



ARL-TR-8134 • SEP 2017



Transire, a Program for Generating Solid-State Interface Structures

by Caleb M Carlin and Berend C Rinderspacher

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Transire, a Program for Generating Solid-State Interface Structures

by Caleb M Carlin and Berend C Rinderspacher
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) September 2017		2. REPORT TYPE Technical Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Transire, a Program for Generating Solid-State Interface Structures				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Caleb M Carlin and Berend C Rinderspacher				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-WMM-G Aberdeen Proving Ground, MD 21005-5069				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-8134	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Transire is a new program developed for the generation and manipulation of crystalline, planar interface structures. Requiring only the unit cell coordinates of the crystals and a simple input file, Transire includes tools for manipulating the interface along multiple degrees of freedom with optional support for energy calculations using the LAMMPS and CP2K computational chemistry packages. In addition, Transire incorporates a non-equilibrium Green's function-based electron transport property calculator. Three test cases are presented to demonstrate the usage of Transire: the misorientation of the graphene bilayer, the interface energy as a function of misorientation of copper grain boundaries, and electron transport transmission across the gallium nitride/silicon carbide interface.					
15. SUBJECT TERMS crystalline interface, electron transport, python, computational chemistry, grain boundary					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 38	19a. NAME OF RESPONSIBLE PERSON Caleb M Carlin
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-306-2811

Contents

List of Figures	v
List of Tables	v
1. Introduction	1
2. Design Goals	3
3. Method of Solution	3
3.1 Generating Atomic Structures	4
3.2 Search Options and Interface Generators	6
3.2.1 Surface Search	6
3.2.2 Twist Angle Search	6
3.2.3 Markov Chain Constrained Search	7
3.2.4 Separation Optimizer	8
3.2.5 Molecule Insertion	8
3.3 Electron Transport Properties	8
3.4 Obtaining and Installation	8
3.5 Usage	10
3.6 Output	11
3.7 Limitations	11
4. Results and Discussion	12
4.1 Graphene	12
4.2 Copper Grain Boundaries	14
4.3 GaN/SiC Heterojunction	17
5. Conclusion	21
6. References	23
Appendix. Sample pycp2k Input File	25

List of Symbols, Abbreviations, and Acronyms	28
Distribution List	29

List of Figures

Fig. 1	Determining the supercell representation of a unit cell with full xy periodicity	5
Fig. 2	The number of atoms in parallel graphene sheets interface structures generated by twist rotation using a) Transire and b) an analytical solution for Moiré structures reproduced from Campanera et al.....	14
Fig. 3	Interface energies for the copper-copper grain boundaries with Miller indices pairs 011/100, 011/011, and 001/001. Energies given for interface structures for twist rotations at every 0.1° (dashed line) and the RAS method (triangles). The solid line marks the lower envelope of the RAS results showing a close match with full twist angle search results.	17
Fig. 4	Transmission spectra across GaN 001/SiC 001 interface with 59.9° twist rotation	21

List of Tables

Table 1	Lowest energy structures for GaN/SiC interfaces	18
---------	---	----

INTENTIONALLY LEFT BLANK.

1. Introduction

Developing a better understanding of electron transport (ET) properties at or across crystal interfaces and applying that understanding to a wide range of materials is facilitated by having tools that are both powerful and easy to use. Whether it is a grain boundary or heterojunction, the representation in computational methods, called here an interface structure, is a finite pair of crystals in contact along a plane that are replicated infinitely along one or more axis using periodic boundary conditions. Methods for calculating the properties of an interface structure belong to either classical molecular mechanics (MM) or quantum mechanics (QM) with each having advantages and limitations.

Studies using MM methods excel at predicting the change in properties of interfaces for a wide range of variations on an interface structure. The relatively low cost of MM calculations allows for calculating the interfacial energy of thousands of interface configurations containing hundreds of thousands of atoms within reasonable time frames. To generate and evaluate the many possible interface configurations, US Army Research Laboratory (ARL) researchers use in-house scripts to work with commercial math software.^{1,2} Commercial software with graphical interface builders like MedeA provide many options for generating and customizing interface structures and integrated support for software packages for MM calculations.^{3,4} Molecular mechanics methods, however, are limited by the availability of potential energy force fields for all combinations of the elements included in the calculation and cannot be used to calculate ET properties.

Studies using QM methods are limited by the computational cost of calculations that scale from N^2 to N^8 depending on the method used, where N reflects the size of the system being simulated. Because of the cost limitation, QM studies of interface structures focus on a single or small number of interface configurations that are generated manually or use a program with a graphical interface. For the smaller set of interface structures studied compared to MM studies, QM methods allow for calculating quantum properties including ET properties. In addition, QM methods require only a basis set and pseudopotential file for each element, which are readily available.

