

Supercell-core software: A useful tool to generate an optimal supercell for vertically stacked nanomaterials

Cite as: AIP Advances **10**, 105105 (2020); <https://doi.org/10.1063/5.0023984>

Submitted: 03 August 2020 • Accepted: 12 September 2020 • Published Online: 02 October 2020

Tomasz Necio and  Magdalena Birowska



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

[Hybrid functionals based on a screened Coulomb potential](#)

The Journal of Chemical Physics **118**, 8207 (2003); <https://doi.org/10.1063/1.1564060>

[Quantum ESPRESSO toward the exascale](#)

The Journal of Chemical Physics **152**, 154105 (2020); <https://doi.org/10.1063/5.0005082>

[Siesta: Recent developments and applications](#)

The Journal of Chemical Physics **152**, 204108 (2020); <https://doi.org/10.1063/5.0005077>

AIP Advances
Biophysics & Bioengineering Collection

Read Now!

Supercell-core software: A useful tool to generate an optimal supercell for vertically stacked nanomaterials

Cite as: AIP Advances 10, 105105 (2020); doi: 10.1063/5.0023984

Submitted: 3 August 2020 • Accepted: 12 September 2020 •

Published Online: 2 October 2020



View Online



Export Citation



CrossMark

Tomasz Necio and Magdalena Birowska^{a)} 

AFFILIATIONS

Faculty of Physics, University of Warsaw, Pasteura 5, 00-092 Warsaw, Poland

^{a)}Author to whom correspondence should be addressed: Magdalena.Birowska@fuw.edu.pl

ABSTRACT

Vertically oriented materials, such as van der Waals heterostructures, which have novel hybrid properties, are crucial for fundamental scientific research and the design of new nano-devices. Currently, most available theoretical methods require applying a supercell approach with periodic boundary conditions to explore the electronic properties of such nanomaterials. Herein, we present supercell-core software, which provides a way to determine the supercell of non-commensurate lattices, in particular van der Waals heterostructures. Although this approach is very common, most of the reported work still uses supercells that are constructed “by hand” and on a temporary basis. The developed software is designed to facilitate finding and constructing optimized supercells (i.e., with small size and minimal strain accumulation in adjacent layers) of vertically stacked lattices. The developed software tool is accessible as open-source free software.

© 2020 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0023984>

I. INTRODUCTION

When a 2D crystal is placed on top of another layer (substrate), either it can adjust its position (e.g., by rotating to follow the periodic potential of the substrate), resulting in a commensurate state, or the layers can exhibit a small lattice mismatch, in which case the interlayer binding energy (via weak van der Waals forces) can compensate for the increase in the elastic energy. Both of these occurrences may lead to experimentally observed superperiodic structures, which are commonly known as moiré patterns. Such moiré superlattices have been widely studied in the context of many different systems, including bilayer graphene,^{1–4} trilayer graphene,⁵ and bilayer black phosphorus,⁶ as well as for many heterostructures, such as graphene on hexagonal boron nitride (hBN)^{7,8} or bilayer transition metal dichalcogenides (TMDCs).^{9,10}

Hybrid structures consisting of different types of layers connected via weak van der Waals forces represent a new class of hybrid crystals known as van der Waals heterostructures.¹¹ These structures are of broad interest to many researchers throughout the world due to the novel hybrid properties arising in these

materials, which are distinct from their individual layer components.^{12–14} Many of these properties can be precisely controlled by rotating the two stacked atomic layers with respect to one another,^{4,6,13} highlighting the fact that manipulating this unique “twist angle” degree of freedom can allow for control of the nanoscale properties of such nanomaterials.^{13,15}

The strain itself is the subject of a new research field in solid state physics, called straintronics,¹⁶ in which strain engineering methods are used to develop next-generation devices for information, sensors, and energy-saving technologies. The strain-induced physical effects in nanolayers or heterostructures, such as changes in the band structure, or electronic, optical, or magnetic properties,^{17–19} are fundamentally important. Understanding these relationships is a prerequisite for developing new technologies using such nanomaterials. The strain distribution can be investigated experimentally using a variety of experimental methods,^{8,20} such as atomic force microscopy (AFM), scanning tunneling microscopy (STM), and/or Raman spectroscopy.

Proper modeling is required in order to better understand the phenomena and underlying physics behind the structure–property relationships. The modeling of these types of

nanomaterials mostly focuses on electronic band structure calculations, and one of the most accurate *ab initio* methods is based on density functional theory (DFT). Widely used software packages, such as VASP,^{21,22} Quantum Espresso,²³ and SIESTA,²⁴ are limited in terms of their supercell approaches because only a few hundred atoms can be considered. However, stacking adjacent layers with different lattice parameters or different lateral crystal symmetries may allow for appropriate modeling of superperiodicity in very large lateral cells or detection of aperiodic structures. Incommensurate lattices are outside the scope of the present paper. One approach directed toward describing aperiodic layered structures can be found in Ref. 5. In the current work, we discuss a developed approach based on commensurate lattices. We propose supercell software capable of finding the optimal supercell, i.e., those with small size and low strain distribution. Specifically, we have developed a software package that searches through all possible superperiodicities arising from multiples of primary cells for a given rotation angle between the top and bottom 2D lattices. The software allows for determination of the optimal “magic angles” between the adjacent vertically stacked layers and the resulting moiré patterns.

There are few available builders that can handle the construction of supercells. To the best of our knowledge, those that are freely available, such as VESTA,²⁵ only enable *ad hoc* construction by hand without finding the optimal supercell. The other capable programs, such as Quantum Wise,²⁶ which have an appropriately implemented builder, must be purchased. Herein, we present the package supercell-core, which is free of charge and allows the user to find the optimal supercell for a given twist angle and “n” number of layers constituting the van der Waals structures. To the best of our knowledge, none of the above-mentioned codes calculate the strain distribution of adjacent layers for a particular angle between the layers.

In this article, we first present the mathematical background for the software, along with explanations of the applied algorithms and discussion of relevant technical details. Then, the code is briefly described, focusing on its functionalities. Finally, we present the practical applications of the software and provide several examples.

II. COMPUTATIONAL METHODOLOGY

A. Mathematical description

Our methodology is based on a commensurability condition, which requires long-range order in the sets of lattice planes at the interfaces of *n* vertically stacked layers.

