Received : 8 January 2025
Revised : 23 March 2025
Accepted : 10 April 2025
Published: 11 April 2025
Issued : 30 June 2025

DOI: doi.org/10.21009/1.11105

Cloud-Based Teaching Tool of AlCu Band Gap Simulations Using GPAW: A Python-Driven Approach for Undergraduate Student

Anggara Budi Susila[a)], Haris Suhendar[b)], Riser Fahdiran[c)]

Department of Physics, Universitas Negeri Jakarta, Jl. R. Mangun Muka Raya No. 11, RT.11/RW.14, Rawamangun, Kec. Pulo Gadung, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta, 13220, Indonesia

✉: [a)]anggarabs@unj.ac.id, [b)]haris_suhendar@unj.ac.id, [c)]riser-fahdiran@unj.ac.id

Abstract

This work introduces a computational teaching module that leverages Python, Google Colab, and the GPAW package to simulate the electronic band structure of AlCu materials. While powerful, traditional Density Functional Theory (DFT) tools like Quantum ESPRESSO or VASP often present steep learning curves and software installation challenges. By contrast, GPAW operating within Python and its seamless integration with Google Colab provides a user-friendly, platform-independent environment for students to explore quantum simulations without local setup requirements. The simulation workflow is highly efficient, with key processes such as structure creation taking only 7 milliseconds, structural relaxation requiring 51.2 seconds, and band structure calculations completing in just 40 seconds. In this educational framework, students model AlCu and its doped variants, visualize band structures, and analyze changes in the electronic properties induced by doping. The approach supports active learning and reinforces core solid-state physics, quantum mechanics, and computational materials science topics. Sample notebooks, learning outcomes, and classroom integration strategies are presented, aiming to democratize access to DFT education through open-source, cloud-based tools.

Keywords: educational teaching tool, material simulation, AlCu, density function theory, GPAW

INTRODUCTION

Electronic band theory is a cornerstone of modern physics and materials science, explaining why materials behave as insulators, semiconductors, or conductors. Mastery of concepts like valence and conduction bands is essential for understanding semiconductors, solar cells, and microelectronic devices. However, these concepts often remain abstract in traditional curricula. Incorporating computational models can ground band theory in concrete examples. For instance, doping a wide-bandgap semiconductor like ZnO with elements such as Al or Cu is known to alter its band gap, illustrating the principles of band engineering (Ma et al,2019; Dejam et al,2023; Wang et al,2015; Zhang et al,2023). Similarly, the aluminum copper (Al - Cu) alloy system provides a metallic contrast that helps students see how changing atomic composition affects electronic structure. For instance, Li et al. (2024) utilized first-principles calculations to analyze the influence of Mg doping on the structural stability and mechanical properties of the Al_2Cu precipitate in Al-Cu-Mg alloys, revealing enhanced stability and strength with doping. AlCu alloys are particularly relevant due to their widespread applications in electronics, electrical contacts, and batteries, where their electrical conductivity and

resistance to corrosion play a crucial role in performance (Sabolsky et al,2025; Wei et al,2024; Gu et al, 2016).

Despite its importance, teaching electronic structure through ab initio methods faces practical barriers. Traditional density-functional theory (DFT) codes typically require complex installations, Linux proficiency, and powerful computers, which can be prohibitive in a classroom setting. These barriers include high-performance requirements, black-box learning, installation, and maintenance complexity. Conventional DFT simulations often run only on clusters or supercomputers. Leveraging ordinary personal devices for such simulations was once impractical, but modern free and open-source software (FOSS) now enables a decentralized BYOD (bring-your-own-device) approach (Magers et al., 2021; Lehtola et al., 2022; Ju et al., 2024). Installing DFT packages (especially proprietary ones) can require substantial administrative effort. FOSS solutions and cloud platforms remove these hurdles by eliminating costly software maintenance. Many students may use DFT tools as opaque black boxes without understanding the underlying physics. This hampers deeper learning unless the tools are accompanied by pedagogical guidance (Hrubeš, 2024; Bylaska, 2024; Di Felice, 2023).