Generating interface structures in a way that ensures the correct periodic boundary conditions presents a number of challenges. Stacking any 2 crystals with periodic boundaries parallel to the interface is conceptually simple as long as the unit cells defining the crystals have surfaces that occupy a mutual plane. Any mismatch in the dimensions of the unit cell representation of the full crystal become negligible in the limit of infinite periodicity parallel to the interface plane as long as the actual

crystal structure isn't stretched or compressed. However, programs for performing MM and QM calculations are generally not capable of handling the complicated crystal structures that result from simply stacking 2 mismatched unit cells. Instead, the stacked crystals must be representable as a single Bravais lattice with orthorhombic lattices being the simplest choice to use as a standard. Determining the orthorhombic representation of an arbitrary unit cell is a matter of determining the number of replications of the unit cell along each lattice vector needed so that the difference in the projection of any 2 lattice vectors on the same axis is below some error tolerance value. Matching the dimensions of 2 orthorhombic cells for the purpose of stacking requires determining the minimum number of replications of each cell that is needed parallel to the intended interface plane to reduce the difference in the length of the 2 lattice vectors along the same axis to ensure the mismatch strain is below a chosen threshold. Solving these 2 problems for a set of unit cells allows the construction of a full interface structure that can be used for QM and MM calculations.

The ability to generate interface structures using 2 arbitrary unit cells is important, but alone can only produce a tiny fraction of the possible interface structures formed by combining 2 crystals. New interface structures can be derived from an initial interface structure by varying the cut planes that define the surfaces at the interface and the 3-D rotation, forming a phase space used to characterize interfaces along 5 degrees of freedom (5DOF). Additional degrees of freedom correspond to the translations parallel to the surface plane and the separation between the 2 sides of the interface structure. Incorporating all of these degrees of freedom allows for generating a complete set of interface structures using all possible simple combinations of 2 crystals.

Small perturbations along the 5DOF can have a significant impact on the size of the resulting interface structure. The work of Shallcross et al.⁵ demonstrated for rotations of one graphene sheet in a bilayer that the number of atoms in the resultant structure varies by several orders of magnitude. While such large structures are not a problem for MM methods, QM calculations are limited to only those structures that are small enough for the QM methods used for the calculation. There is no guarantee that the lowest interface energy will correspond with a small interface structure or that a small interface structure will have a high density of coincidence sites.

The gold standard approach to calculating electron transmission involves a divide-and-conquer approach and solving the non-equilibrium Green's Function (NEGF).^{6,7} The interface structure is divided into 3 portions using 2 planes that are parallel to the interface plane defined as the left node, the right node, and the scattering region. The scattering region is chosen to be large enough that the 2 node

regions have no electron overlap and the electron structure of each node closely matches the bulk crystal. The resultant Hamiltonian and overlap matrices are assumed to be block diagonal with all zero elements for the coupling of the 2 node regions. The Hamiltonian and overlap matrix blocks are used to construct the NEGF, which is used to solve the transmission function. Solving for the transmission over a range of energy levels near the Fermi level yields the transmission function that is used to predict the ET behavior of crystalline systems.^{8,9}

2. Design Goals

The push to expand the capabilities of QM methods to include crystal interface problems previously left to MM methods requires the development of new tools. With this in mind, the presented solution must adequately address the following design goals:

- 1) Program that combines the strengths of both MM and QM approaches to studying crystal interfaces.
- 2) Limited and easy-to-use input requirements to minimize the learning curve for users with less experience using computers.
- 3) A modular approach to implementing interface manipulation tools to maximize:
 - a) User control over how and when the tools are used, and
 - b) Easier inclusion of additional tools.
- 4) Support for interfacing with a range of QM- and MM-based programs to provide options for the user and to take advantage of future developments in the computational chemistry community.

3. Method of Solution

Transire*, a program written using the Python programming language, was developed to address the previously listed design goals. Specifically, Transire is a program for the generation and manipulation of planar interface structures and calculating trans-boundary ET properties.

* Transire is Latin for to cross over, to traverse.

3.1 Generating Atomic Structures

The method of generating grain boundary atomic structures for simulations is separated into 3 steps: unit cells, super cells, and interface structures.

- 1) A unit cell is provided by the user as part of the input and can be any shape and composition as long as it forms a complete lattice space under replication.
- 2) A super cell is the orthorhombic representation of the crystal structure that preserves the full periodicity by replicating the unit cell and wrapping the resulting atoms into an orthorhombic cell.
- 3) An interface structure is formed by stacking super cells along the Z-axis and replicating each super cell parallel to the grain boundary to minimize the strain of the 2 super cells.

Generating the super cell corresponding to a unit cell involves solving for the integer multiples of the lattice vectors such that the projection of any 2 lattice vectors onto a shared axis is equal.

Representing the problem as a matrix equation, the lattice matrices \mathbf{A} and \mathbf{B} correspond to the unit cell and super cell, respectively.

$$\mathbf{PA} = \mathbf{B} . \quad (1)$$

The diagonal matrix \mathbf{P} has only integer values (P_{ii}) that satisfy Eqs. 2 and 3.

$$P_{ii}A_{ii} = B_{ii} . \quad (2)$$

$$P_{jj}A_{ji} = B_{ji} . \quad (3)$$

For the projection condition to be satisfied, it is necessary that $B_{ii} = B_{ji}$ for $j \neq i$ and the problem reduces to a direct relationship between the diagonal elements of \mathbf{P} .

$$P_{ii} \frac{A_{ii}}{A_{ji}} = P_{jj} . \quad (4)$$

The elements of \mathbf{P} are solved as follows:

- 1) Define matrix \mathbf{C} equal to lattice matrix \mathbf{A} .
- 2) Select a non-zero nondiagonal element of \mathbf{C} and the diagonal element in the same column.
- 3) Solve for the smallest integer values P_{ii} and P_{ji} that satisfy Eq. 4.