In order to formulate this condition mathematically, we can consider two planar lattices: A (bottom layer; substrate) and B (top layer), where lattice B is placed on top of lattice A. We denote the unit vectors of both lattices as \mathbf{a}_i and \mathbf{b}_i , respectively, where $i \in \{1, 2\}$. The task is to find two new vectors, \mathbf{c}_i , which are commensurate to both \mathbf{a}_i and \mathbf{b}_i ,²⁷ such that

$$\exists m_{ij} \in \mathbb{Z} \wedge \exists n_{ij} \in \mathbb{Z} : \mathbf{c}_i = \sum_{j=1}^2 (m_{ij})^T \mathbf{a}_j = \sum_{j=1}^2 (n_{ij})^T R_\theta \mathbf{b}_j, \quad (1)$$

where R_θ is the 2D rotation matrix based on the rotation angle between layers, θ [Fig. 1(a)]. Depending on the specific case, θ might

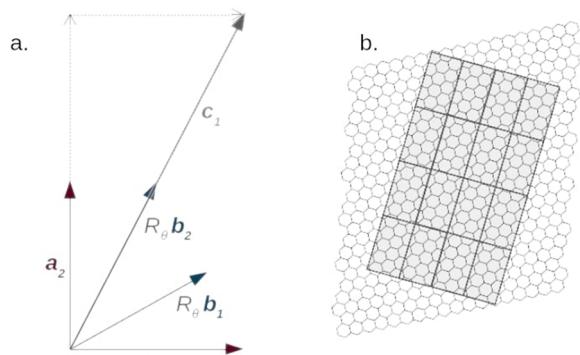


FIG. 1. (a) Schematic construction of a superlattice vector, $\mathbf{c}_1 = 2\mathbf{a}_2 + \mathbf{a}_1 = 2R_\theta\mathbf{b}_2$. (b) Example of two incommensurate lattices (hexagonal and rectangular).

be fixed, or it might be a free parameter. The angle, θ , is hereafter referred to as the twist angle.

In principle, there may be no set of integers, m_{ij} , n_{ij} , which satisfies both of the above equations, even if θ is treated as a free parameter [Fig. 1(b)]. Thus, the commensurability condition is enforced by applying strain to the top layer. This corresponds to a linear transformation of the B lattice’s elementary cell. Note that layer A (the substrate) is always unstrained. The new task is to find the linear transformation that introduces minimal strain into the system.

For now, let us consider a fixed value of θ . We denote the modified vectors of the rotated lattice B as $\tilde{\mathbf{b}}_i$. Then, we can define its strain tensor, ε , as a 2D matrix

$$\sum_{j=1}^2 (\varepsilon + I)^{ij} R_\theta \mathbf{b}_j = \tilde{\mathbf{b}}_i. \quad (2)$$

The m_{ij} coefficients can easily be calculated given the n_{ij} and $\tilde{\mathbf{b}}_i$ values. We must find a matrix, \tilde{B} , for every set of four numbers, n_{ij} , such that $\tilde{B}(\mathbf{b}_i) = \tilde{\mathbf{b}}_i$. Then, from all sets of n_{ij} , we choose the one with minimal strain. To quantify the strain, we define the norm $L_{(1,1)}(\varepsilon)$, which is equal to $\sum_{ij} |\varepsilon_{ij}|$.

Note that to find a nearly unstrained supercell for a given value of θ , the software searches all possible values of n_{ij} (up to some maximum value, defined by the `max_el` parameter in the code) in order to be certain that the optimal supercell is determined. However, this approach (hereafter referred to as the Direct algorithm) has a time complexity of $O(N^4)$, where N is the upper limit of n_{ij} that is considered. However, the problem of finding a nearly unstrained supercell can be greatly simplified. It is important to note that when the strain is zero, then $\mathbf{b}_i = \tilde{\mathbf{b}}_i$ [see Eq. (2)]. For every pair of n_{i1} , n_{i2} , it is possible to write independent linear equations for m_{i1} , m_{i2} ,

$$(\mathbf{a}_1 \ \mathbf{a}_2)^{-1} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} (\mathbf{b}_1 \mathbf{b}_2) \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}. \quad (3)$$

The “quality” of each n_{ij} pair can be assessed by taking a measure of how far the corresponding m_{ij} value is from an integer

solution $\frac{\text{v-round}(\mathbf{v})}{\|\mathbf{v}\|}$, where $\mathbf{v} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}$. This algorithm, known as the Fast algorithm, effectively describes $O(N^2)$ in time complexity and gives the same results as a direct search for cases where an optimal supercell's strain is zero or near-zero. It can be used to find the moiré patterns or nearly unstrained unit cells. However, the Fast algorithm does not guarantee an optimal solution for cases where an optimized supercell is significantly strained. Our analysis indicates that both algorithms give the same results when the norm $L_{(1,1)}(\epsilon)$ does not exceed a value of 0.015 (see Fig. 2). Thus, it guarantees that the strain calculated with the Fast algorithm is not overestimated.

These problems trivially extend to cases where θ is a free parameter, by treating it as an additional degree of freedom when searching the parameter space. When considering multiple layers, B_1, B_2 , etc., we can exploit the fact that all lattices must ultimately be commensurate, and we do not modify lattice A at all. Effectively, each layer can be fit to the same lattice A supercell. Therefore, we can repeat most of the above steps for each layer independently and record the calculated "qualities" of every set of m_{ij} values for each layer. To select the final supercell (which is determined by its m_{ij} coefficients), we then use the criterion of the lowest sum of the norm $L_{(1,1)}(\epsilon)$ for all B_i layers.

B. Implementation

The code is implemented as a Python package using the Python numerical library, NumPy.²⁸ Every 2D crystal and every heterostructure are represented by a Python object. Optimization of the strain for the given values of the twist angle, θ , for each layer is carried out by first preparing a list of all possible combinations of those values. Then, for each combination, an array of all possible vectors of the supercell in the substrate lattice vectors basis is prepared, which generates an $N \times N \times 2$ array, where each two-dimensional vector

stored in that array is a grid point in the substrate lattice vector basis. The value of N describes the extent to which we check $N = 2n + 1$ (where n is the highest integer that can be accepted as the n_{ij} coefficient). Certain basic symmetries are utilized to decrease the number of required calculations (decreasing the number of n_{ij} values, e.g., the symmetry with respect to interchanges \vec{c}_1 and \vec{c}_1 , rotation about 180°).