Educators have begun using cloud-based, user-friendly environments that integrate Python coding with materials simulations to overcome these challenges. For example, several studies report using Google Colaboratory (Colab) notebooks to teach science: Vallejo et al. developed a set of Colab notebooks to introduce coding and thermodynamics to chemistry students, and Hirschi et al. provided Colab-accessible Python notebooks to demystify DFT fundamentals (Vallejo, 2022). In this work, we extend this paradigm to band-structure education. We use GPAW (a Python-based DFT code) together with the Atomic Simulation Environment (ASE) in a Jupyter/Colab notebook (Mortensen, 2024). This Python-driven, cloud-based approach allows students to simulate the band structures of model materials (e.g. Al–Cu alloys and doped ZnO) without local software installation or specialized hardware. By focusing on a browser-based workflow, students can concentrate on physical concepts and visualization rather than setup details, aligning with modern computational curricula.

METHODS

This study employs a cloud-based computational approach to develop an interactive educational tool for teaching solid-state physics concepts, particularly electronic band structure and material doping through simulation. The tool consists of a Google Colaboratory (Colab) notebook that runs Python-based code for Density Functional Theory (DFT) simulations using GPAW (Grid-Based Projector Augmented Wave method) and the Atomic Simulation Environment (ASE). This notebook allows students to build crystal structures, perform geometry optimizations (relaxation), and compute and visualize band structures for AlCu and doped-AlCu materials from a web browser, without installing local software.

The simulation workflow was implemented entirely in a Colab notebook, utilizing the libraries shown in TABLE 1.

TABLE 1. Tools employed for material simulation for material simulation teaching

Tools	Utility
GPAW	Electronic structure calculations using real-space grids and plane waves.
ASE	Atomic manipulation, structure building, and file I/O.
Numpy	Numerical operations
Matplotlib	Visualization
Google Colab	Cloud-based code execution with a Jupyter-like interface.

The Colab environment provides a lightweight, interactive setting accessible from any device with an internet connection, enabling seamless integration into classroom activities. The AlCu alloy structure was constructed using ASE by placing aluminum and copper atoms in a face-centered cubic lattice, then expanded into a 3×3×3 supercell. To study doping effects, one atom was substituted with another element (e.g., Zn), allowing investigation of changes in electronic properties. Structural relaxation was carried out using GPAW with a plane-wave basis set (300 eV cutoff), PBE functional, and a 4×4×4 k-point grid. The BFGS optimizer, combined with a UnitCellFilter, was used to minimize

forces below 0.05 eV/Å. After relaxation, a fixed-density calculation was performed to compute the electronic band structure along a high-symmetry path in the Brillouin zone. The process was implemented in a Google Colab notebook, integrating code, visualization, and guided instructions. It is an effective and accessible educational tool for high school and undergraduate students learning about material simulations and band structure analysis.

RESULTS AND DISCUSSION

The proposed Google Colab notebook successfully implements a complete simulation workflow for AlCu alloy materials, including structure creation, relaxation, and electronic band structure calculation. By leveraging Python packages such as ASE (Atomic Simulation Environment) and GPAW (a real-space density functional theory calculator), students are guided through a hands-on simulation without installing complex software locally. All simulations run on Google's cloud infrastructure, making this approach hardware-independent and ideal for classrooms or remote learning.

