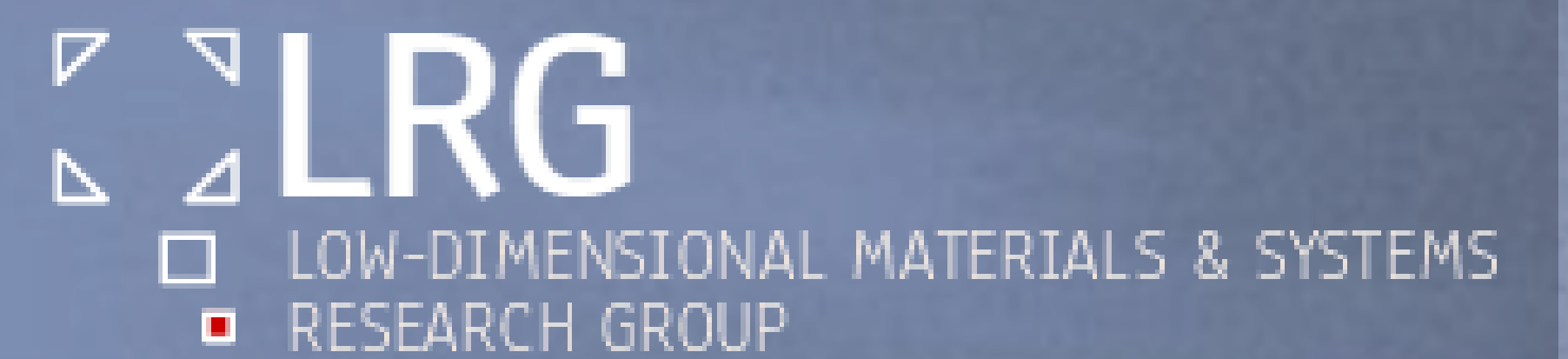


gpaw-tools: UI/GUI scripts for GPAW software

B. Sarikavak-Lisesivdin, S. B. Lisesivdin

Gazi University, Faculty of Science, Department of Physics, 06500 Teknikokullar, Ankara, Turkiye



GPAW is a density functional theory (DFT) code based on the Projector Augmented Wave (PAW) method written primarily in Python. It has high integration with ASE (Atomic Simulation Environment) software and Python language. However, this high level of integrity requires knowledge of the Python language and therefore limits the number of GPAW users worldwide. In this study, we are reporting a free cross-platform Python scripts called *gpaw-tools* for an end user who is not interested in the Python language itself and is interested in pure materials science. The aim of *gpaw-tools* scripts are to offer user interface (UI) and Tcl/Tk based graphics user interface (GUI) software. With *gpaw-tools*, the end user can calculate, view and save the elastic properties, state of density (DOS), partial DOS (PDOS), band structure, charge densities and optical properties (both RPA and BSE) of the studied structure without knowledge of Python language. The *gpaw-tools* scripts are free software and distributed with MIT license.

1. Introduction

Density Functional Theory (DFT) is one of the most successful methods for calculating the electronic, structural, optical and many other properties of one-dimensional (1D), 2D and 3D periodic atom systems and molecules [1]. It is a widely used method as it gives accurate results for metals, semiconductors and insulators. Due to the long history of the DFT method [2], there are many different codes written in Fortran, most of which are relatively old. Without a programming language knowledge to use DFT codes, basic Linux usage and BASH command language knowledge are required, so simpler approaches are needed to reach a wider user base. Some graphical user interface (GUI) solutions are presented to solve this problem [3,4]. These GUIs can generate input files for the client-server model and then use it or run it as a command, these input files can be processed, and the results collected and presented.

GPAW, which is a finite difference DFT code in real space [5] based on the projector augmented wave method (PAW) [6]. GPAW differs from these commonly used DFT codes because it is written in mostly Python and C programming languages. GPAW operations can be performed powerfully with Python scripts (as a library), as commands, and even with Jupyter [7] notebooks. In addition to the standard ground state DFT calculations in GPAW, extensions such as time dependent DFT [8], Bethe-Salpeter equation [9], GW approximation [10] can be performed. However, since GPAW is primarily used as a Python library, it is important to write a proper script and all these powerful features of GPAW can be enjoyed simply by browsing GPAW's extensive documentation.