- 4) Determine the least common multiple for the new and old values of P and update P .
- 5) Set C_{ji} equal to 0.
- 6) Repeat until C is diagonal.
- 7) Solve for $B = PA$.

Multiplying across a row corresponds with replicating the unit cell along one of the lattice vectors. The off-diagonal elements being a multiple of the diagonal element in the same column corresponds with a coincidence of the corners of the supercell (Fig. 1). The orthorhombic supercell is created by wrapping the replicated supercell into the cell defined by the set of diagonal lattice vectors.

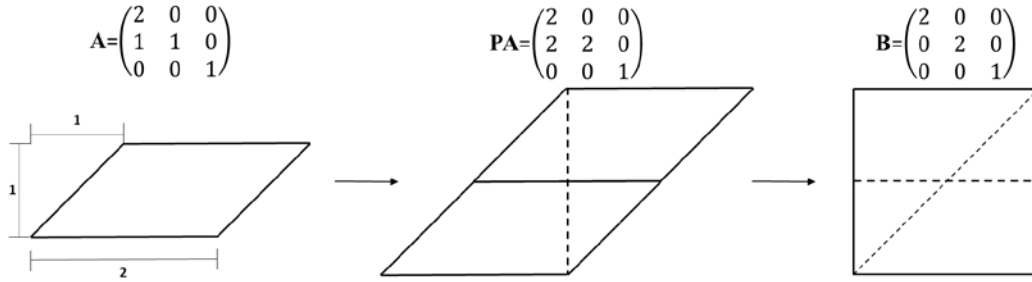


Fig. 1 Determining the supercell representation of a unit cell with full xy periodicity

If A has only integer values, then P can be solved exactly. For non-integer elements of A , P can be solved to within some margin of error (Transire default = 0.05). Once P has been determined, the unit cell is replicated along each lattice vector, the resulting lattice vectors are projected onto the corresponding Cartesian axis, and all atoms outside the new orthorhombic cell are shifted to a matching periodic position within the cell.

The generation of interface structures from super cells requires replicating the super cells parallel to the grain boundary to minimize the differences in the dimensions of the super cells along the x and y dimensions, taking the z axis as normal to the interface plain. The desired relationship between super cells B and C with replication matrices R and S is analogous to Eq. 4.

$$RB = SC \quad (5)$$

As all matrices in Eq. 5 are diagonal and only the lattice constants parallel to the interface need to be matched, the desired solution is the set of smallest integers that satisfy 2 relational equations:

$$R_{11} \frac{B_{11}}{C_{11}} = S_{11} \quad , \quad (6)$$

$$R_{22} \frac{B_{22}}{C_{22}} = S_{22} \quad . \quad (7)$$

Replicating the super cells using the results of Eq. 5 minimizes the amount of strain on the crystal structures when stacking the super cells.

The interfacial energy is defined in Transire as the total energy of interface structure divided by the area of the rectangle formed by the 2 lattice vectors parallel to the interface in units of Angstrom squared. Interfacial energies can be compared between structures with the same number of layers specified in the input as the number of atoms is directly proportional to the surface of the interface.

3.2 Search Options and Interface Generators

Tools are included in Transire for constructing the interface structure incorporating 3 of the 5DOF with support for either specifying the parameters along each degree of freedom or sampling over a range of values for one or more degrees of freedom. Additional tools support minimizing the interface energy by a random walk Markov Chain and numerical optimization of the separation distance between super cells at the interface.

3.2.1 Surface Search

Transire requires a pair of Miller Indices to define the crystal surfaces of the super cells facing the interface in the initial interface structure, but a range of values for each Miller Index can optionally be provided. If more than one value is provided for the indices, then Transire constructs a list of all combinations of surface pairs for the range of indices that satisfy the rules for Miller Indices. All subsequent tools apply to each surface pair separately.

3.2.2 Twist Angle Search

The rotation of one super cell relative to the other around the axis perpendicular to the interface results in a twist misorientation. The twist angle search in Transire generates interface structures corresponding to a set of angles given in the input, or angles in a range given in the input using stepsize, number of steps, and starting angle. The interface structures that are generated can be written to coordinate files with the extended-xyz format for use with other software packages. In addition, the interface energy can be calculated for all interface structures generated with the resultant energies printed in a format for easy plotting.