Simulations Workflow

Library Installation

The required libraries for this research include ASE (Atomic Simulation Environment) for atomic modelling, GPAW for electronic structure calculations based on density functional theory (DFT), and Matplotlib for data visualization. These libraries were installed dynamically within the Colab environment using pip commands, ensuring platform independence and eliminating the need for local software setup. This installation approach is particularly beneficial in an educational context, as it simplifies the technical barrier for students and ensures uniform computational environments across users. The installation runtime only takes 1 minute, then can be continue to the next step. The code for installation is given in the following line.

```
!apt install python3-mpi4py cython3 libxc-dev gpaw-data
!pip -q install gpaw jarvis-tools spglib
```

Structure Creation

The face-centered cubic (FCC) crystal structure of the binary alloy AlCu was generated using the Atoms class in ASE. The lattice constant was set to 6.0 Å, and two atoms Al and Cu were positioned at (0, 0, 0) and (0.5a, 0.5a, 0.5a), respectively, where a is the lattice constant. This configuration reflects a simple binary alloy model, where Al and Cu atoms occupy alternating FCC lattice sites.

```
from ase import Atoms

a = 4  # konstanta kisi dalam Ångstrom
alcu = Atoms('AlCu',
             positions=[(0, 0, 0),
                        (0.5*a, 0.5*a, 0.5*a)],
             cell=[a, a, a],
             pbc=True)
```

To verify the initial structure, the unit cell was visualized using ASE's view() function. This step allows inspection of atomic arrangement and symmetry, helping students intuitively understand the spatial configuration of atoms in a crystal lattice. The structure of AlCu material is shown in FIGURE 1.

```
from ase.visualize import view

view(alcu, viewer='x3d')  # gunakan viewer='x3d' di Jupyter
```

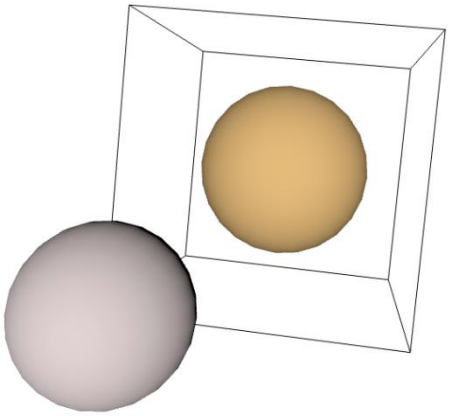

FIGURE 1. Visualization of single unit cell of AlCu material

To better simulate bulk material properties and to allow room for doping studies, the unit cell was replicated using the `make_supercell()` function. A 3×3×3 transformation matrix was applied to create a supercell comprising 54 atoms. This size ensures a sufficient volume for structural relaxation and electronic structure analysis while remaining computationally manageable for educational purposes. The structure for supercell is shown in FIGURE 2.

```
from ase.build import make_supercell

# Membuat supercell 3x3x3
supercell = make_supercell(alcu, [[3, 0, 0], [0, 3, 0], [0, 0, 3]])
supercell.center()
supercell.set_pbc(True)

view(supercell, viewer='x3d')  # gunakan viewer='x3d' di Jupyter
```

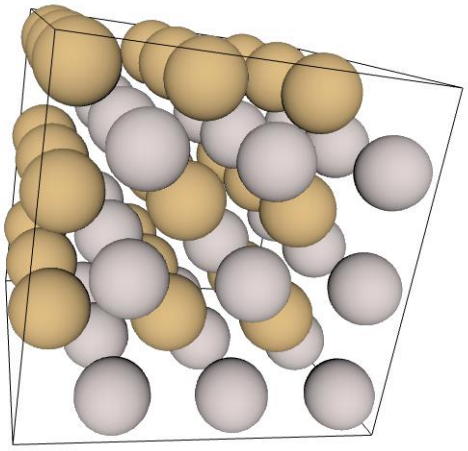

FIGURE 2. Supercell structure of AlCu extended in 3 x 3 matrix cells

Structure Relaxation

Once the AlCu structure was created, structural relaxation was performed to minimize the total energy of the system and obtain the most stable atomic configuration. This process simulates how atoms adjust their positions to reduce internal forces, providing a realistic starting point for subsequent electronic structure calculations. For the purpose of interactive learning, relaxation was only performed on the AlCu structure instead of the AlCu supercell structure due to the long running time with many atom calculations.

```
from gpaw import GPAW, PW, FermiDirac
from ase.constraints import UnitCellFilter
from ase.optimize import BFGS

calc = GPAW(mode=PW(300),
            kpts=(4, 4, 4),
            xc='PBE',
            occupations=FermiDirac(0.1),
            txt='alcu_output.txt')

ucf = UnitCellFilter(alcu)
alcu.set_calculator(calc)

ucf = UnitCellFilter(alcu)
opt = BFGS(ucf, trajectory='relax.traj')
opt.run(fmax=0.05)
```

