

# Teaching Reciprocal Space to Undergraduates via Theory and Code Components of an IPython Notebook

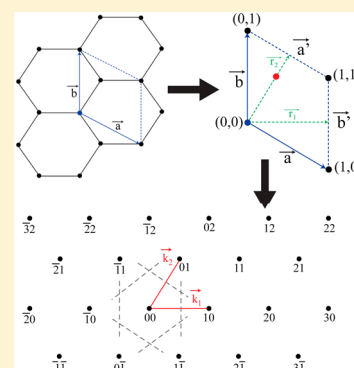
Matthew N. Srncic,<sup>†</sup> Shiv Upadhyay,<sup>‡</sup> and Jeffrey D. Madura<sup>\*‡</sup>

<sup>†</sup>Department of Chemistry, Physics, and Engineering, Franciscan University, Steubenville, Ohio 43952, United States

<sup>‡</sup>Department of Chemistry and Biochemistry, Center for Computational Sciences, Duquesne University, Pittsburgh, Pennsylvania 15282, United States

**S** Supporting Information

**ABSTRACT:** In this technology report, a tool is provided for teaching reciprocal space to undergraduates in physical chemistry and materials science courses. Reciprocal space plays a vital role in understanding a material's electronic structure and physical properties. Here, we provide an example based on previous work in the *Journal of Chemical Education* literature, carry out a calculation of reciprocal space vectors by hand, and present a computational tool for applying this mathematical process to other systems. Along with this paper, we have provided the stand-alone Python program that converts real space vectors to reciprocal space vectors and have also provided an IPython notebook that combines theory with components of the full Python code so that instructors can use this tool in their physical chemistry or materials science lectures.



**KEYWORDS:** Upper-Division Undergraduate, Physical Chemistry, Computer-Based Learning, Solid State Chemistry, Computational Chemistry

Solid-state materials are frequently investigated for their exciting physicochemical applications, which include photovoltaic solar cells, optics, and thermoelectrics.<sup>1–3</sup> In order to evaluate a material for these applications, the solid-state physicist/chemist inspects the electronic structure of the solid. Components of the electronic structure include the band structure and density of states, topics covered in undergraduate physical chemistry and materials science courses. For example, the band structure diagram depicts the energy of bands as a function of k-points on a path connecting the high symmetry k-points in the irreducible Brillouin zone.<sup>4,5</sup> Understanding the meaning of these k-points requires knowledge of reciprocal space, the mathematical relationship between real and reciprocal space vectors, and the role that these reciprocal space vectors play in determining the physical properties of solids. Several studies have focused on these topics in the literature, but none have mathematically shown the stepwise transformation of these vectors along with a computational tool for automating the process.<sup>6–8</sup> Although the topics have been briefly discussed, the mathematical process and tools presented herein have yet been presented in the chemistry education literature.

Electronic structure calculations have been carried out on a variety of compounds, ranging from simple to compositionally complex. To carry out such calculations, several software packages have been developed and utilized.<sup>9–11</sup> However, simple examples can be solved by hand to illustrate the important concepts of an electronic structure calculation to

students. In a 1991 *J. Chem. Educ.* article, Pisanty utilized the Hückel method to determine the band structure of graphite.<sup>12</sup> This provided an excellent example for students to consider the crystal structure and reciprocal lattice, set-up the secular determinant and solve for the energy of the bands, and, ultimately, interpret their results. However, the details necessary to make this an effective pedagogical tool are lacking, particularly for the conversion of real space vectors to reciprocal space vectors in the two-dimensional graphite example.

We feel that the absence of the mathematical details for this vector conversion in Pisanty's work reflects a larger void in the presentation of reciprocal space to undergraduates. Oftentimes, textbooks and other educational tools in solid-state chemistry define the real space vectors and simply skip ahead to the final reciprocal space vectors, leaving the student to wonder what mathematical "magic" took place in between. In three-dimensional examples, the details may be shown for carrying out this conversion using the formulas provided in eq 1 below. Furthermore, the inclusion of a rotational matrix in the two-dimensional case poses a challenge for the student. Pisanty's work presents such an example (graphite), and the mathematical details are omitted for simplification.<sup>12</sup> This leads some instructors to omit the topic of reciprocal space from their classes altogether due to lack of proper tools to

Received: May 26, 2016

Revised: October 13, 2016

explain the time-consuming examples and abstract nature of the topic.