To automate the process of locating the twist angles that result in interface structures with the fewest atoms, the reducing angle search (RAS) is implemented with the following steps:

- 1) A normal twist angle search is carried out over the range of angles and stepsize given in the input file and the interface structures that were successfully generated are returned.
- 2) A number of interface structures specified in the input (`ras_factor`) and the corresponding twist angles are chosen in order of increasing interface structure size.
 - a) Optionally, all interface structures that are successfully generated are returned along with the corresponding angles (`ras_all_angles`).
- 3) The stepsize is reduced by a factor of 10 and a twist angle search is performed for 10 steps before and after each twist angle returned in the previous step, selecting the smallest interface structure from each angle search.
- 4) Step 3 is repeated the number of times specified in the input (`ras_depth`).
 - a) Optionally, the interface energies of the final list of interface structures can be calculated.

The RAS is a valuable tool for locating and sampling the twist angles that produce interface structures that can be studied within the computational limits of the hardware being used.

3.2.3 Markov Chain Constrained Search

To locate a lower energy interface structure that is near but not accessible by the twist angle search, a Markov Chain search of translations, rotations, or both is implemented in Transire. Translations involve random shifts along the axes parallel to the interface while the rotations are twist rotations around the axis perpendicular to the interface with max translation and rotation in a single step set at 2.0 Å and $\pm\pi$, respectively. If both translations and rotations are used, the perturbation at each step is randomly chosen. No acceptance criteria is used for each step; instead, each step is used as the basis for the following step and the interface structure with the lowest interface energy from all Markov Chain steps is returned at the end of the search. A record of each step is printed along with the overall translation and rotation necessary to reproduce each step.

3.2.4 Separation Optimizer

Interactions between the crystals on either side of the interface change under rotations and translations, introducing a new variable in locating interface energy minima. The interface separation optimizer tool uses the Broyden-Fletcher-Goldfarb-Shanno¹⁰ algorithm quasi-Newton method to locate the separation that minimizes the interface energy without the need to calculate first or second derivatives. When the optimizer finishes, the resultant separation distance is stored and used for generating any subsequent interface structures.

3.2.5 Molecule Insertion

Crystalline tri-layers and molecular bridges can be constructed using Transire to insert the contents of a coordinate file at the interface. The object to be inserted is aligned with the center of the interface structure and is not resized or reshaped to fit the dimensions of the existing interface. Each time an interface structure is generated, Transire begins with the modified unit cells and constructs the full interface structure. As a result, any inserted objects are removed when the next interface structure is generated; however, the interface structure with the inserted crystal layer or molecule can be used for ET property calculations.

3.3 Electron Transport Properties

The ET property calculator included in the Atomic Simulation Environment (ASE)¹¹ requires only the localized Hamiltonian and overlap matrices divided into submatrices representing the central interface region and the bulk regions on the extreme sides of the interface. The needed matrices are printed during the QM energy calculation and read back in by Transire. The matrices are separated into the submatrices based on the user-defined bulk regions and passed to ASE. The transmission across the interface surface is calculated over a range of energies relative to the Fermi level with the range and stepsize defined by the user. In addition, the total conductance for the range of energy levels is calculated and outputted in units of siemens.

3.4 Obtaining and Installation

Using Transire requires installing the following software. All commands are given in bold.

- 1) Python 2.7
 - a) Check to see whether Python is installed with the command **python**
 - b) If not, download from <https://www.python.org/downloads/>

- c) Run the installer
- 2) Scipy/Numpy
 - a) Check to see whether numpy and scipy are installed using **import numpy** and **import scipy** inside the python interpreter
 - b) If not, follow download and installation instructions at <https://www.scipy.org/scipylib/download.html>
- 3) ASE 3.12
 - a) Download the code with **git clone -b 3.12.0 https://gitlab.com/ase/ase.git**
 - b) Within the ASE folder run **python setup.py install --user**
 - c) Ensure that ~/.local/bin is added to the PATH environmental variable
- 4) Mpmath 0.19
 - a) Download the code with **git clone https://github.com/fredrik-johansson/mpmath**
 - b) Within the mpmath folder run **python setup.py install --user**
- 5) SymPy 1.0
 - a) Download the code with **git clone git://github.com/sympy/sympy.git**
 - b) Within the Sympy folder run **python setup.py install --user**
- 6) CP2K (optional)
 - a) Follow the instructions for downloading and compiling CP2K found at <https://www.cp2k.org/howto:compile>
 - b) Ensure that the location of the executable is included in the PATH variable
- 7) Pycp2k (optional)
 - a) Download the code **with git clone git://github.com/SINGROUP/pycp2k.git**
 - b) Within pycp2k folder run **python setup.py install --user** and follow instructions
- 8) LAMMPS (optional)

- a) Follow the instructions for downloading and installing LAMMPS found at http://lammps.sandia.gov/doc/Section_start.html
 - Follow instructions for serial or parallel LAMMPS
 - Follow instructions for compiling a shared library (mode=shlib)
 - Ensure location of liblammps.so is included in LD_LIBRARY_PATH variable
- 9) lammpslib (optional)
 - a) The download link for lammpslib.py can be found from the ASE website: <https://wiki.fysik.dtu.dk/ase/ase/calculators/lammps.html#module-ase.calculators.lammps>
 - b) Ensure the location of lammpslib.py is included in the PYTHON_PATH variable

3.5 Usage

The input requirements for Transire consist of a text file with keyword options, 1 or 2 unit cell coordinate files, and optionally a text file with method-specific options for energy calculations. The input file consists of keyword/value lines of the format:

```
keyword = value
```

In cases where the keyword takes an array of values such as a set of Miller indices, the values of the array are entered as individual values separated by spaces.

```
array_keyword = value_1 value_2 value_3
```

Keywords that take logical values, the value is either “True” or “False” without the quotation marks.

```
logical_keyword = True or False
```

Lines in the input file that begin with # or do not contain one of the keywords are ignored. The input file is specified with the command line option -i when Transire.py is executed.

