x^2) + c_2 J_{3/4}(\frac{1}{2}x^2)}{c_1 J_{1/4}(\frac{1}{2}x^2) + c_2 J_{-1/4}(\frac{1}{2}x^2)}$
SPEAK TO ME!
$F(y, y', y'') = 0$
$y' = f(y)$
0.00 1.0000
0.05 1.0500
0.10 1.1053
0.15 1.1668
0.20 1.2360
0.25 1.3144

(a) analytical **(b)** qualitative **(c)** numerical

- Work out analytical solutions yourself without a computer
- Very few qualitative probems in this course
- Use Python to obtain numerical solutions

Image from page 28 of *A First Course in Differential Equations* 8th ed. by Zill

Tentative Course Outline

Unit	Description	Sessions
1	Calculus	July 9-18
2	Linear algebra and multivariable calculus	July 23-August 1
3	Combinatorics, probability, and statistics	August 6-15
4	Covariance matrices, PCA, and stochastic calculus	August 20-22

References

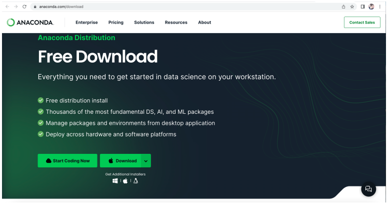
This is an incomplete list of references used to create the notes for this course. You do not need to purchase any of the books listed.

- James Stewart, *Calculus*, Brooke/Cole, 3rd ed., 1995.
- Walter Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, 1976.
- Charles Pugh, *Real Mathematical Analysis*, Springer, 2002
- Serge Lang, *Linear Algebra*, Springer, 3rd ed., 1987.
- Steven Roman, *Advanced Linear Algebra*, Springer, 2nd ed., 2005.
- Morris DeGroot and Mark Schervish, *Probability and Statistics*, Pearson, 4th ed., 2013.
- Marcos Lopez de Prado, *Machine Learning for Asset Managers*, Cambridge University Press, 2020.
- Martin Haugh, *A Brief Introduction to Stochastic Calculus*, Access date June 2025, ⟨`https://www.columbia.edu/~mh2078/FoundationsFE/IntroStochCalc.pdf`⟩, Columbia University, 2016.

Python

Python Installation

- We'll be using Python to obtain numerical solutions.
- If you haven't used Python before, I suggest downloading Anaconda at `https://www.anaconda.com/download`



Anaconda Distribution

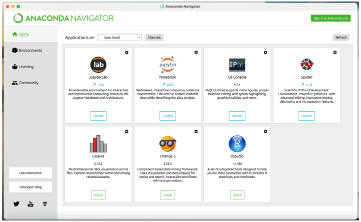
Free Download

Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

Anaconda

After you've installed and opened Anaconda, a screen like this will appear. You have several choices for IDEs. Spyder is useful for data analysis. Jupyter Notebook is popular for explanatory work, so students and teachers tend to use it. You can use any IDE.



JupyterLab
1.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console
4.7.4
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
4.1.3
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.23.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Packages

In this class we'll mostly be using the modules below.

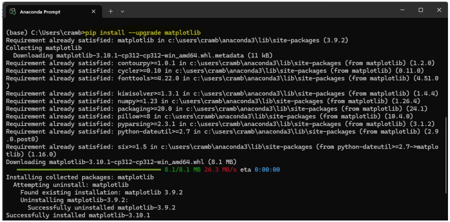
- *NumPy*: Useful for mathematical calculations.
- *Pandas*: Has data frame data type—it allows you to hold data in a format sort of like a spreadsheet.
- *Matplotlib:* Useful for graphing. Somewhat quirky syntax but very popular nonetheless.
- *SciPy*: Useful for scientific computing. It has more advaned math and statistics functions than NumPy.