With the vast amount of materials science and alternative energy research taking place, students will undoubtedly come across materials and their reciprocal lattices as they investigate electronic properties. The tools presented herein aim at bridging the gap between the abstract mathematical nature of reciprocal space and its real world applications. We expand on the work of Pisanty, beginning with an on-paper two-dimensional example for graphite in the [Supporting Information](#). Here, students will gain an appreciation for the mathematical details behind the reciprocal space vector conversion before applying their knowledge in a Python program that more quickly carries out the conversions. In addition, an IPython notebook is included so that instructors can seamlessly integrate the topic in their materials science or physical chemistry courses. This notebook simultaneously implements components of the theory and code, providing an efficient means of presenting the topic while also working on examples in class. Students will gain an enhanced appreciation for the reciprocal space topic from seeing the mathematical details implemented in the Python computer code. We feel that this hands-on experience will help build future solid-state chemists and bridge the gap between the topic's abstract nature and real world applications.

### ■ TWO-DIMENSIONAL EXAMPLE

Students and instructors should refer to the [Supporting Information](#) for the "on-paper" derivation of the reciprocal space vectors for two-dimensional graphite. Within the [Supporting Information](#), users will find the [2D\\_Example.docx](#) file. There, we highlight two real space translation vectors and convert them to reciprocal space vectors, demonstrating the mathematical process in detail.

### ■ TWO-DIMENSIONAL PRACTICE

After working the above two-dimensional example by hand, students are encouraged to work an additional case for practice. Here, students should set-up the necessary real space translation vectors for MoS<sub>2</sub> (molybdenum(IV) sulfide) and convert them to reciprocal space vectors. In working this problem, students should keep in mind the following questions:

- What do  $\vec{a}$  and  $\vec{b}$  represent in [Figure 1 of the Supporting Information](#) document?
- What are the values of  $\vec{a}$  and  $\vec{b}$  for graphite? Use the scientific literature.
- What is the crystal system of graphite? What is the crystal system of MoS<sub>2</sub>?
- How does the crystal system of these two compounds govern the reciprocal space vector calculation setup and results?
- Briefly comment on how the reciprocal space vectors for MoS<sub>2</sub> differ from those of graphite. (Assume you selected the same real space translation vectors to begin.)
- Use the IPython notebook or Reciprocal.py Python program in the [Supporting Information](#) to verify your results.

For additional practice, students should then independently select other compounds (select your favorite solid-state compounds from general/physical chemistry or materials science courses), whose crystal systems are different than that of graphite and MoS<sub>2</sub>. Here, students will set up the unit cell,

select real space translation vectors, and ultimately convert them to reciprocal space vectors. In a similar fashion as above, students should then use the Python program to verify their results.

### ■ THREE-DIMENSIONAL EXAMPLE

The corresponding formulas necessary to convert real space translation vectors to reciprocal space vectors for a three-dimensional solid are provided in [eq 1](#).

$$\begin{aligned} \mathbf{k}_1 &= \frac{2\pi \mathbf{r}_2 \times \mathbf{r}_3}{\mathbf{r}_1 \cdot (\mathbf{r}_2 \times \mathbf{r}_3)} & \mathbf{k}_2 &= \frac{2\pi \mathbf{r}_1 \times \mathbf{r}_3}{\mathbf{r}_2 \cdot (\mathbf{r}_1 \times \mathbf{r}_3)} \\ \mathbf{k}_3 &= \frac{2\pi \mathbf{r}_1 \times \mathbf{r}_2}{\mathbf{r}_3 \cdot (\mathbf{r}_1 \times \mathbf{r}_2)} \end{aligned} \quad (1)$$

In the three-dimensional case, the dot products of the real and reciprocal space vectors must satisfy the conditions in [eq 2](#).

$$\begin{aligned} \mathbf{r}_1 \cdot \mathbf{k}_1 &= \mathbf{r}_2 \cdot \mathbf{k}_2 = \mathbf{r}_3 \cdot \mathbf{k}_3 = 2\pi \\ \mathbf{r}_1 \cdot \mathbf{k}_2 &= \mathbf{r}_2 \cdot \mathbf{k}_1 = \mathbf{r}_1 \cdot \mathbf{k}_3 = \mathbf{r}_3 \cdot \mathbf{k}_1 = \mathbf{r}_2 \cdot \mathbf{k}_3 = \mathbf{r}_3 \cdot \mathbf{k}_2 = 0 \end{aligned} \quad (2)$$