The 1 or 2 unit cell coordinates for grain boundaries or heterojunctions, respectively, require only the coordinates and the lattice vectors. ASE supports a large number of input formats that meet this requirement.¹² The crystallographic

information file and extended-xyz formats work particularly well. The coordinate file or files are specified in the input file.

Additional user input required for energy calculations using CP2K¹³ or LAMMPS¹⁴ is given in a separate file. CP2K options are specified using the pycp2k format.

```
calc.CP2K_INPUT.FORCE_EVAL.DFT.SCF.QS.Method = "GPW"
```

To generate the output necessary for electron transfer properties the following lines must be included in the CP2K input file:

```
DFT = calc.CP2K_INPUT.FORCE_EVAL.DFT
DFT.PRINT.AO_MATRICES.Core_hamiltonian = ".True."
DFT.PRINT.AO_MATRICES.Overlap = ".True."
DFT.PRINT.AO_MATRICES.FileName = "hamiltonian"
```

The LAMMPS file consists of a single list declaration in Python format with each line of the LAMMPS input as an element in the list. Most of the LAMMPS parameters are automatically generated, but the `pair_style` and `pair_coeff` keywords that govern the inter-atomic potentials must be specified for all combinations of elements present in the grain boundary or heterointerface.

```
Commands = [
    "neighbor      3 bin",
    "neigh_modify  every 1",
    "pair_style    bop",
    "pair_coeff * * AlCu.bop.table Al Cu"
]
```

3.6 Output

When Transire runs, information is printed to the screen or standard output comprised mostly of error and warning messages. Almost all error messages reflect an issue with generating a specific interface structure and are not a cause for concern. Messages that begin with “Terminating Error” do represent an issue that must be fixed for Transire to perform properly. General Transire output is printed to a file specified with the input keyword “`output_file`” (default is `transire.out`), while tool and search specific output options can be turned on in the input file.

3.7 Limitations

The primary limitations for Transire are the capabilities of the computer systems used. The process replicating the unit cells to ensure periodicity is maintained and the 2 surfaces match can result in a massive number of atoms in the interface

structure. Both the memory requirements to store the coordinates of all the atoms in active memory, and the practical computational costs of calculating the energy of interface structures on the order of 5000 or more atoms using CP2K, limit the capabilities of this program beyond the limitations of the software. Electron transport calculations in particular are limited by the size of the simulated system as the memory requirement scales as N^2 where N is the number of basis functions in the scatter or central region. Future work will address the memory issue for ET calculations by exploiting the sparse nature of the Hamiltonian matrix to reduce memory and computational costs.

4. Results and Discussion

4.1 Graphene

The stacking of graphene sheets provides both an example of using Transire and an opportunity to compare the interface structures to those generated by studies using similar methods.^{5,15} The input file necessary to locate viable interface structures for twist rotations in the range of 0° to 60° is presented here:

```
crys_a_file = graphene.xyz
crys_a_surface = 0 0 1
crys_a_layers = 1
crys_b_surface = 0 0 1
crys_b_layers = 1
crys_b_file = graphene.xyz
separation = 0.3
tolerance = 1.0e-1
calculate_initial_energy = False
project_name = graphene
max_atoms = 100000
output_file = transire.out
search_list = 1
angles_stepsize = 0.01
number_of_angles = 6000
angle_calculate_energy = False
angle_write_coord_file = False
angle_write_energy_file = False
```

The first 6 lines define the coordinate file to read in the graphene unit cell, the Miller Indices for orienting the 2 unit cells, and the number of layers for each crystal. As the graphene sheets are parallel and monolayers, the indices correspond to the same orientation in the coordinate file and no replication along the axis perpendicular to the interface. The separation is a vacuum of specified width placed

between the 2 super cells when the interface structure is generated. The tolerance defines the error cutoff when determining the orthorhombic super cell dimensions and the dimensions of the interface structure. The `project_name` and `output_file` are used for naming the files that various pieces of information are printed to. The `max_atoms` sets the upper limit on the number of atoms used when generating the interface structure as an early check to avoid wasting computer resources by filtering out exceedingly large structures.