In the Fast algorithm ("fast" option), the strain tensor is calculated for all possible \mathbf{c}_i vectors, and $L_{(1,1)}(\epsilon)$ is analyzed to assess the quality of the specific configuration in a vectorized operation. Ultimately, for a given combination of interlayer angles, the configuration that gives the lowest strain is chosen. In the minimization process, smaller, less narrow cells are treated preferentially. If the Direct algorithm is applied by the user, the procedure follows the mathematical description of our solution for cases in which the desired strain is near-zero [see Eq. (3)].

For drawings, the code uses the Matplotlib Python library.²⁸ An optional dependency of the library is pandas,²⁹ which is applied if the user wants to log intermediate steps of the strain minimization procedure.

III. DESCRIPTION OF THE CODE

The software is available in the official Python package index as supercell_core. Assuming that a distribution of Python software is installed, a user can download the package with the following command: pip install supercell_core. The recommended way to use the package in a Python script, Python console, or Jupyter notebook is via an import supercell_core as the sc statement. The source code is also available from Ref. 30 in the form of a Git repository. The repository contains example files, described in Sec. IV in the supercell_core/examples subdirectory, and a README file with usage examples and a link to online documentation. The package documentation is also provided interactively in the Python console via Python's built-in help function.

The software enables the user to find the optimal configuration for an arbitrary number of vertically stacked layers, and for any set of acceptable interlayer angles between them. The parameter $L_{\max}(A) = \max |m_{ij}|$ can be used to control the trade-off between computational cost, the quality of the resulting supercell, and its size (max_el in the code, refers to the substrate layer denoted A). Supercell-core accepts definitions of the lattices either performed manually using the lattice function or read from a VASP POSCAR file with the read_POSCAR command. For convenience, users are encouraged to use supercell-core with a Jupyter notebook or an interactive Python console.

The user can control the positions of the layers in the z -direction by changing the z -components of the unit cell vectors of the layers. Note that the strain distribution is only calculated in lateral coordinates. This software stacks unit cells of the layers directly on top of each other. For example, if one layer with unit cell vectors $(a, b, 0), (c, d, 0), (0, 0, h)$ contains an atom at position (x, y, z_1) , and the layer directly above it contains an atom at (x, y, z_2) , then the distance between these atoms in the calculated heterostructure will be $h - z_1 + z_2$.

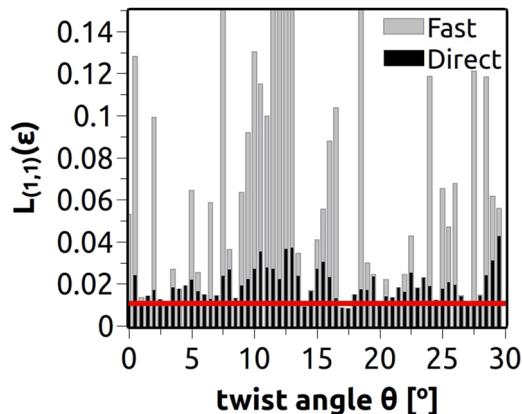


FIG. 2. Comparison of the Fast and Direct algorithms implemented in the supercell-core software. The norm of the strain tensor is calculated for an optimal supercell for a phosphorene/graphene heterostructure as a function of the twist angle, θ . The red line indicates the value of $L_{(1,1)}(\epsilon) = 0.015$, for which both algorithms give the same results. Note that for the highly strained lattices, the discrepancy between the algorithms can be substantial, and in those cases, the use of the Direct algorithm is recommended.

It is important to note that the inclusion of different stacking configurations in primary cells of the layers can be achieved by translating the atoms via a corresponding vector in one of the cells. The described approach constructs the supercell independent of the stacking configurations of the layers, since it is only based on the lattice vectors. Thus, the atomic positions are correspondingly scaled and translated in a new supercell.

Supercell optimization and strain calculation methods can be performed on a Heterostructure object. Choices related to the optimization algorithm, the maximum n_{ij} value, and whether to store intermediate results for each combination of interlayer angles in a log are conducted by specific arguments in the opt method. All results of these calculations are stored in Result objects. Saving results to XCrysDen XSF and/or VASP POSCAR files is possible using methods for the Lattice object. Every Result object contains a superlattice method, which returns a Lattice object corresponding to the heterostructure (containing the correct positions of atoms in all layers of the new supercell). If logging is enabled, the log can be saved to a CSV file. When this option is used, for each twist angle, the code lists in the order the following details: twist angles “theta_0” etc. in radian unit (at the end in °); the sum of norm $L_{(1,1)}(\epsilon)$ “max_strain”; lateral size in Å² “supercell_size”; matrices describing the vectors of the supercell on the basis of substrate and Cartesian coordinates denoted “M,” “N,” etc.; “supercell_vectors” [$\vec{c}_1 = (\text{supercell_vectors_11}, \text{supercell_vectors_21})$, $\vec{c}_2 = (\text{supercell_vectors_12}, \text{supercell_vectors_22})$, given in Å]; strain tensor ϵ_{ij} denoted “strain_tensor_layer” for each layer B_i ; and the number of atoms “atom_count.” If there are multiple layers, the user obtains a table with a row for every combination of twist angles. The list takes the form of a pandas.DataFrame file, which makes it easy to view, transform, and save the data (e.g., to the CSV format).

The package is also able to draw the positions of atoms in the supercell (projected onto an xy plane) using the Matplotlib plotting library.

IV. EXAMPLES

Here, we present three examples that demonstrate the capabilities of the developed software and verify the correctness of the obtained results. Specifically, bilayer graphene (BG), trilayer graphene (TG), and hBN/graphene/phosphorene are considered.