As an exercise for students, the following real space translation vectors ([eq 3](#)) should be converted to reciprocal space vectors for a three-dimensional system and the correct reciprocal space vectors should be validated using the dot products in [eq 2](#).

$$\begin{aligned} \mathbf{r}_1 &= a(1, 0, 0) & \mathbf{r}_2 &= a\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right) \\ \mathbf{r}_3 &= a(0, 0, 2) \end{aligned} \quad (3)$$

### ■ RECIPROCAL SPACE PYTHON PROGRAM

The Reciprocal.py program is written in version 2.7 of the Python programming language.<sup>13</sup> This program runs on Windows, OSX, and Linux operating systems. Explicit instructions for installing Python on these operating systems can be found in the [Supporting Information](#). A .zip folder named Python\_Files contains the Installation\_Instructions.docx document. Users should contact the authors with any issues or questions that they have regarding installation and use.

### ■ PEDAGOGICAL FRAMEWORK

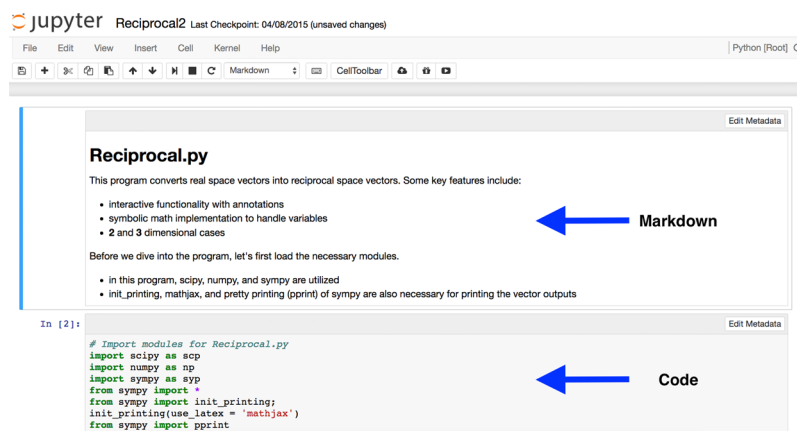
This tool is best suited for materials science and physical chemistry classrooms. The following pedagogical framework is recommended for using this tool in the classroom.

#### Incorporating This Approach in the Curriculum

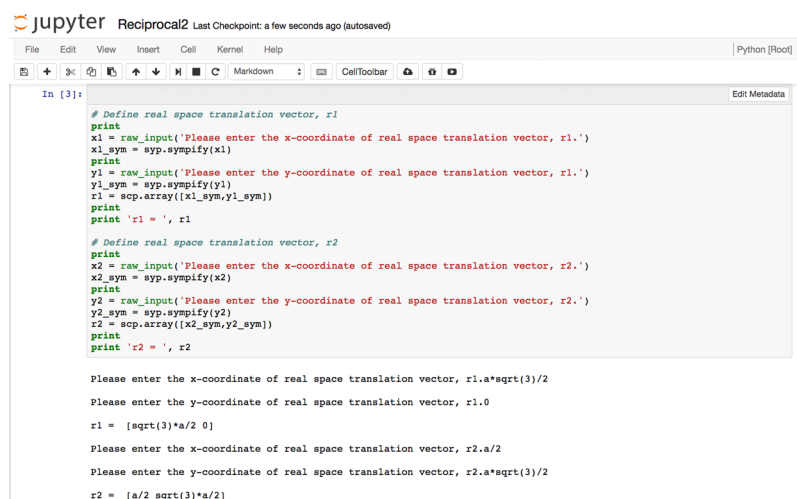
Instructors should evaluate their curriculum and identify the proper place for introducing the topic of reciprocal space. Unfortunately, most physical chemistry texts save solid-state materials for special topics or chapters later in the book, which sometimes are not covered for the sake of time. We anticipate this topic and tool will find its home at the conclusion of the quantum mechanics semester of physical chemistry. Solid materials are typically discussed following spectroscopy, and we strongly encourage discussing reciprocal space and utilizing the Python tools herein as a bridge to connect crystal lattices and unit cells to the electronic properties of solids.