Package Installation

To install or upgrade a module, go into Anaconda Promt (Windows) or Terminal (Mac/Linux), and type

```
pip install ...
```

In this example, I'm upgrading matplotlib.



```
(base) C:\Users\cramb>pip install --upgrade matplotlib
Requirement already satisfied: matplotlib in c:\users\cramb\anaconda3\lib\site-packages (3.9.2)
Collecting matplotlib
  Downloading matplotlib-3.10.1-cp312-cp312-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (4.51.0
)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.23 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\cramb\anaconda3\lib\site-packages (from matplotlib) (2.9
.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\cramb\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplo
tlib) (1.16.0)
Downloading matplotlib-3.10.1-cp312-cp312-win_amd64.whl (8.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.1/8.1 MB 26.3 MB/s eta 0:00:00
Installing collected packages: matplotlib
  Attempting uninstall: matplotlib
    Found existing installation: matplotlib 3.9.2
    Uninstalling matplotlib-3.9.2:
      Successfully uninstalled matplotlib-3.9.2
Successfully installed matplotlib-3.10.1
```

Python Graphing Example

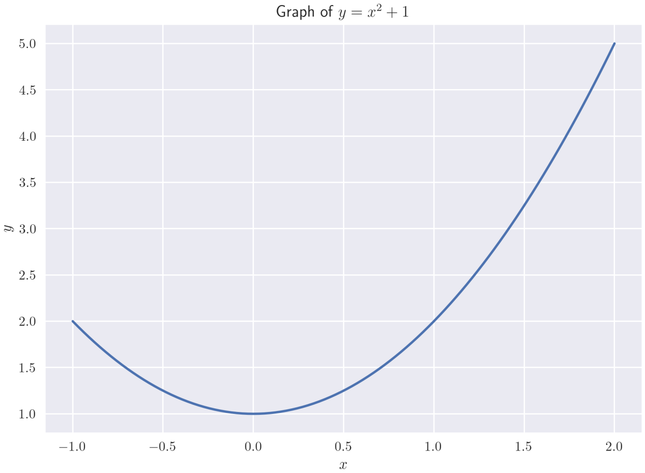
Example

Let $f(x) = x^2 + 1$. Use Python to graph f on the domain $[-1, 2]$.

```
# Import modules
import numpy as np
import matplotlib.pyplot as plt

# These three steps add style

# Use LaTex
plt.rcParams['text.usetex'] = True

# Increase image resolution
plt.rcParams['figure.dpi'] = 300

# Use Seaborn style
plt.style.use('seaborn-v0_8')

# Ready to do the math part

# Define f
def f(x):
    return x**2 + 1

# Another option is to use a lambda
    function
# f = lambda x: x**2 + 1

# Get 100 x-values on [-1, 2]
x_vals = np.linspace(-1, 2, 100)
```

```
# Use list comprehension to get y-values
y_vals = [f(x) for x in x_vals]

# Automatically vectorized so this works
    too
# y_vals = f(x_vals)

# Generate the plot
plt.plot(x_vals, y_vals)

# Label the x-axis
plt.xlabel('$x$')

# Label the y-axis
plt.ylabel('$y$')

# Give the graph a title
plt.title('Graph of $y = x^2 + 1$')

# Save the figure
plt.savefig(path + 'ex0-1')

# Display the plot
plt.show()
```

Python Graphing Result



Python Optimization Example

Example

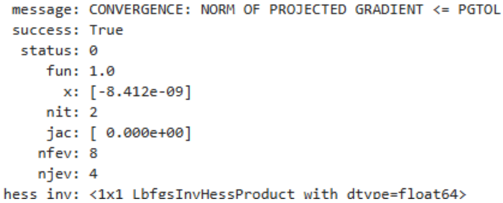
Use Python to find the minimum of $f(x) = x^2 + 1$ on the interval $[-1, 2]$.

Solution. From the graph on the previous page, we know that the minimum is $y = 1$ which occurs when $x = 0$. But let's use Python to verify this. Suppose the code above is still in our local environment.

Python Optimization Example

```python
# Import minimize from scipy
from scipy.optimize import minimize

# Minimize function; set bounds equal to the domain
minimize(f, x0 = [1], bounds = [(-1, 2)])
```

The output is shown below.



```
  message: CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL
  success: True
   status: 0
      fun: 1.0
        x: [-8.412e-09]
      nit: 2
      jac: [ 0.000e+00]
     nfev: 8
     njev: 4
 hess_inv: <1x1 LbfgsInvHessProduct with dtype=float64>
```