



A Collection of Scripts for Ab Initio Calculations: Migrating from Python 2 to 3

BY HELENA YANG

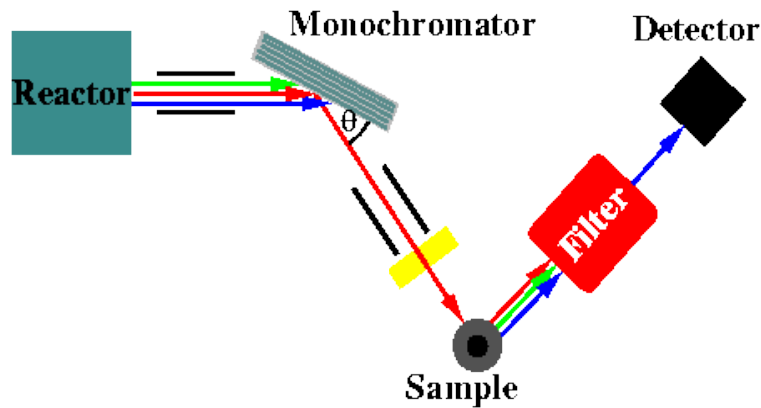
MENTORED BY WEI ZHOU, PHD



Usage of Scripts (Phonon)

Ab Initio: using known structures of molecules/crystals to study properties

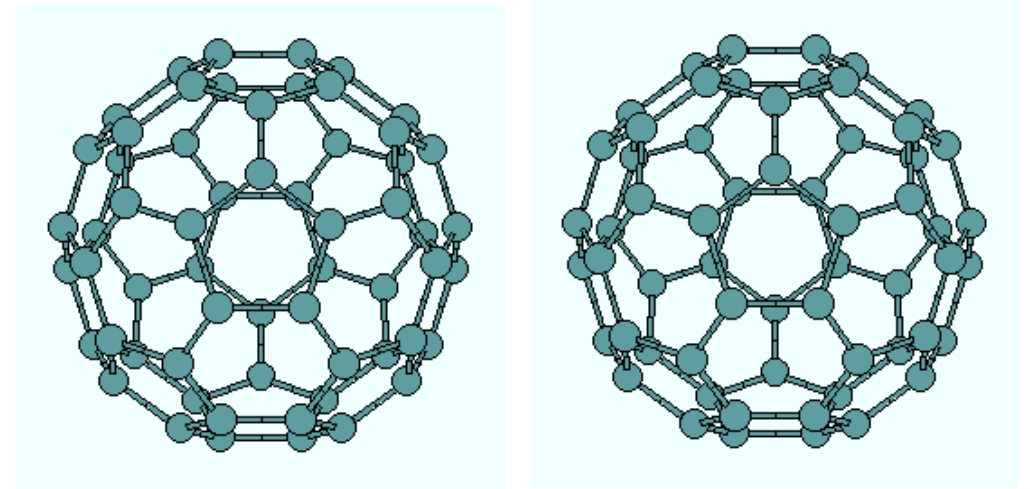
Scripts interpret experimental data from neutron **spectroscopy**



Filter-Analyzer Neutron Spectrometer (FANS)



Vibrational Modes of Water Molecules



2 Vibrational Modes of C₆₀ Molecule

Transitioning Process

Scripts written in Python 2

- Incompatible with Python 3
- No longer supported in 2020

Total Code Transitioned (phonon + auxiliary):

123 scripts

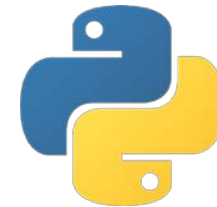
30,818 lines

Goals:

- Transition scripts to run in Python 3
- Improve accuracy of calculations, if possible

Process:

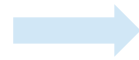
1. Apply 2to3 Fixers
2. Adjust to stricter code behavior
3. Transition Outdated Modules / Scripts
4. Parallel Testing
5. Improve Transition Efficiency



Step 1: Apply 2to3 fixers in command line

Automated Python 2 to 3 code translator

Python 2



Python 3

PRINT statement

PRINT() function

MAP()

LIST(MAP())

•
•
•

•
•
•

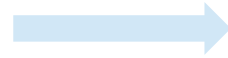
Final files:

Edited script (with Python 3 fixers)

Backup of original script (Python 2)

Step 2: Adjust to stricter code behavior

Python 2



Python 3

DIVISION operations

/ floor division

ex. $5/2 \rightarrow 2$

STRING module

`string.split(mystr)`

SPACE/TAB interchangeable

⋮

DIVISION operations

/ true division

ex. $5/2 \rightarrow 2.5$

STRING methods

`mystr.split()`

SPACE/TAB incompatible

⋮

Require
manual
fixes

Step 3: Transition outdated modules

Python 2



Python 3

**NUMERIC
LINEARALGEBRA**] modules

NUMPY module

Carry over

array(a)
matrixmultiply(a, b)
⋮

array(a)
dot(a, b)
⋮

axis=0 'takes'
rows of matrix

Exceptions

take(a, indices, axis=0)
eigenvectors(a)
Gives left-hand (row) evecs
⋮

take(a, indices, axis=-1)
linalg.eig(a)
Gives right-hand (column) evecs
⋮

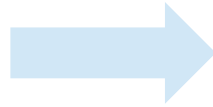
axis=-1 'takes'
columns of matrix

A 3x4 matrix is shown with a light blue vertical highlight on the first column and a light yellow horizontal highlight on the second row. The matrix elements are labeled as follows:

$a_{1,1}$	$a_{1,2}$	$a_{1,3}$...
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$...
$a_{3,1}$	$a_{3,2}$	$a_{3,3}$...
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮

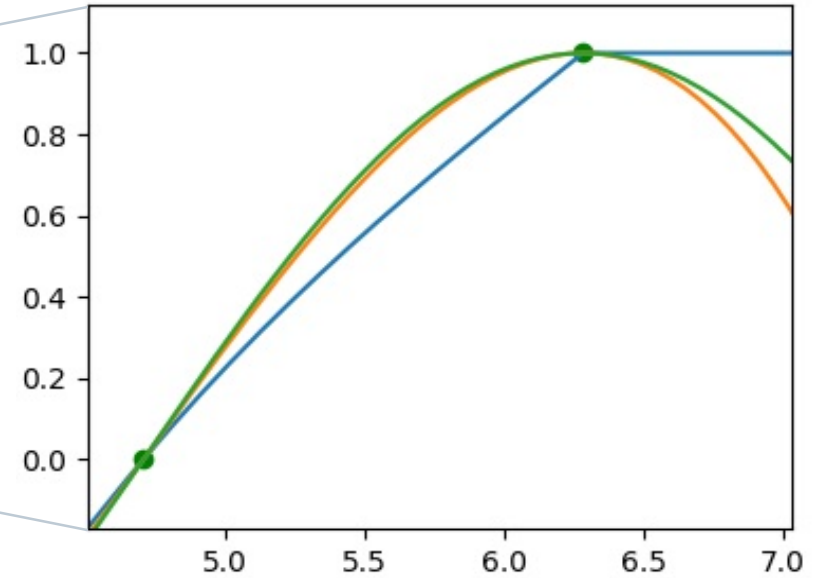
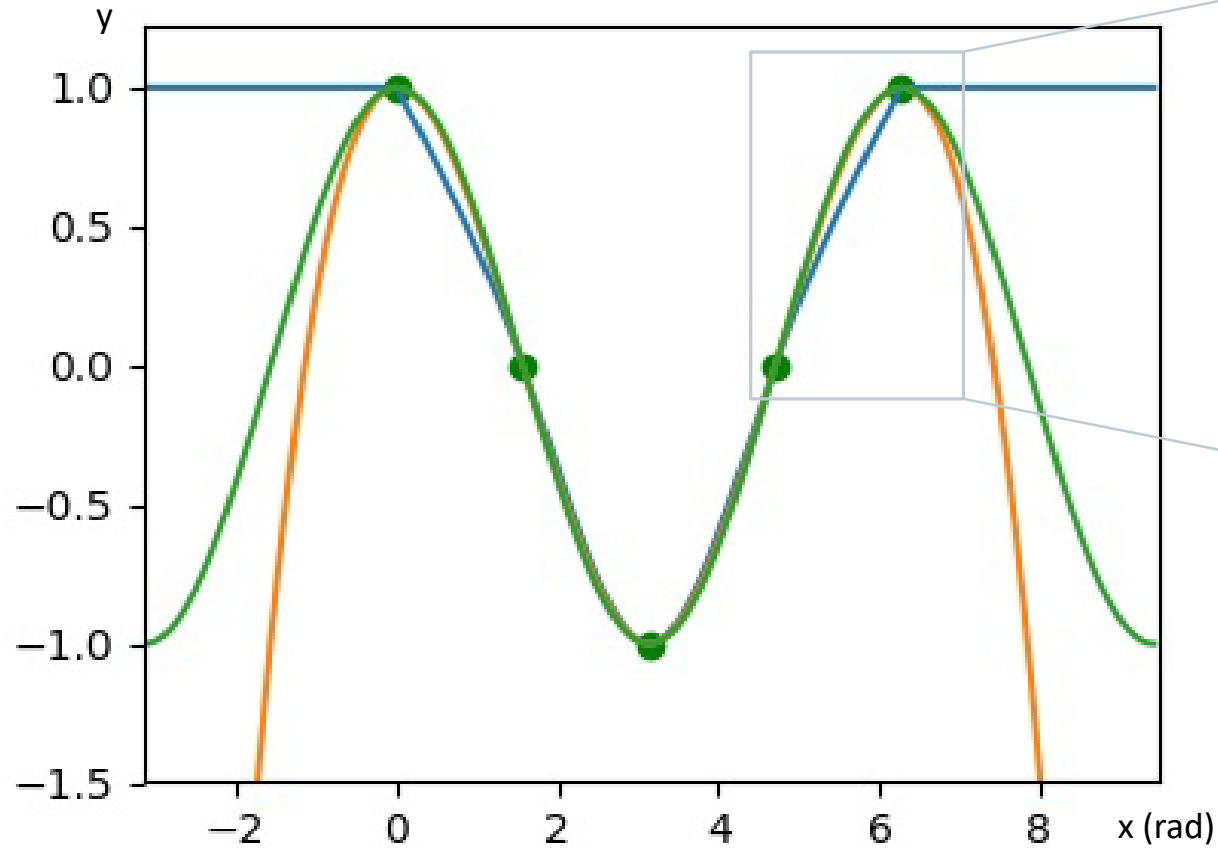
SPLINE