The examples included in this section are available in the code’s GitHub repository in the examples directory.³⁰

A. Moiré patterns in bilayer graphene (BG)

Here, we examine the construction of the supercell of bilayer graphene, which is a prototypical example of a bilayer system. For this reason, we use a two-atom unit cell of graphene with lattice vectors $\vec{a}_1 = (\frac{\sqrt{3}a}{2}, -\frac{a}{2})$ and $\vec{a}_2 = (\frac{\sqrt{3}a}{2}, \frac{a}{2})$, assumed for both layers, where $a = 2.46$ Å is a lattice constant of graphene. One of the graphene layers can be rotated with respect to the other by a relative angle, θ (twist angle). Supposing we want to find the twist angles that lead to the appearance of moiré patterns, we can use the Heterostructure.opt method from the code and set the algorithm parameter to moiré. It is also recommended to increase the value of $L_{max}(A)$ (e.g., max_el to 20 or 50) in order to catch patterns that are only visible at

long distance, and to use a small step for the values of θ (e.g., 0.025°). Then, the user can apply the log functionality to assess which angles are most promising (for details, see Sec. III).

In this example, we set max_el to 20 and the range for the twist angle as $\theta \in (0^\circ, 30^\circ)$, with the step equal to 0.1°. The optimal supercells were found by applying these parameters, and the maximum strain tensor components of these supercells are presented in Fig. 3. Particular angles exist when the supercells are small (tens to hundreds of atoms) and have low strain values (<0.03%). Some of the lowest-strain angles are collected in Table I. In the case where two identical layers with a twist angle equal to 0° are considered, the strain tensor is exactly zero, and the generated supercell is identical to the primary cell. Because of this obvious result, we do not include this case in Table I. Note that the lattice mismatch is commonly calculated as the difference between the lattice constants, even for rotated layers with strain distributions that are different from those determined by this simple approximation. Therefore, supercells resulting from the construction of such twisted layers are assumed to exhibit low strain values, although, in fact, the strain components (e.g., shear strains) can be substantial and can influence the material’s electronic properties. The number of atoms in the substrate layer can be calculated using the equation $\det(M) \times N_{at}$, where N_{at} is the number of atoms in the primary cell and M is equal to $M = (m_{ij})$. The number of atoms in other layers can be calculated in an analogous manner.

The results (Table I) demonstrate that for small twist angles, the size of the supercell is large and the number of atoms is high, whereas for large twist angles, the situation is reversed. These results confirm a well-known fact observed for twisted bilayer structures where the relevant length scale is on the order of $1/\theta$.³¹ In particular, for small twist angles such as 1.1° or 2°, the periodicity becomes large enough that it cannot be handled as lattice inputs for DFT codes. However, other potential approaches, such as tight binding (TB) methods,³² can be used instead. Moreover, our

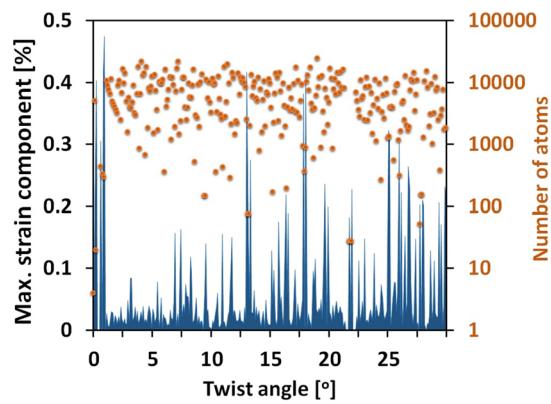


FIG. 3. Absolute values of the maximum strain tensor component and the number of atoms are presented for the supercells of bilayer graphene as a function of the twist angle. All supercells are generated using the developed software with the resolution angle $\Delta\theta = 0.1^\circ$ up to 30° and an $L_{max}(A) \leq 20$ parameter. For clarity, only the results with the maximum strain component $\leq 0.5\%$ are presented. Visible dips in the blue bar plot correspond to configurations with moiré patterns. Note that there are many twist angles for which the number of atoms is in the hundreds.

TABLE I. Structural information corresponding to bilayer graphene supercells generated by *supercell-core* software. The results are presented for the optimal supercells (small system size and nearly unstrained layers) and for particular twist angles between the graphene layers. The supercells of bilayer graphene have been generated with twist angle resolution 0.1° and $L_{\max}(\text{Å}) \leq 20$ parameters. Note that the supercell vectors can be calculated from $\vec{c}_1 = m_{11}\vec{a}_1 + m_{21}\vec{a}_2$ and $\vec{c}_2 = m_{12}\vec{a}_1 + m_{22}\vec{a}_2$ or from the 2D rotation matrix and n_{ij} parameters [see Eq. (1)]. The m_{ij} , n_{ij} , ε_{ij} matrices are taken from *log* files. Only the maximal strain tensor component is presented.

Twist angle θ (deg)	Substrate layer (A) $M = (m_{ij}) \in \mathbb{Z}^{2 \times 2}$	Top layer (B) $N = (n_{ij}) \in \mathbb{Z}^{2 \times 2}$	Supercell size (\AA^2) $ \vec{c}_1 \times \vec{c}_2 \times \sin(\angle(\mathbf{c}_1, \mathbf{c}_2))$	Max. strain component (B_1) $\varepsilon_{ji} = (\varepsilon_{ij})^T$	Supercell No. atoms
1.1	$\begin{pmatrix} 30 & 60 \\ -60 & -29 \end{pmatrix}$	$\begin{pmatrix} 29 & 60 \\ -60 & -30 \end{pmatrix}$	$127.8 \times 127.8 \text{\AA}^2 \times \sin(61.1^\circ)$	$\varepsilon_{xx} = \varepsilon_{yy} = 0.027\%$	10 920
2.0	$\begin{pmatrix} 50 & 33 \\ -49 & -16 \end{pmatrix}$	$\begin{pmatrix} 49 & 33 \\ -50 & -17 \end{pmatrix}$	$121.8 \times 70.3 \text{\AA}^2 \times \sin(30^\circ)$	$\varepsilon_{xy} = -0.01\%, \varepsilon_{yx} = 0.01\%$	3 268
3.9	$\begin{pmatrix} 9 & 17 \\ -17 & -8 \end{pmatrix}$	$\begin{pmatrix} 8 & 17 \\ -17 & -9 \end{pmatrix}$	$36.2 \times 36.2 \text{\AA}^2 \times \sin(60^\circ)$	$\varepsilon_{xy} = 0.019\%, \varepsilon_{yx} = -0.019\%$	868
6.0	$\begin{pmatrix} 12 & 37 \\ 21 & 57 \end{pmatrix}$	$\begin{pmatrix} 21 & 62 \\ 12 & 31 \end{pmatrix}$	$130.7 \times 107.5 \text{\AA}^2 \times \sin(1.9^\circ)$	$\varepsilon_{xy} = -0.019\%, \varepsilon_{yx} = 0.019\%$	364
17.9	$\begin{pmatrix} 12 & 37 \\ 21 & 57 \end{pmatrix}$	$\begin{pmatrix} 21 & 62 \\ 12 & 31 \end{pmatrix}$	$71.2 \times 201.7 \text{\AA}^2 \times \sin(1.9^\circ)$	$\varepsilon_{yx} = -0.006\%$	372
21.8	$\begin{pmatrix} 59 & 55 \\ 13 & 12 \end{pmatrix}$	$\begin{pmatrix} 73 & 68 \\ -16 & -15 \end{pmatrix}$	$163.5 \times 152.2 \text{\AA}^2 \times \sin(0.1^\circ)$	$\varepsilon_{yx} = -0.025\%$	28
27.8	$\begin{pmatrix} 9 & 11 \\ 42 & 47 \end{pmatrix}$	$\begin{pmatrix} 33 & 38 \\ 21 & 23 \end{pmatrix}$	$116 \times 131.2 \text{\AA}^2 \times \sin(0.8^\circ)$	$\varepsilon_{xy} = -0.008\%$	156
29.4	$\begin{pmatrix} 19 & 49 \\ -14 & -31 \end{pmatrix}$	$\begin{pmatrix} 14 & 39 \\ -19 & -46 \end{pmatrix}$	$42 \times 105.6 \text{\AA}^2 \times \sin(6.6^\circ)$	$\varepsilon_{yx} = -0.028\%$	388