#### Classroom Implementation

In preparation for presenting/discussing this topic in class, instructors should encourage their students to read this paper (they should also revisit Pisanty's original paper<sup>12</sup> for additional



**Figure 1.** Screenshot of the IPython notebook via Jupyter, illustrating the simultaneous presentation of concept, theory, and notes in the *markdown* section and Python programming commands in the *code* section.<sup>14</sup>



**Figure 2.** Screenshot of the IPython notebook via Jupyter, illustrating the Python programming commands (top) followed by the interactive comments (bottom) that are returned to the user as the program runs.<sup>14</sup>

topic context) prior to the class it will be covered. This allows students time to refresh their linear algebra and practice the matrix math operations prior to class. This will enhance classroom discussion and allow students to follow along more closely in class. It is **strongly advised** that students follow the installation instructions document and attempt to install/use the Python software on their personal computer (if available) prior to class. However, if students only read the paper and familiarize themselves with the terminology, concepts, and mathematics in advance, it will still aid significantly in the lecture's success.

Before class, it is imperative that the instructor has the IPython notebook working properly via Jupyter<sup>14</sup> on the campus computer in their classroom or on their own personal laptop for connection to the classroom's AV equipment. This tool is best utilized when students also have computer access. Here, the instructor must evaluate which is most appropriate in their classroom. We anticipate that most students will have laptops available, but if this is not the case, the instructor may need to reserve a computer lab and install the Python software on these machines for the students, so that they can also interact with the IPython notebook. If this cannot be accomplished, the instructor should simply have their computer prepared to run the notebook, and allow students to come up

in groups and examine the notebook's functions. The instructor can also reserve time outside of class for students to work with the IPython notebook in groups and answer the associated questions.

To begin class, instructors should first discuss the importance of solids and their properties while also covering basic terminology of crystals. This serves as a nice means of introducing reciprocal space. We anticipate that this will be best achieved as an open discussion with the class.

Following this brief 5–10 min discussion on the topic's significance, the instructor should transition to the IPython notebook via Jupyter. The power of this tool is that it allows the instructor to simultaneously present conceptual/theoretical content (text and equations) and their computer code (the Reciprocal.py Python program) in the same environment (Figures 1 and 2). Prior to the use of IPython, instructors were limited to screen-jumping between their lecture notes and a separate program to run computer code. Here, within the IPython notebook, instructors can present their concepts and theory in the *markdown* sections of the notebook. Following sections of theory, the instructor then has the ability to run his/her Python program using the *code* sections of the notebook. This provides a single, clear resource for presenting the content to students, allowing them to focus on the material, rather than

being distracted by the instructor transitioning between resources.

Utilizing the IPython notebook via Jupyter, we advise that instructors quickly (10–15 min) revisit the mathematical steps for converting real space translation vectors to reciprocal space vectors via the two-dimensional graphite example in the paper. This is provided in the *markdown* section of the IPython notebook. Once students are comfortable with this mathematical conversion, the instructor can proceed to the *code* section of the notebook and illustrate to students that these tedious mathematical processes can be translated into a series of computer-interpreted commands and the Python software can be utilized to quickly calculate the same values. Using the Python code within the IPython notebook, instructors should focus on first using the tool to validate the results that students obtained by carrying out the calculation by hand.

### Facilitating Student Use of the IPython Notebook

Next, instructors should allow students to interact with the computer in small groups. Here, they will get a feel for the IPython notebook prior to the end of class. This will be valuable experience so that the students can successfully answer the associated questions outside of class.

The instructor should remain after class to handle any Python installation questions and also make him/herself available outside of class to help students getting the software setup on their personal machines. If the instructor will have the software preinstalled on campus computers or will be allowing students to use his/her computer, this step is unnecessary.

The instructor can choose how they wish to evaluate students on this topic. We suggest an assignment consisting of the questions in the [Two-Dimensional Practice](#) section above. Here, students can work in small groups and should have access to computers for utilizing the IPython notebook.