The `search_list`, `angles_stepsize`, and `number_of_angles` instruct Transire to perform a twist angle search (`search_list = 1`) over the range of 0° (default) to 60° (`number_of_angles = 6000`) in 0.01° increments (`angles_stepsize = 0.01`). The `calculate_initial_energy`, `angle_calculate_energy`, `write_coord_file`, and `write_angle_energy_file` are included to stop the default behavior of calculating the interface energy at each step and printing out results not used in the present calculation.

The results can be extracted from the `output_file` using `grep` from the command line with the following form:

```
grep angle output_file
```

A sample of the resultant output is reproduced here:

```
printing output: angle = 36.81 atoms = 1474
printing output: angle = 36.82 atoms = 1608
printing output: angle = 36.83 atoms = 1474
printing output: angle = 36.84 atoms = 9648
printing output: angle = 36.85 atoms = 1474
printing output: angle = 36.86 atoms = 1608
printing output: angle = 36.88 atoms = 26368
printing output: angle = 36.89 atoms = 13504
printing output: angle = 36.9 atoms = 8680
printing output: angle = 36.91 atoms = 7072
```

Plotting the number of atoms in the interface structures versus the twist angle shows how wide the variation in interface sizes for small changes in twist angle (Fig. 2). Comparing to previously reported work,^{5,15} Transire reproduces approximately the symmetry around the 30° twist rotation, decreasing to minima on around $\pm 8^\circ$ and then increasing to asymptotically near 0° and 60° . Significantly, Transire differs with a minimum within 0.1° of 30° and a broadened lower bound for the rotations greater than 30° . Both differences are due to the difference between the analytical solution employed for generating moiré patterns in previous works and the numerical approach employed here. Reducing the error tolerance in Transire will result in fewer overall interface structures generated but a higher percentage of

interface structures that correspond to the analytically determined structures as the tighter tolerance acts to filter out noise in the numerical solution.

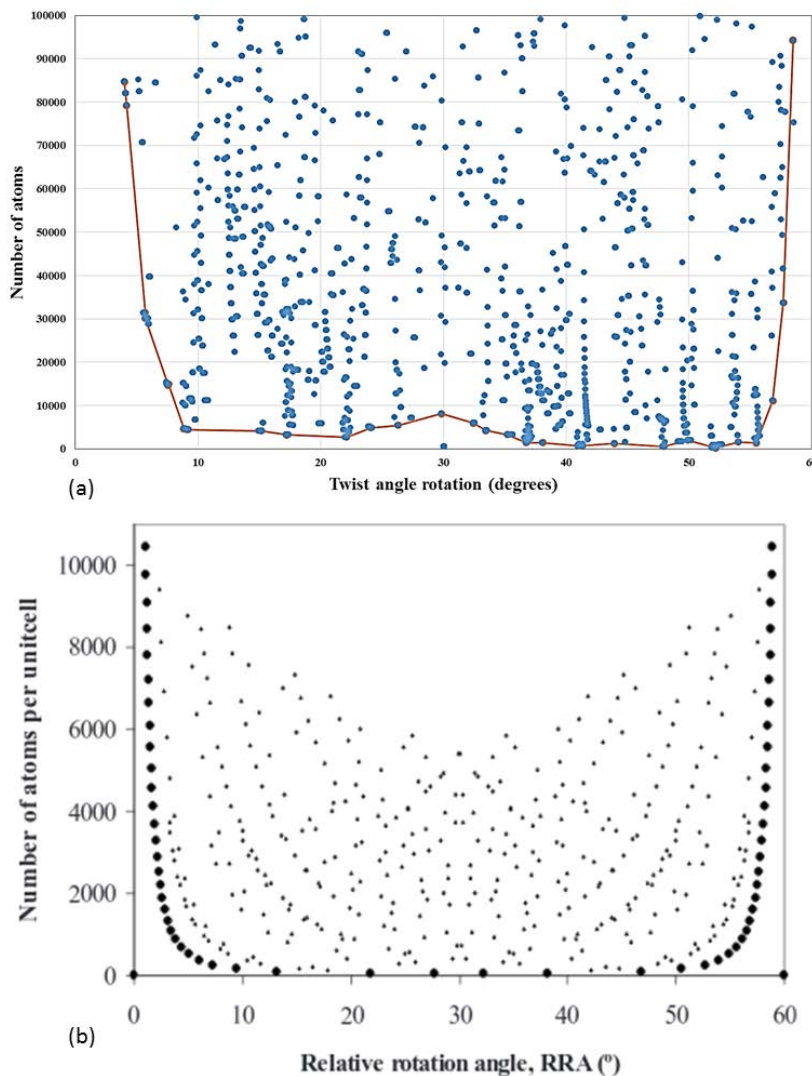


Fig. 2 The number of atoms in parallel graphene sheets interface structures generated by twist rotation using a) Transire and b) an analytical solution for Moiré structures reproduced from Campanera et al.¹⁵