Third party script (Python 2)



SCIPY.INTERPOLATE.CubicSpline

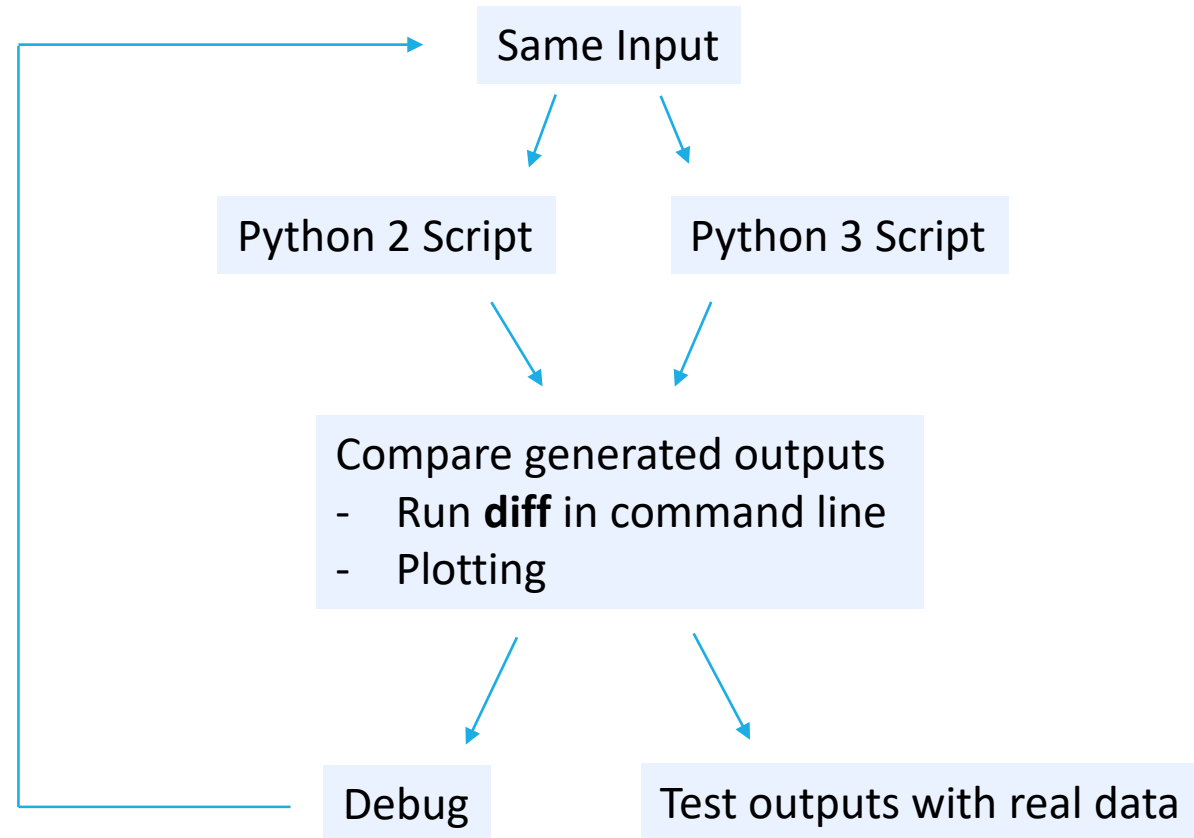
Spline Comparison Using $\cos(x)$ Data Points