Based on script above, we configure DFT using `GPAW` calculator with parameter for energy cutoff 300 eV (`mode=PW(300)`), sampling the Brillouin zone (`kpts=(4, 4, 4)`), exchange correlation (`xc='PBE'`), Fermi-Diract smearing width (`occupations=FermiDirac(0.1)`), and output written directory (`txt='alcu_output.txt'`). To allow relaxation of both atomic positions and unit cell parameters, the `UnitCellFilter` was applied to the AlCu structure. This wrapper ensures that forces on both atoms and the unit cell are considered during optimization. The `BFGS` (Broyden-Fletcher-Goldfarb-Shanno) optimization algorithm was employed to minimize forces acting on atoms. The relaxation continued until the maximum force on any atom was less than 0.05 eV/Å (`opt.run(fmax=0.05)`), which is a common convergence threshold in solid-state simulations. The atomic trajectory during the relaxation process was recorded in `'relax.traj'`.

To evaluate and visualize the progress of the structure relaxation process, the potential energy at each optimization step was extracted from the trajectory file generated by the BFGS optimizer (`relax.traj`). Using ASE's `read()` function, all atomic configurations along the relaxation path were loaded. For each configuration, the total potential energy was calculated and plotted using `matplotlib` (FIGURE 3). After relaxation the atomic position change into new coordinate with lowest potential energy as can be show using `alcu.positions`

```
import matplotlib.pyplot as plt
from ase.io import read

traj = read('relax.traj', index=':')
energies = [atoms.get_potential_energy() for atoms in traj]

plt.plot(energies, marker='o')
plt.xlabel('Iteration Step')
```

```
plt.ylabel('Potensial Energy(eV)')
plt.title('AlCu Structure Relaxation')
plt.grid(True)
plt.show()
```

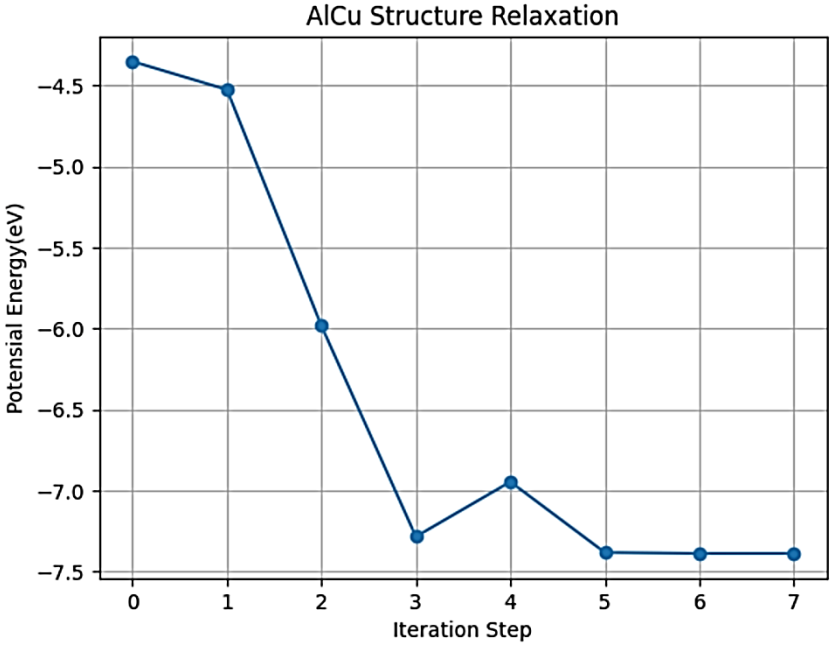

FIGURE 3. Plot of potential energy for strcutre relaxation in each iteration

After completing the structural relaxation, the band structure of the AlCu alloy was calculated to investigate its electronic properties. First, the ground-state total energy of the relaxed structure was obtained to confirm the stability of the configuration. The converged wavefunction and electron density were then saved to a .gpw file using the GPAW calculator. This file serves as the input for a non-self-consistent (NSCF) band structure calculation.

```
alcu.get_potential_energy()
calc.write('alcu_gs.gpw')
```