4.2 Copper Grain Boundaries

The RAS method is demonstrated using copper to construct both sides of the interface. The choice of pure copper crystals enables the use of LAMMPS for energy calculations and a setting for the max number of atoms in the hundred thousands. In addition to changing both `crys_a_file` and `crys_b_file` to point to the copper crystal unit cell coordinate file, the following lines will be added/changed to the input file.

```

energy_method = lammmps
remove_duplicates = True
surface_search = True
range_surface_h_a = 0 1
range_surface_k_a = 0 1
range_surface_l_a = 0 1
range_surface_h_b = 0 1
range_surface_k_b = 0 1
range_surface_l_b = 0 1
angles_stepsize = 0.1
number_of_angles = 1800
angle_write_energy_file = True
angle_write_coord_file = False
angle_calculate_energy = True
angle_restart_initial = True
lammmps_input = lammmps_cucu

```

The option `remove_duplicates` activates an optional check for atoms that are overlapping in the interface structure without accounting for periodicity whenever an interface structure is generated within Transire. The `surface_search` defines the range of surfaces given as Miller indices, defining the interface normal for each crystal that all subsequent searches will be carried out on. The list of Miller Indices are formed by taking all combinations of the `range_surface` inputs excluding only the possibility of all “a” or “b” indices being 0. In this example, Transire will generate interface structures for all 49 surface pairs with twist rotations from 0° to 180° in 0.1° steps. Assuming a conservative 20% failure rate for constructing the interface structures, this results in 70,560 energy calculations. To reduce the number of necessary energy calculations, the RAS can be used by making the following changes to the input.

```

angles_stepsize = 1
number_of_angles = 180
ras_depth = 2
ras_energy = True
ras_all_angles = True

```

The `ras_depth` and `angle-stepsize` means that the initial search will use a stepsize of 1° and the second search will use a stepsize of 0.1°. The number of angles to look for using RAS can be set using `ras_factor` (not used here) or using the number of rotations that result in successfully constructed interface structures found during the first search by setting `ras_all_angles = True`. The `ras_energy` option instructs Transire to calculate the interface energy of the structures constructed using the final list of angles located using RAS.

The results for selected pairs of Miller indices are presented in Fig. 3. For the details of the energy surface near $0 \text{ eV}/\text{\AA}^2$ to be visible, the structures with energy greater than $50 \text{ eV}/\text{\AA}^2$ are plotted but not visible on the graphs. The figures show 3 sets of data corresponding to the direct angle search with 0.1° stepsize (dashed line), the RAS search (triangle points), and a line following the lower bounds of the RAS search. As expected, the RAS search does not reproduce the full angle search results; however, the RAS results do capture the approximate shape of the lower bound of the full angle search as evidenced by the line marking the lower bound of the RAS results. The full angle search operating in serial had an average time per pair of Miller indices of 16 h while the RAS search on the same equipment had an average time of 3 h per surface. The RAS method in Transire can be used to generate the approximate energy plot of an interface structure versus twist angle, requiring much less computational time than a full twist angle search of the same interface structure.