## CONCLUSION

Upon mastering the calculation of reciprocal space vectors, these can be used to construct the first Brillouin zone of a solid. As shown in Figure 4 of Pisanty's article,<sup>12</sup> the Brillouin zone of graphite contains high symmetry k-points that can be used to construct the band structure diagram. From the solid's band structure, the electronic band gap can be determined, which ultimately governs the material's effectiveness in solar cell, optical, and thermoelectric applications. Here, we have shown one small component of these calculations. However, through this example and the accompanying Python program, we have systematically shown how one can obtain the reciprocal space vectors necessary for more elaborate electronic structure calculations. Instructors should progress through the IPython notebook as constructed, but we also encourage them to customize this notebook to their needs. Instructors are welcome to make changes, additions, or exclude certain sections if they desire.

## ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available on the ACS Publications website at DOI: [10.1021/acs.jchemed.6b00392](https://doi.org/10.1021/acs.jchemed.6b00392).

Full Python code (Reciprocal.py) for OSX Terminal or Anaconda Prompt in Windows, IPython notebook (Reciprocal.ipynb) for instructors to use in their classrooms, "on-paper" two-dimensional example (2D\_Example.docx), Python installation instructions

(Installation\_Instructions.docx), and Jupyter configuration file (jupyter\_notebook\_config.py) (ZIP)

## AUTHOR INFORMATION

### Corresponding Author

\*E-mail: [madura@duq.edu](mailto:madura@duq.edu).

### Notes

The authors declare no competing financial interest.

## REFERENCES

- (1) Polman, A.; Knight, M.; Garnett, E. C.; Ehrler, B.; Sinke, W. C. Photovoltaic materials: Present efficiencies and future challenges. *Science* **2016**, *352* (6283), aad4424-1–10.
- (2) Litchinitser, N. M.; Sun, J. Optical meta-atoms: Going nonlinear. *Science* **2015**, *350* (6264), 1033–1034.
- (3) Bell, L. E. Cooling, heating, generating power, and recovering waste heat with thermoelectric systems. *Science* **2008**, *321* (5895), 1457–1461.
- (4) Meyer, M.; Glaus, S.; Calzaferri, G. Introduction to basic terms of band structures. *J. Chem. Educ.* **2003**, *80* (10), 1221.
- (5) Gerstein, B. C. Chemistry, bands, bonds, and solids: A translator's note. *J. Chem. Educ.* **1973**, *50* (5), 316–322.
- (6) Ladd, M. F. C. The language of lattices and cells. *J. Chem. Educ.* **1997**, *74* (4), 461–465.
- (7) Mjoberg, P. J.; Ljunggren, S. O.; Ralowski, W. M. Reciprocal vectors in function space. *J. Chem. Educ.* **1975**, *52* (4), 212–214.
- (8) Kettle, S. F. A.; Norrby, L. J. The Brillouin zone—an interface between spectroscopy and crystallography. *J. Chem. Educ.* **1990**, *67* (12), 1022–1028.
- (9) Blaha, P.; Schwarz, K.; Madsen, G. K. H.; Kvasnicka, D.; Luitz, J. *WIEN2k: An Augmented Plane Wave Plus Local Orbitals Program for Calculating Crystal Properties User's Guide, WIEN2k 14.2*; Institute of Physical and Theoretical Chemistry, Vienna University of Technology: Vienna, Austria, 2014.
- (10) Kresse, G. VASP Group, Theoretical Physics Departments: Vienna. <http://www.vasp.at/> (accessed Sep 2016).
- (11) Dovesi, R.; Orlando, R.; Erba, A.; Zicovich-Wilson, C. M.; Civalieri, B.; Casassa, S.; Maschio, L.; Ferrabone, M.; De La Pierre, M.; D'Arco, P.; Noel, Y.; Causa, M.; Rerat, M.; Kirtman, B. CRYSTAL14: A program for the ab initio investigation of crystalline solids. *Int. J. Quantum Chem.* **2014**, *114*, 1287–1317.
- (12) Pisanty, A. The electronic structure of graphite—A chemist's introduction to band theory. *J. Chem. Educ.* **1991**, *68* (10), 804–808.
- (13) Python Software Foundation. *Python Language Reference, version 2.7*; Available at <https://www.python.org/> (accessed Sep 2016).
- (14) Pérez, F.; Granger, B. E. IPython: A System for Interactive Scientific Computing. *Comput. Sci. Eng.* **2007**, *9* (3), 21–29. Available at <http://ipython.org/> (accessed Sep 2016).