To compute the band structure, the GPAW `fixed_density()` method was used, which reuses the previously obtained charge density and bypasses the full self-consistent field (SCF) iteration. This approach reduces computational time while maintaining accurate band energy evaluation. The band energies were calculated along a high-symmetry path in the Brillouin zone (Γ–X–M–Γ–R), using 60 evenly spaced k-points. A total of 28 bands were included in the calculation, with symmetry operations disabled to allow for flexible path definitions. The results were stored in a BandStructure object, which was later used to generate plots and analyze features such as band dispersion and the presence or absence of a band gap.

```
calc = GPAW('alcu_gs.gpw').fixed_density(
    nbands=28,
    symmetry='off',
    kpts={'path': 'GXMGR', 'npoints': 60},
    convergence={'bands': 8})

bs = calc.band_structure()
```

The band structure was visualized using the `bs.plot()` function, which produced a graph of energy eigenvalues along the selected k-point path. The plot was saved as a PNG image file and displayed interactively using Matplotlib. The energy window was set from –5.0 eV to 10.0 eV to clearly show both occupied and unoccupied states relative to the Fermi level (FIGURE 4).

```
bs.plot(filename='bandstructure.png', show=True, emin=-5.0, emax=10.0)
```

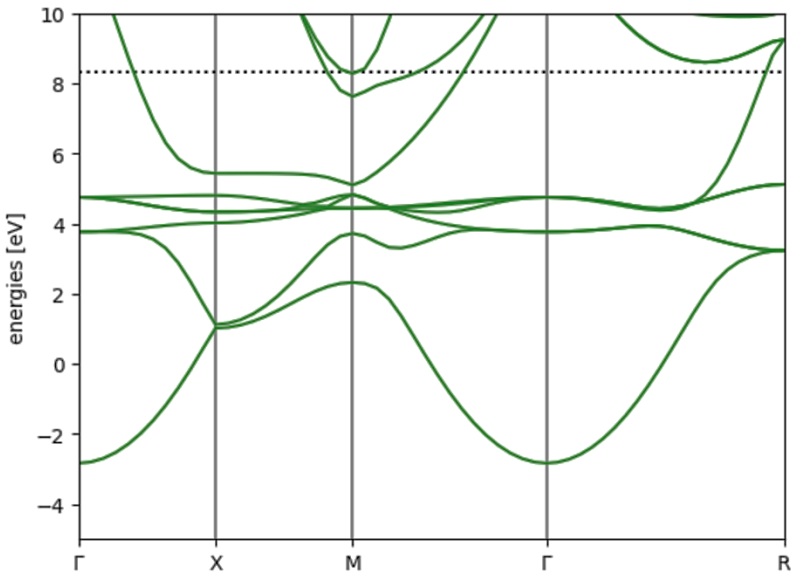

FIGURE 4. Visualization of AlCu bandstructure from DFT calculation using GPAW

Based on FIGURE 4, it is evident that the Fermi level intersects multiple energy bands. FIGURE 4 indicates that the AlCu alloy exhibits metallic behavior, with no band gap separating the valence and conduction bands. Electrons can move freely across the Fermi level, supporting electrical conductivity. Furthermore, the band curvature near the Fermi level provides qualitative insight into the effective mass of charge carriers: flatter bands indicate heavier carriers, while steeper bands indicate lighter carriers. We have compared our findings with experimental data and results from conventional DFT software to ensure the accuracy and reliability of the band structure results obtained using GPAW. For instance, a study by Hüser et al. (2013) demonstrated that band gaps calculated using the G0W0@LDA approximation in GPAW are in excellent agreement with experimental values for a range of semiconductors and insulators. This validation supports the credibility of the GPAW simulations used in our study. This plot serves as a powerful educational tool, enabling students to connect abstract quantum mechanical concepts with visual representations of electronic properties. It also provides a practical demonstration of how crystal symmetry and atomic arrangement influence electronic band structure.