supercell-core software successfully identifies the angles that have been reported previously based on STM experiments, specifically 21.8%¹ and 1.1%¹³ for bilayer systems. In addition, we present the visualization of the optimal supercell generated for the twist angle of 6° , as shown in Fig. 4, where the moiré pattern is clearly visible along with various stacking configurations. The different stacking configurations of bilayer graphene and their crucial impact on the electronic properties have been highlighted in many papers so far

(see, e.g., Ref. 33); however, for a twisted layer, many stacking configurations are automatically included within one supercell (see, e.g., Fig. 4).

B. Trilayer systems: Trilayer graphene (TG) and hBN/graphene/phosphorene

The natural extension of the bilayer system is to simply add another layer. In principle, our code allows the user to find the optimal supercell for n layers exhibiting different types of symmetries.

Thus, in this example, we consider a trilayer van der Waals heterostructure consisting of hexagonal boron nitride, graphene, and phosphorene. This structure is significantly more complicated than the previous example. Although graphene and hBN share the same hexagonal symmetry and have similar lattice constants (2.46 Å and 2.52 Å, respectively), phosphorene has a rectangular lateral cell with lattice constants equal to $a = 3.26$ Å and $b = 4.35$ Å [optimized lattice constants obtained in local-density approximation (LDA) for the monolayer taken from Ref. 14] (see Fig. 5). These differences between the layers make finding an optimal supercell more difficult.

The supercell-core software can be used to find the optimal configuration for this example with any combination of twist angles. The layer strains resulting from the generated supercells are presented in Fig. 6 as a function of two twist angles, θ_1 and θ_2 . These angles (θ_1, θ_2) correspond to the relative rotation of the graphene (layer B_1) and phosphorene (layer B_2) layers with respect to the

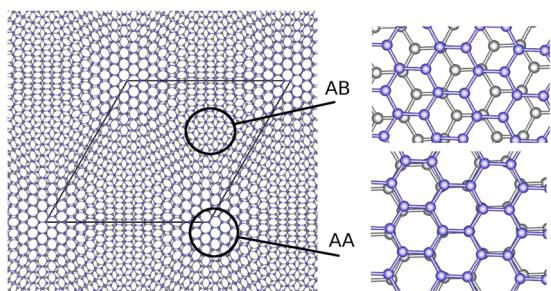


FIG. 4. The moiré pattern forms on bilayer graphene with a relative rotation angle (twist angle) between the layers equal to $\theta = 6^\circ$. A unit cell determined by the software supercell-core is marked in blue and contains 364 atoms. Various local stacking configurations are clearly visible. AA and AB stackings are shown in zoom-in view on the right side of the picture. Gray and violet points indicate the carbon atoms from the bottom and top layers, respectively.

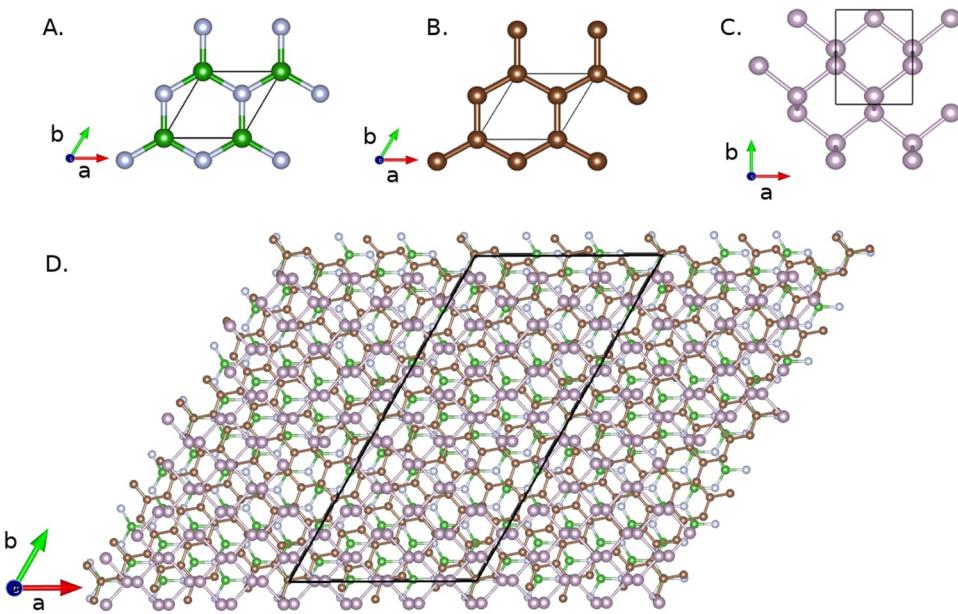


FIG. 5. Primary cells of the (A) hBN, (B) graphene, and (C) phosphorene layers used for constructing the optimal van der Waals trilayer hBN/G/P supercell and (D) an example of an optimal supercell for a twisted trilayer hBN/G/P structure with twist angles equal to $\theta_1 = 10.9^\circ$ and $\theta_2 = 29.9^\circ$, corresponding to graphene and phosphorene rotations with respect to the hBN layer.