This proceeding reports new high-level UI/GUI scripts that can be used to interact with GPAW to perform required calculations in a simple and neat way. These scripts are simply called *gpaw-tools* and are written in Python 3. Tkinter, the standard Python interface of the Tcl/Tk GUI toolkit, is used for the GUI portion of the *gpaw-tools*.

2. Implementation

2.1. Architecture

gpaw-tools is an integration of Python scripts for simplifying interatomic potential calculations, GPAW calculations, and some DFT related optimizations. It also has a graphical user interface for the GPAW calculations. The Python scripts of *gpaw-tools* mostly build on the Atomic Simulation Environment (ASE) and GPAW. The pseudo-relation between ASE, GPAW and *gpaw-tools* is shown in Figure 1. This is also the logo of *gpaw-tools* as well. In addition to these two important software, *gpaw-tools* uses As Soon As Possible (ASAP), KIM-API, kimpy and Elastic packages for the related calculations. The relationships between the *gpaw-tools* and these packages are shown in Fig. 2.

2.2. Usage

2.2.1 Interatomic Potential-based Optimization

Batch configuration of atoms must be provided in the script as a CIF file. The universal LJ potential of Elliott and Akerson is used as a general potential that can be used for all elements of the periodic table [11].

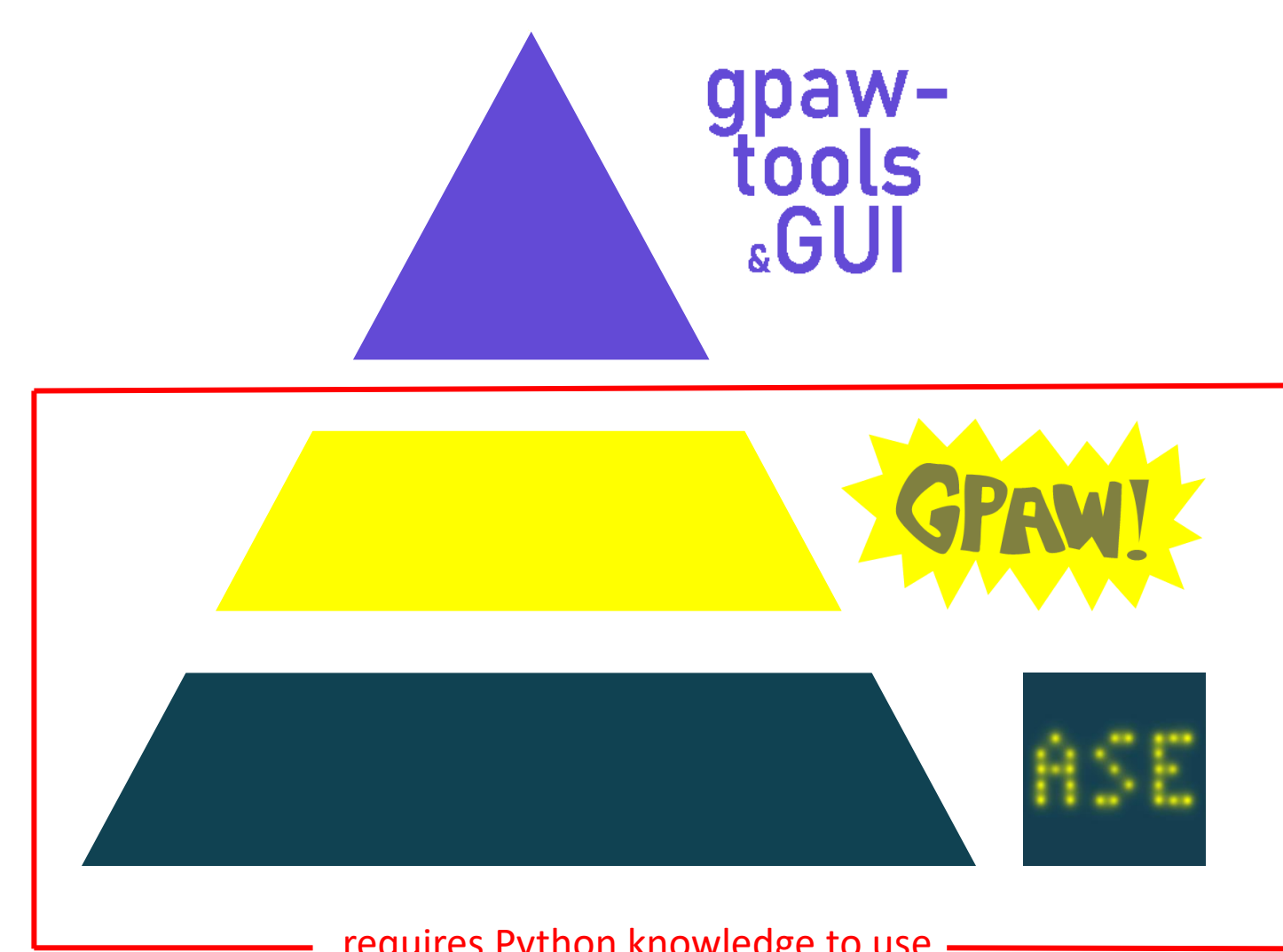


Figure 1, The *gpaw-tools* is positioned over GPAW and ASE in terms of being user friendly and easy to learn.