Additionally, the cloud-based GPAW simulations executed through Google Colab are highly efficient, making them ideal for classroom use. The running times for the key stages of the simulation process demonstrate the efficiency of the cloud-based GPAW simulations. The installation time was 1 minute and 1 second, allowing students to set up the necessary environment quickly. Once the setup was complete, the structure creation process took only 7 milliseconds, ensuring a fast start. The structural relaxation step required 51.2 seconds, while generating the relaxation plot took just 220 milliseconds. Finally, the band structure calculation was completed in 40 seconds. These swift processing times enable students to efficiently run simulations and observe results, making the approach highly suitable for educational purposes.

Educational Effectiveness

The integration of this simulation into a Google Colab notebook was highly effective for educational purposes. Students can access the content with a single click and run it entirely in their browser. The clear code structure, inline explanations, and visual outputs (e.g., 3D atomic structures, band structure plots) promote interactive and intuitive learning (Vallejo, 2022; Bravenec, 2023). Importantly, the notebook is structured to support both guided instruction and independent exploration. The structured format of the notebook guides students through key concepts in computational materials science from lattice construction and structural relaxation to band structure analysis. Each section includes Python scripts with comments and outputs that provide immediate visual feedback. This supports the understanding of electronic structure theory and enhances computational thinking, an increasingly important skill in modern science and engineering education.

Furthermore, modifying parameters such as lattice constants, atomic positions, or doping elements allows students to engage in inquiry-based learning (Gholam, 2019). By observing how these changes affect energy levels and band structures, students gain a deeper understanding of structure–property relationships in materials. The visualization tools provided by ASE and Matplotlib reinforce these lessons by converting abstract numerical results into intuitive plots, helping students connect quantum mechanics with tangible electronic behavior.

From a pedagogical perspective, this approach aligns well with the constructivist learning model, where knowledge is actively built through exploration and experimentation (Zhang et al., 2025). It supports differentiated instruction, allowing more advanced students to delve into the Python source code or test custom material configurations, while others follow the step-by-step workflow. Instructors can also use the notebook as a formative assessment tool by assigning tasks such as simulating different alloy compositions or comparing calculated band gaps (Koksalan, 2024).

Classroom Usability and Student Engagement

The designed simulation notebook demonstrates strong potential for usability in classroom environments and actively promotes student engagement. Since the entire simulation is hosted on Google Colab, it bypasses the technical challenges often faced when introducing scientific software in school or university labs (Di Felice, 2023; Vallejo, 2022). Students only need a Google account and internet access to begin running simulations, which significantly reduces setup time and encourages participation regardless of the type of device or operating system used. This accessibility makes it ideal for both in-person instruction and remote learning settings.

From a practical standpoint, the notebook's linear structure, divided into logically sequenced cells covering installation, structure creation, relaxation, and band structure visualization, supports ease of use. Each code block is accompanied by explanations, which help students understand what the code does and why it is necessary in the workflow. This structure aligns well with instructional scaffolding, allowing educators to build on students' knowledge progressively without overwhelming them (Elmansi, 2019; Cheng, 2024).

Student engagement is further enhanced through interactivity and visualization. The ability to view atomic structures in 3D, modify them, and instantly see changes in band structure fosters a sense of experimentation and discovery. This is particularly effective in reinforcing concepts such as crystal symmetry, metallic vs. semiconducting behavior, and the effect of atomic doping. Students can run short simulations, visualize outcomes, and hypothesize relationships between structure and electronic properties within a single class session.