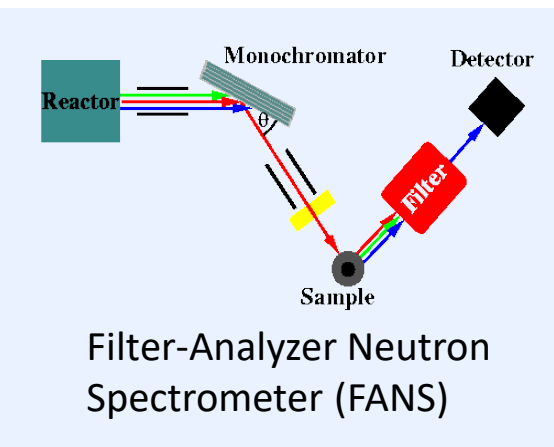


Key

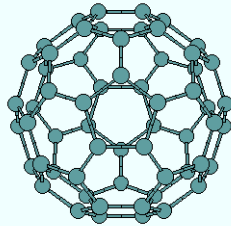
- $\cos(x)$ data points
- $\cos(x)$ graph
- Third party spline
- Scipy CubicSpline

Step 4: Test scripts in parallel

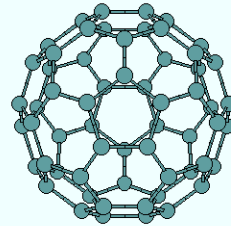




Neutrons counted by detector

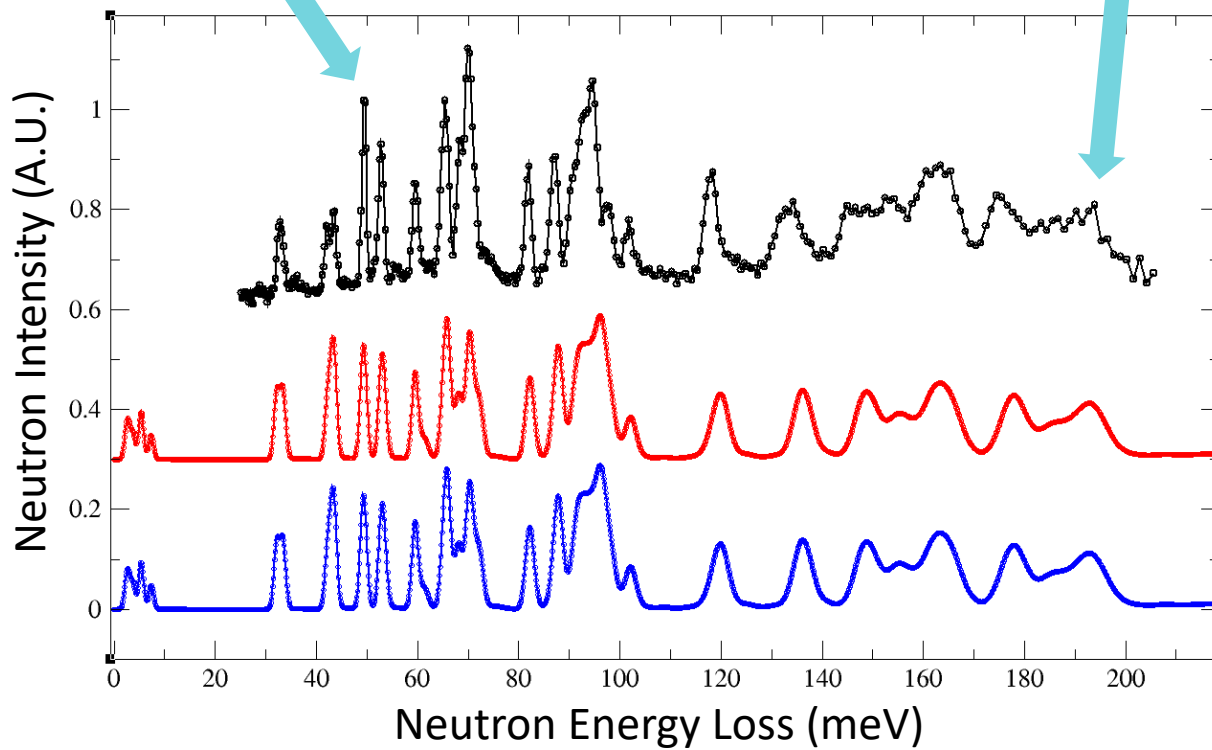


C_{60} Ag(1) mode
54.38420 meV



C_{60} Ag(2) mode
198.27823 meV

Vibrational Spectrum of C_{60}



Vibrational Energy

Experimental Data gathered from FANS

Python 2 Output (Calculated Spectrum)

Python 3 Output (Calculated Spectrum)

5. Improving Transition Efficiency

Supplementary version of Python's 2to3

Rewrites most Numeric
& LinearAlgebra
functions with NumPy

Rewrites former string
functions as string methods

Replaces any tabs
(\t) with 8 spaces

Prints line number & content
of functions that are difficult
to transition

Limitations:

- **NUMERIC** and **MATH** modules share functions (ex. **pi**, **sin()**, **cos()**)
- **int()** may need to be added to index values to ensure edited scripts run

Conclusion

We successfully migrated a subset of scripts (phonon + auxiliary).

We wrote and tested a supplementary 2to3 script (addresses manual fixes).

→ Useful for other scripts (ex. thermal expansion, elastic properties)

Things I learned:

- Phonon programming
- Neutron spectroscopy
- Linear algebra applications

Total Code Transitioned:

123 scripts

30,818 lines

Acknowledgements

Special thanks to:

My mentor, Wei Zhou;

SHIP mentors Markus Bleuel, Heather Chen-Mayer, Jacob LaManna, Brian Maranville, Ryan Need, & William Ratcliff;

SHIP Directors Julie Borchers, Joe Dura, & Yamali Hernandez;

The NIST Center for Neutron Research & The Center for High Resolution Neutron Scattering

References

1. Ascher, D., F. Dubois, P., Hinsen, K., Hugunin, J., & Oliphant, T. (2001). *An Open Source Project: Numerical Python*.
2. Chaplin, M. (2019). *Water Vibration Modes* [GIF].
3. Filter-Analyzer Neutron Spectrometer (FANS). (2004). Retrieved July 31, 2019, from https://ncnr.nist.gov/instruments/fans/fans_princ.html
4. Jing, D., & Pan, Z. (2009). Molecular vibrational modes of C60 and C70 via finite element method. *European Journal of Mechanics - A/Solids*, 28(5), 948-954. doi:10.1016/j.euromechsol.2009.02.006
5. Jones, E., Oliphant, T., Peterson, P., & others. (2001). *SciPy: Open source scientific tools for Python*. Retrieved from “<http://www.scipy.org/>”
6. Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.
7. Python Software Foundation. Python Language Reference, version 3.7. Available at “<http://www.python.org/>”