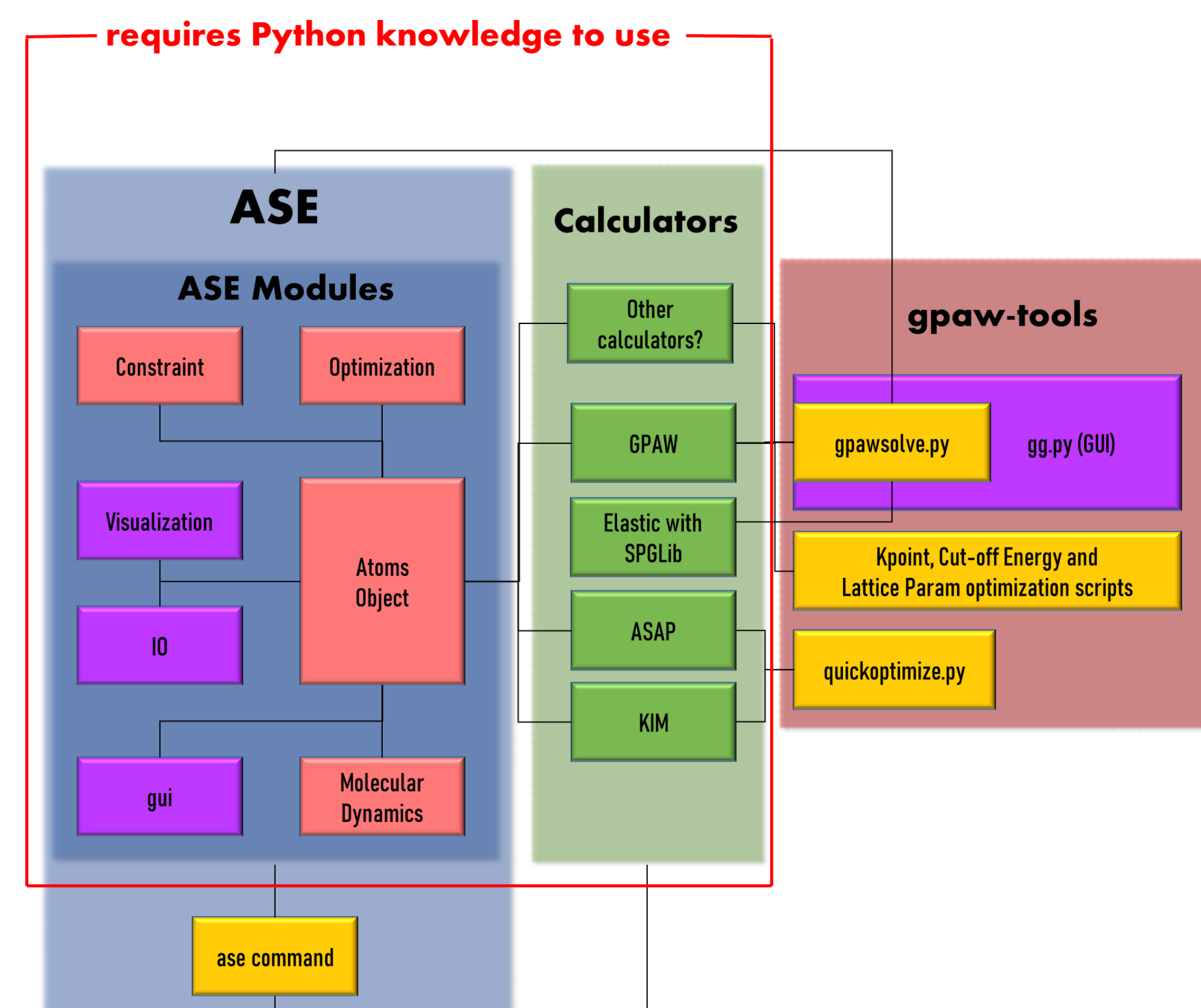


Figure 2 General relationships between ASE, calculators and *gpaw-tools* package.

However, the full potential of the OpenKIM project can be used [12]. The script can be called from the command line like this:

```
python quickoptimize.py <input file.cif>
```

2.2.2 Main solver script: "gpawsolve.py"

The main script *gpawsolve.py* can be run as a command. There is a GUI called *gg.py* which uses *gpawsolve.py* for calculations.

Figure 3 shows a screenshot of *gg.py*. To use it as a system-wide program, the user must add the *gpaw-tools-main* folder to the *\$PATH* variable in the configuration file, which is a well-known process. The program can be called from the command line in the following ways:

```
gpawsolve.py -v -h -o -i <input file.py> -g <geometry file.cif>
```