Moreover, the notebook use of real research tools such as ASE, GPAW, and Python provides students with early exposure to industry-relevant methods (Mortensen, 2024), enhancing their motivation by demonstrating how abstract physics concepts are applied in cutting-edge materials science. This hands-on approach has received positive feedback from educators, who note that students appreciate the practical nature of the work, in contrast to more theoretical lessons. Overall, the notebook fosters active learning, promotes curiosity, and supports diverse learning paths, all contributing to a deeper understanding and improved retention of complex physics and materials science topics.

CONCLUSION

We have developed learning tools using Google Colab to teach computational materials science, specifically utilizing GPAW and ASE for simulating and analyzing the electronic band structure of AlCu materials. By removing technical barriers, these cloud-based notebooks allow students to easily access and run simulations with just a Google account and internet access, providing hands-on experience with fundamental research tools. The notebook's structured format supports active learning, instructional scaffolding, and differentiated instruction, fostering deeper understanding and engagement with complex physics concepts. This approach enhances motivation by demonstrating the real-world application of theoretical knowledge, making it an effective tool for both in-person and remote learning environments. The notebook used for teaching is accessible at the following link: Google Colab Notebook. This resource provides a practical, step-by-step guide for students to simulate and analyze the band structure of materials, offering interactive learning with immediate visual feedback, which strengthens their understanding of materials science principles.

REFERENCES

Bravenec, A.D. & Ward, K.D. (2023). Interactive Python Notebooks for Physical Chemistry. *Journal of Chemical Education*, 100(2), pp. 933–940. doi: https://pubs.acs.org/doi/10.1021/acs.jchemed.2c00665

Bylaska, E.J. et al. (2024). Electronic Structure Simulations in The Cloud Computing Environment. *Journal of Chemical Physics*, 161(15), p. 150902.

Cheng, R., Barik, T., Leung, A., Hohman, F., & Nichols, J. (2024). BISCUIT: Scaffolding LLM-Generated Code with Ephemeral UIs in Computational Notebooks. *arXiv*, doi: https://doi.org/10.48550/arXiv.2404.07387

Dejam, S. (2023). ZnO, Cu-doped ZnO, Al-doped ZnO and Cu-Al doped ZnO thin films: Advanced micro-morphology, crystalline structures and optical properties. *Materials Science in Semiconductor Processing*, 126, p. 105748. doi: https://doi.org/10.1016/j.rinp.2023.106209

Di Felice, R. et al. (2023). A Perspective on Sustainable Computational Chemistry Software Development and Integration. *Journal of Chemical Theory and Computation*, 19(20), pp. 7056–7076. doi: https://doi.org /10.1021/acs.jctc.3c00419

Elmansi, H., Mostafa, E., El-Sayed, M., & Qoura, A. (2019). Using a Computer-based Scaffolding Strategy to Enhance EFL Preparatory Stage Students' Reading Skills and Self-Regulation. *Journal of Research in Curriculum, Instruction and Educational Technology*, 5(1), pp. 111–134. Available at: https://www.researchgate.net/publication/338019866

Gholam, A. (2019). Inquiry-Based Learning: Student Teachers' Challenges and Perceptions. *Journal of Inquiry & Action in Education*, 10(2), pp. 112–133. Available at: https://files.eric.ed.gov/fulltext/EJ1241559.pdf

Gu, J., Bai, J., Zhu, Y., Qin, Y., Gu, H., Zhai, Y. & Ma, P. (2016). First-principles study of the influence of doping elements on phase stability, crystal and electronic structure of Al_2Cu (θ) phase. *Computational Materials Science*, 111, pp. 328-333. doi: https://doi.org/10.1016/j.commatsci.2015.09.049

Hrubeš, J., Jaroš, A., Nemirovich, T., Teplá, M. & Petrželová, S. (2024). Integrating Computational Chemistry into Secondary School Lessons. *Journal of Chemical Education*, 101(6), pp. 2343–2353. doi: https://doi.org/10.1021/acs.jchemed.3c00908