hBN layer (layer A), respectively. The dark navy blue regions in Fig. 6 indicate the set of angles that correspond to supercells with relatively low strains. Table II provides the details regarding a few generated supercells that contain hundreds of atoms. The results clearly show that the strains in these supercells are relatively high. This is because we constrained the supercell size [$L_{max}(A) \leq 10$] so that the number of atoms is less than one thousand, while our system is composed of three incommensurate lattices. This is a fundamental problem of the system, and not a problem of the program. To

find better supercells, one can loosen the constraints on the supercell size, although this introduces the risk of finding supercells that are too big for DFT calculations. In order to minimize the strain, it is recommended that the user increases the value of the $L_{max}(A)$ parameter. By increasing this parameter, one directly enlarges the supercell size considered and, thus, minimizes the strain (see the last raw results in Table II). Increasing the resolution of the twist angle may also help, in principle, but there is little chance that an optimal supercell would not be found with a resolution on the order of 0.1° .

In Table II, we also present selected supercells for trilayer graphene (TG). Note that whenever one of the twist angles is zero, the results correspond to the bilayer system (first raw results in Table II for $\theta_1 = 0^\circ$, $\theta_2 = 21.8^\circ$). However, the software found a different result than presented for the bilayer system (see Table I for $\theta_1 = 21.8^\circ$). When supercell-core finds a true moiré pattern, any parallelogram with vertices on the repeated patterns' nodes is a zero-strain supercell. The program should then find the smallest of such supercells. However, the calculated strain for each of those supercells is non-zero because of floating point errors. When choosing the optimal supercell, the program has some fixed tolerance (ϵ) of the variation in the strain value, but sometimes, it happens to be too low to find the smallest supercell from a set of equally good supercells. A potential solution is to re-run the calculations for low strain supercells with a lower value of the $L_{max}(A)$ parameter.

Generally, for cases where $n > 3$, it is difficult to obtain the commensurability condition for a relatively small system size, especially the one with different types of layers that exhibit different symmetries.

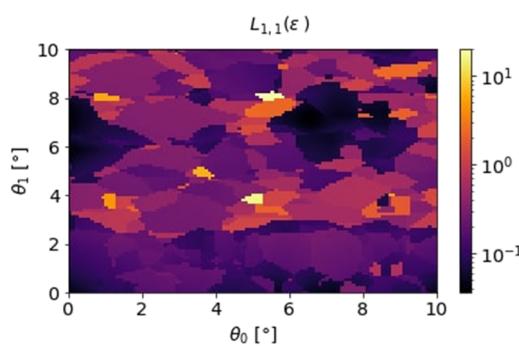


FIG. 6. Strain distribution for an hBN/graphene/phosphorene heterostructure as a function of the twist angles, θ_0 and θ_1 . For the clarity of the presentation, each angle is presented within the range (0° , 10°) and step size of 0.1° . The black color indicates the range of angles with the least amount of strain.

TABLE II. Structural information corresponding to the optimal supercells generated by *supercell-core* software for the trilayer systems: trilayer graphene (TG) and hexagonal boron nitride/graphene/phosphorene (hBN/G/P). The supercells of trilayer graphene have been generated with twist angle resolution equal to 0.1° and $L_{\max}(A) \leq 80$. Only the maximal strain tensor component is presented. The twist angles θ_1, θ_2 correspond to relative rotation of the graphene and phosphorene layers with respect to the hBN layer, respectively. For the clarity of presentation, only one of the maximal strain components is presented; however, in some of the cases presented here, $|\epsilon_{xy}| = |\epsilon_{yx}|$ or $|\epsilon_{xx}| = |\epsilon_{yy}|$.

System $L_{\max}(A)$	Twist angles θ_1 (deg), θ_2 (deg)	Bottom layer (A) $M = (m_{ij}) \in \mathbb{Z}^{2 \times 2}$	Middle layer (B ₁) $N = (n_{ij}) \in \mathbb{Z}^{2 \times 2}$	Top layer (B ₂) $N' = (n'_{ij}) \in \mathbb{Z}^{2 \times 2}$	Supercell size (Å ²) $ \vec{c}_1 \times \vec{c}_2 \times \sin(\angle(\mathbf{c}_1, \mathbf{c}_2))$	Max. strain component layer B ₁ ; layer B ₂ (%)	No. atoms
TG	0, 21.8	$\begin{pmatrix} 12 & 19 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 12 & 19 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 15 & 23 \\ -3 & -6 \end{pmatrix}$	$33.8 \times 50.8 \text{ \AA}^2 \times \sin(3.7^\circ)$	$0; \epsilon_{yx} = -0.005$	126
TG	21.8, 17.9	$\begin{pmatrix} 19 & 16 \\ 10 & -3 \end{pmatrix}$	$\begin{pmatrix} 26 & 17 \\ -1 & -9 \end{pmatrix}$	$\begin{pmatrix} 25 & 17 \\ 1 & -8 \end{pmatrix}$	$62.8 \times 36.2 \text{ \AA}^2 \times \sin(30^\circ)$	$\epsilon_{yx} = -0.03; \epsilon_{yx} = -0.005$	1302
TG	6.0, 27.9	$\begin{pmatrix} 17 & 11 \\ -16 & -5 \end{pmatrix}$	$\begin{pmatrix} 16 & 11 \\ -17 & -6 \end{pmatrix}$	$\begin{pmatrix} 11 & 10 \\ -19 & -9 \end{pmatrix}$	$40.6 \times 23.5 \text{ \AA}^2 \times \sin(30^\circ)$	$\epsilon_{xy} = -0.019; \epsilon_{yx} = -0.2$	546
hBN/G/P	6.7, 0.8	$\begin{pmatrix} 4 & 6 \\ -10 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 7 \\ -10 & 2 \end{pmatrix}$	$\begin{pmatrix} -4 & 6 \\ -4 & -1 \end{pmatrix}$	$21.8 \times 19.9 \text{ \AA}^2 \times \sin(115.7^\circ)$	$\epsilon_{xx} = 1.7; \epsilon_{yx} = -1.9$	403
$L_{\max}(A) \leq 10$							
hBN/G/P	10.9, 29.9	$\begin{pmatrix} 6 & 6 \\ -3 & 6 \end{pmatrix}$	$\begin{pmatrix} 6 & 8 \\ -4 & 4 \end{pmatrix}$	$\begin{pmatrix} 0 & 7 \\ -3 & -3 \end{pmatrix}$	$13 \times 26 \text{ \AA}^2 \times \sin(60^\circ)$	$\epsilon_{xx} = 0.11; \epsilon_{xy} = -1.31$	302
$L_{\max}(A) \leq 10$							
hBN/G/P	11.5, 29.8	$\begin{pmatrix} 8 & 6 \\ 9 & -3 \end{pmatrix}$	$\begin{pmatrix} 11 & 6 \\ 6 & -4 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 \\ -4 & -3 \end{pmatrix}$	$36.9 \times 13 \text{ \AA}^2 \times \sin(62^\circ)$	$\epsilon_{yy} = -0.92; \epsilon_{yx} = 0.23$	437
$L_{\max}(A) \leq 10$							
hBN/G/P	23.1, 4.4	$\begin{pmatrix} 13 & 17 \\ 17 & -30 \end{pmatrix}$	$\begin{pmatrix} 23 & 6 \\ 6 & -29 \end{pmatrix}$	$\begin{pmatrix} 20 & -10 \\ 0 & -13 \end{pmatrix}$	$65.2 \times 65.2 \text{ \AA}^2 \times \sin(120^\circ)$	$\epsilon_{xx} = 0.03; \epsilon_{yy} = 0.09$	3802
$L_{\max}(A) \leq 40$							