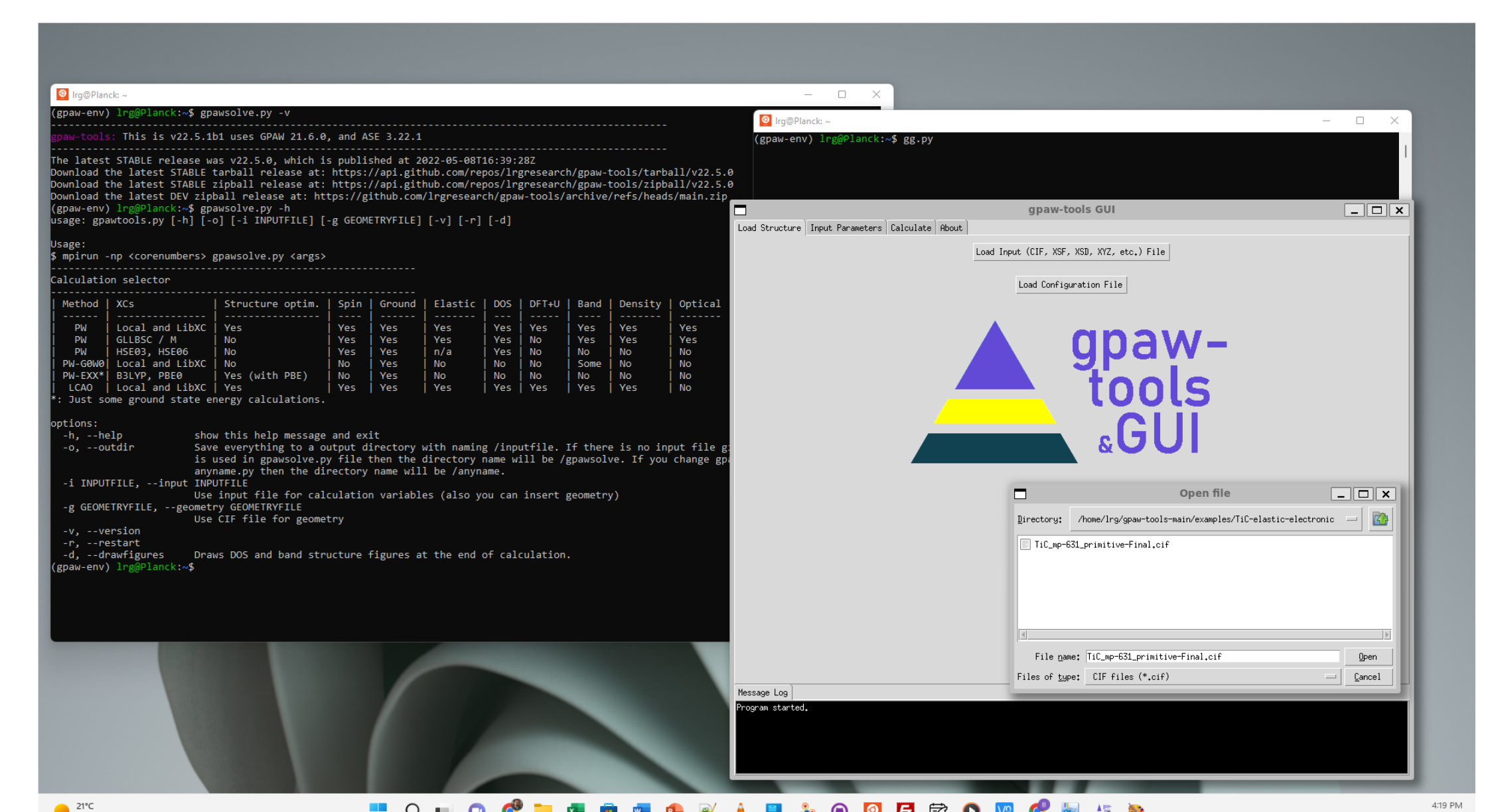


Figure 3, The GUI of *gpaw-tools* called *gg.py* in action.

Multiple arguments can be passed to the program

```
-g, -geometry: Use CIF file as input structure  
-i, -input: Use an input file for variables (input.py).  
-o, -outdir: Store all results in an output directory with the name of the input file  
-h -help: usage help.  
-v -version: version information of running GPAW and ASE code.
```

Since GPAW extensively uses the Message Passing Interface (MPI) programming model for parallel execution, *gpawsolve.py* can also be executed in parallel. To invoke MPI, the program can be called from the command line with one of the commands,

```
mpirun -np <kernel_number> gpawsolve.py <args>
```

3. Conclusion

gpaw-tools is a new tool for researchers who want to reduce complexity and improve performance when using GPAW in their work. It can also be used for educational purposes as it offers a higher level of user interaction.

References

- [1] W. Kohn, L.J. Sham, Self-Consistent Equations Including Exchange and Correlation Effects, Phys. Rev. 140 (4A) (1965) A1133 – A1138,
- [2] R.O. Jones, Density functional theory: Its origins, rise to prominence, and future, Rev. Mod. Phys. 87 (3) (2015) 897 – 923,
- [3] Burai v1.3.2. Available online: <https://github.com/BURAI-team/burai>
- [4] QuantumVITAS v0.3.0. Available online: <https://github.com/quantumVITAS/quantumVITAS>