Hüser, F., Olsen, T., & Thygesen, K. S. (2013). Quasiparticle GW calculations for solids, molecules, and 2D materials. *Journal of Physics: Condensed Matter*, 25(35), p. 353202. doi: https://doi.org/10.1103/PhysRevB.87.235132

Ju, F. et al. (2024). Acceleration without Disruption: DFT Software as a Service. *Journal of Chemical Theory and Computation*, 20(24), pp. 10838–10851. Available at: https://pubs.acs.org/doi/10.1021/acs.jctc.4c00940

Koksalan, S. & Ogan-Bekiroglu, F. (2024). Examination of Effects of Embedding Formative Assessment in Inquiry-Based Teaching on Conceptual Learning. *Science Insights Education Frontiers*, 20(2), pp. 3223–3246. Available at: https://files.eric.ed.gov/fulltext/EJ1416927.pdf

Lehtola, S. & Karttunen, A. J. (2022). Free and Open Source Software for Computational Chemistry Education. *WIREs Computational Molecular Science*, 12(5), p. e1610. Available at: https://wires.onlinelibrary.wiley.com/doi/10.1002/wcms.1610

Li, J., Huang, Y., Zhang, X., & Zhang, L. (2024). Influence of Mg Doping on the Structure and Mechanical Properties of Al_2Cu Precipitated Phase by First-Principles Calculations. *Materials*, 17(1), p. 93. doi: https://www.mdpi.com/1996-1944/17/1/93

Ma, Z., Ren, F., Ming, X., Long, Y., & Volinsky, A.A. (2019). Cu-Doped ZnO Electronic Structure and Optical Properties Studied by First-Principles Calculations and Experiments. *Materials*, 12(1), p. 196. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC6337601/

Magers, B., Chávez, V.H., & Peyton, B.G. (2021). PSI4EDUCATION: Free and Open-Source Programming Activities for Chemical Education with Free and Open-Source Software. in *Teaching Programming across the Chemistry Curriculum, American Chemical Society*, pp. 107–122. doi: https://pubs.acs.org/doi/abs/10.1021/bk-2021-1387.ch008

Mortensen, J.J. et al. (2024). GPAW: An open Python package for electronic structure calculations. *Journal of Chemical Physics*, 160(9). doi: https://doi.org/10.1063/5.0182685

Sabolsky, E.M., Mena, J.A., Mendoza-Estrada, V., González-Hernández, R., Sabolsky, K., & Sierras, K. (2025). Doping Effects on Multivalence States, Electronic Structure, and Optical Band Gap in $LaCrO_3$ under Varied Atmospheres: An Integrated Experimental and Density Functional Theory Study. *ACS Applied Electronic Materials*, 7(6). doi: https://doi.org/10.1021/acsaelm.4c02359

Vallejo, W., Uribe, C.D., & Fajardo, C. (2022). Google Colab and Virtual Simulations: Practical e-Learning Tools to Introduce Python Programming and Density Functional Theory. *ACS Omega*, 7(8). doi: https://pubs.acs.org/doi/10.1021/acsomega.2c00362

Wang, Y., Tang, W., Liu, J., & Zhang, L. (2015). Stress-induced anomalous shift of optical band gap in Ga-doped ZnO thin films: Experimental and first-principles study. *Applied Physics Letters*, 106(16), p. 162101. doi: https://doi.org/10.1063/1.4918933

Wei, Z., Chen, J., Xue, J., Qu, N., Liu, Y., Sun, L., Xiao, Y., Wu, B., Zhu, J., & Tang, H. (2024). Investigation of the Influence of Alloy Atomic Doping on the Properties of Cu-Sn Alloys Based on First Principles. *Metals*, 14(5), p. 552. doi: https://doi.org/10.3390/met14050552

Zhang, W. & Zhou, Y. (2025). Research on Differentiated Teaching Model of Python Programming Based on Learning Data Analysis. *Journal of Computer and Technology Education Research*, 1(3). doi: https://doi.org/10.70767/jcter.v1i3.405