V. CONCLUSIONS

In this report, we present the novel supercell-core software that has been developed to facilitate the construction of the supercell consisting of vertically aligned 2D layers. The package is available as a Python module³⁰ and it is an open-source free software, licensed under the GNU General Public License v3.0. In principle, the code is designed to handle structures comprised of “n” different types of layers exhibiting different lateral symmetries. The developed software allows the user to find the optimal supercell, with the smallest number of atoms and lowest strains experienced by adjacent layers. The methodology is based on a commensurability condition, which implies long-range crystalline order in the sets of lattice planes that make up the van der Waals heterostructure. This condition is enforced by applying strain to the top layers of the structures. In the described approach, the bottom layer is always unstrained.

The software works with POSCAR files, and it is therefore compatible with the DFT software VASP and with the widely used visualization software VESTA. In addition, there are two algorithms that are implemented, namely, Direct and Fast. The former spans all possible configurations from which the supercells can be constructed, while the latter is more efficient and designed to search for unstrained configurations (e.g., those exhibiting moiré patterns). The former is more hefty and resource-intensive, while the latter is a more efficient algorithm.

Depending on the user’s requirements, the developed software also enables construction of the optimal supercells based on the twist angle(s) between the layers. It also allows users to study strained supercells in order to examine the impact of the strain distribution on the various properties of van der Waals heterostructures. Both types of results can be used as structural input for further calculations based on an *ab initio* approach (using DFT software, such as VASP, Quantum Espresso, or SIESTA). Particularly for cases of large supercells containing thousands of atoms, the results provided by our software can be used as structural input for software based on tight binding methods.

ACKNOWLEDGMENTS

This work was funded by the National Science Centre, Poland, Grant No. UMO-2016/23/D/ST3/03446. We acknowledge access to the computing facilities of PL-Grid: Polish Infrastructure for Supporting Computational Science in the European Research Space and the Interdisciplinary Centre for Modelling (ICM), University of Warsaw. We made use of computing resources of TU Dresden ZIH within the project *TransPheMat*.

DATA AVAILABILITY

The data that support the findings of this study are openly available at <https://github.com/tnecio/supercell-core> and can also be found at <https://www.fuw.edu.pl/~birowska>.