- [5] J.R. Chelikowsky, N. Trouiller, Y. Saad, Finite-Difference-Pseudopotential Method - Electronic-Structure Calculations without a Basis, Phys. Rev. Lett. 72 (8) (1994) 1240 – 1243,
- [6] P.E. Blöchl, C.J. Först, J. Schimml, Projector augmented wave method: ab initio molecular dynamics with full wave functions, B. Mater. Sci. 26 (1) (2003) 33 – 41,
- [7] J. F. Pimentel, L. Murta, V. Braganholo and J. Freire, A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks, 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), 2019, pp. 507-517.
- [8] J. Yan, J.J. Mortensen, K.W. Jacobsen, K.S. Thygesen, Linear density response function in the projector augmented wave method: Applications to solids, surfaces, and interfaces, Phys. Rev. B 83 (2011), 245122.
- [9] J. Yan, K.W. Jacobsen, K.S. Thygesen, Optical properties of bulk semiconductors and graphene/boron nitride: The Bethe-Salpeter equation with derivative discontinuity-corrected density functional energies, Phys. Rev. B 86 (2012), 045208.
- [10] C. Rostgaard, K.W. Jacobsen, K.S. Thygesen, Fully self-consistent GW calculations for molecules, Phys. Rev. B 81 (2010), 085103.
- [11] R.S. Elliott, A. Akerson, Efficient universal shifted Lennard-Jones model for all KIM API supported species, OpenKIM, v003, 2015.
- [12] E.B.Tadmor, R.S. Elliott, J.P. Sethna, R.E. Miller, C.A. Becker, The potential of atomistic simulations and the Knowledgebase of Interatomic Models, JOM 63 (2011) 17.

For more:

<https://lrgresearch.org/gpaw-tools>