REFERENCES

- ¹E. Cisternas, M. Flores, and P. Vargas, "Superstructures in arrays of rotated graphene layers: Electronic structure calculations," *Phys. Rev. B* **78**, 125406 (2008).
- ²T. Ohta, J. T. Robinson, P. J. Feibelman, A. Bostwick, E. Rotenberg, and T. E. Beechem, "Evidence for interlayer coupling and moiré periodic potentials in twisted bilayer graphene," *Phys. Rev. Lett.* **109**, 186807 (2012).
- ³B. Hunt, J. D. Sanchez-Yamagishi, A. F. Young, M. Yankowitz, B. J. LeRoy, K. Watanabe, T. Taniguchi, P. Moon, M. Koshino, P. Jarillo-Herrero, and R. C. Ashoori, "Massive Dirac fermions and Hofstadter butterfly in a van der Waals heterostructure," *Science* **340**, 1427–1430 (2013).
- ⁴Y. Cao, V. Fatemi, A. Demir, S. Fang, S. L. Tomarken, J. Y. Luo, J. D. Sanchez-Yamagishi, K. Watanabe, T. Taniguchi, E. Kaxiras, R. C. Ashoori, and P. Jarillo-Herrero, "Correlated insulator behaviour at half-filling in magic-angle graphene superlattices," *Nature* **556**, 80–84 (2018).
- ⁵Z. Zhu, P. Cazeaux, M. Luskin, and E. Kaxiras, "Modeling mechanical relaxation in incommensurate trilayer van der Waals heterostructures," *Phys. Rev. B* **101**, 224107 (2020).
- ⁶P. Kang, W.-T. Zhang, V. Michaud-Rioix, X.-H. Kong, C. Hu, G.-H. Yu, and H. Guo, "Moiré impurities in twisted bilayer black phosphorus: Effects on the carrier mobility," *Phys. Rev. B* **96**, 195406 (2017).
- ⁷R. Decker, Y. Wang, V. W. Brar, W. Regan, H.-Z. Tsai, Q. Wu, W. Gannett, A. Zettl, and M. F. Crommie, "Local electronic properties of graphene on a bn substrate via scanning tunneling microscopy," *Nano Lett.* **11**, 2291–2295 (2011).
- ⁸C. R. Woods, L. Britnell, A. Eckmann, R. S. Ma, J. C. Lu, H. M. Guo, X. Lin, G. L. Yu, Y. Cao, R. V. Gorbachev, A. V. Kretinin, J. Park, L. A. Ponomarenko, M. I. Katsnelson, Y. N. Gornostyrev, K. Watanabe, T. Taniguchi, C. Casiraghi, H.-J. Gao, A. K. Geim, and K. S. Novoselov, "Commensurate-incommensurate transition in graphene on hexagonal boron nitride," *Nat. Phys.* **10**, 451–456 (2014).
- ⁹J. Kang, J. Li, S.-S. Li, J.-B. Xia, and L.-W. Wang, "Electronic structural moiré pattern effects on MoS₂/MoSe₂ 2D heterostructures," *Nano Lett.* **13**, 5485–5490 (2013).
- ¹⁰E. M. Alexeev, D. A. Ruiz-Tijerina, M. Danovich, M. J. Hamer, D. J. Terry, P. K. Nayak, S. Ahn, S. Pak, J. Lee, J. I. Sohn, M. R. Molas, M. Koperski, K. Watanabe, T. Taniguchi, K. S. Novoselov, R. V. Gorbachev, H. S. Shin, V. I. Fal'ko, and A. I. Tartakovskii, "Resonantly hybridized excitons in moiré superlattices in van der Waals heterostructures," *Nature* **567**, 81–86 (2019).
- ¹¹I. V. Geim and A. K. Grigorieva, "van der Waals heterostructures," *Nature* **499**, 419–425 (2013).
- ¹²J. Kunstmüller, F. Mooshammer, P. Nagler, A. Chaves, F. Stein, N. Paradiso, G. Plechinger, C. Strunk, C. Schüller, G. Seifert, D. R. Reichman, and T. Korn, "Momentum-space indirect interlayer excitons in transition-metal dichalcogenide van der Waals heterostructures," *Nat. Phys.* **14**, 801 (2018).
- ¹³Y. Cao, V. Fatemi, S. Fang, K. Watanabe, T. Taniguchi, E. Kaxiras, and P. Jarillo-Herrero, "Unconventional superconductivity in magic-angle graphene superlattices," *Nature* **556**, 43–50 (2018).
- ¹⁴M. Birowska, J. Urban, M. Baranowski, D. K. Maude, P. Plochocka, and N. G. Szwacki, "The impact of hexagonal boron nitride encapsulation on the structural and vibrational properties of few layer black phosphorus," *Nanotechnology* **30**, 195201 (2019).
- ¹⁵S. Carr, D. Massatt, S. Fang, P. Cazeaux, M. Luskin, and E. Kaxiras, "Twistronics: Manipulating the electronic properties of two-dimensional layered structures through their twist angle," *Phys. Rev. B* **95**, 075420 (2017).
- ¹⁶A. A. Bukharaev, A. K. Zvezdin, A. P. Pyatakov, and Y. K. Fetisov, "Straintronics: A new trend in micro- and nanoelectronics and materials science," *Phys. - Usp.* **61**, 1175–1212 (2018).
- ¹⁷T. Hu, J. Zhou, and J. Dong, "Strain induced new phase and indirect–direct band gap transition of monolayer InSe," *Phys. Chem. Chem. Phys.* **19**, 21722–21728 (2017).
- ¹⁸S. Tamleh, G. Rezaei, and J. Jalilian, "Stress and strain effects on the electronic structure and optical properties of SCN monolayer," *Phys. Lett. A* **382**, 339–345 (2018).
- ¹⁹K. Milowska, M. Birowska, and J. A. Majewski, "Mechanical, electrical, and magnetic properties of functionalized carbon nanotubes," *AIP Conf. Proc.* **1399**, 827–828 (2011).
- ²⁰J. Xue, J. Sanchez-Yamagishi, D. Bulmash, P. Jacquod, A. Deshpande, K. Watanabe, T. Taniguchi, P. Jarillo-Herrero, and B. J. LeRoy, "Scanning tunnelling microscopy and spectroscopy of ultra-flat graphene on hexagonal boron nitride," *Nat. Mater.* **10**, 282–285 (2011).
- ²¹G. Kresse and J. Furth Müller, "Efficiency of *ab initio* total energy calculations for metals and semiconductors using a plane-wave basis set," *Comput. Mater. Sci.* **6**, 15–50 (1996).
- ²²G. Kresse and J. Furthmüller, "Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set," *Phys. Rev. B* **54**, 11169 (1996).
- ²³P. Giannozzi and B. Stefano, "Quantum espresso: A modular and open-source software project for quantum simulations of materials," *J. Phys.: Condens. Matter* **21**, 395502 (2009).
- ²⁴J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal, "The SIESTA method for *ab initio* order-N materials simulation," *J. Phys.: Condens. Matter* **14**, 2745–2779 (2002).
- ²⁵K. Momma and F. Izumi, "VESTA: A three-dimensional visualization system for electronic and structural analysis," *J. Appl. Crystallogr.* **41**, 653–658 (2008).
- ²⁶S. Smidstrup, T. Markussen, P. Vancreveld, J. Wellendorff, J. Schneider, T. Gunst, B. Verstichel, D. Stradi, P. A. Khomyakov, U. G. Vej-Hansen *et al.*, "Quantumatik: An integrated platform of electronic and atomic-scale modelling tools," *J. Phys.: Condens. Matter* **32**, 015901 (2020).
- ²⁷D. Wolf and S. Yip, *Materials Interfaces: Atomic-Level Structure and Properties*, 1st ed. (Chapman & Hall, 1992).
- ²⁸P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. Contributors, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nat. Methods* **17**, 261–272 (2020).
- ²⁹Pandas Development Team, pandas-dev/pandas: Pandas, 2020.
- ³⁰See <https://github.com/tnecio/supercell-core> for the information about the software supercell-core, 2020, the code can be also found at <https://www.fuw.edu.pl/~birowska>.
- ³¹G. Trambly de Laissardière, D. Mayou, and L. Magaud, "Localization of Dirac electrons in rotated graphene bilayers," *Nano Lett.* **10**, 804–808 (2010).
- ³²See <https://www.physics.rutgers.edu/pythtb/index.html> for python Tight Binding (PythTB), 2020.
- ³³M. Birowska, K. Milowska, and J. A. Majewski, "van Der Waals density functionals for graphene layers and graphite," *Acta Phys. Pol. A* **120**, 845–848 (2011).