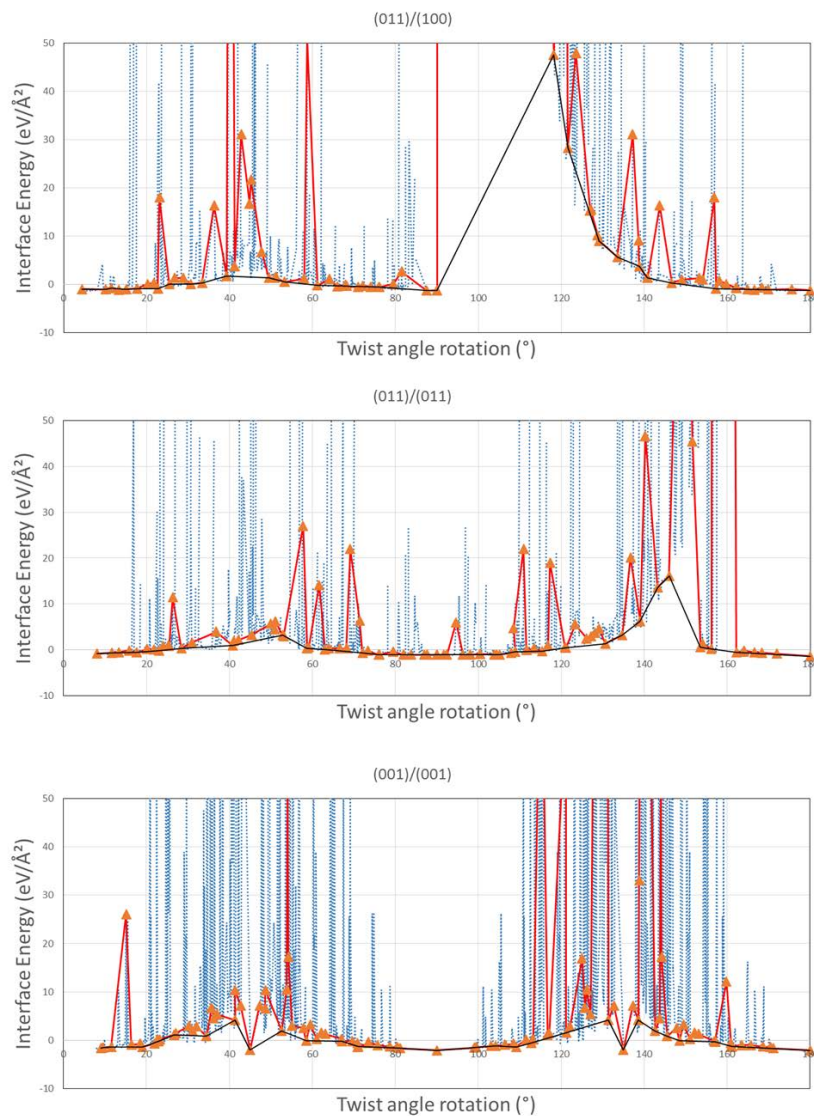


Fig. 3 Interface energies for the copper-copper grain boundaries with Miller indices pairs 011/100, 011/011, and 001/001. Energies given for interface structures for twist rotations at every 0.1° (dashed line) and the RAS method (triangles). The solid line marks the lower envelope of the RAS results showing a close match with full twist angle search results.

4.3 GaN/SiC Heterojunction

For the much more costly QM calculations using CP2K, the RAS method serves the alternate purpose of locating the interface structures for which interface energies can be calculated with the constraints of the hardware being used. To demonstrate the application of Transire to a heterojunction using QM calculations, the RAS method is used to locate minima in interface energy for gallium nitride (GaN) and silicon carbide (SiC) for all symmetry unique surfaces for indices in the range 0–1

and twist angles ranging from 0° to 180°. The Transire input file is the same as for the copper/copper RAS search with the following changes/additions:

```
crys_a_file = GaN.cif
crys_b_file = SiC.cif
energy_method = cp2k
max_atoms = 5500
cp2k_input = cp2k_gan
max_mpi_processes = 128
atoms_per_process = 13
working_directory = ./
```

The keywords `crys_a_file`, `crys_b_file`, `cp2k_input`, and `working_directory` should all be changed to the filenames and directory structure of the user. The keyword `max_atoms` can be determined by experimenting with cp2k calculations of increasing size. `max_mpi_processes` is used when running in parallel and should be equal to the number of processes designated by batch queueing system. The `atoms_per_process` is used to scale the number of processors used for the cp2k calculation to reduce inefficiency caused by using too many processes for small chemical systems. The value for `atoms_per_process` can be optimized by testing similar to `max_atoms`, but 13 has been found to be a reasonable value. See the Appendix for a sample pycp2k input file.

Of the 21 symmetry unique pairs of Miller indices for the GaN/SiC interface, 12 resulted in 3 or more successfully generated interface structures from twist rotations. The largest number of interface structures for a single surface is 30 for the GaN 001/SiC 001 surfaces while the average of the 12 surface pairs is 9.75 structures per surface pair. This is too few data points to predict the interface energy surface using the lower envelope method described above with certainty. Transire does locate 6 unoptimized minima presented in Table 1 that can serve as the basis for further study. The values in Table 1 represent the lowest energy interface structure for each pair of Miller indices and not the overall 6 lowest energy interface structures.

Table 1 Lowest energy structures for GaN/SiC interfaces

GaN surface	SiC surface	Twist angle (°)	Interface energy (eV/Å ²)
001	001	23.7	-102.700
001	011	120	-73.038
001	111	30	-72.174
111	001	96.8	-120.725
111	011	100.3	-59.036
111	111	18.9	-40.591

Electron transport calculations in Transire occur either as part of a Transire run or starting from the restart files printed from a previous run. To run an inline ET calculation, add the following keywords to the input file:

```
perform_ET = True
number_of_layers_a = 2
number_of_layers_b = 2
exclude_coupling = True
energy_levels_ET = -3 5 0.1
```

The number of layers of unit cells, as defined when the interface structure is generated and after the surface cut has been applied to the user-generated unit cell, to assign to each bulk side region must be multiples of 2 but also small enough that the central region is larger than either side region. The `exclude_coupling` keyword disables reading in the coupling between the side and central regions from the Hamiltonian, instead assuming the coupling is the same as between the layers in the side regions. The `energy_levels_ET` takes 3 values representing the lower bound, upper bound, and stepsize for the energy levels at which the transmission will be calculated relative to the Fermi level. To start instead from a previous restart file, include the following lines:

```
ET_restart = True
restart_path = {path to folder}
restart_file = {file name without extension}
calculate_initial_energy = False
```

The `restart_path` and `restart_file` keywords specify the absolute path to the folder containing the results of a previous run and the filename associated with all output for a single interface structure. Setting `calculate_initial_energy` to `False` means that the energy calculation does not need to be repeated prior to the ET calculation.

The output of the ET calculation includes a list of the transmission values at each energy level and the conductance in siemens as calculated by the area under the transmission spectrum. A sample of the output for the GaN 001/ SiC 001 interface with a 59.9° twist rotation is reproduced here:

```
====Conductance=====
In range from -3.0 eV to 5.0 eV relative to Fermi
level
Conductance = 0.000120389023556 siemens
=====
Energy = -3.000 Transmission = 1.04189268874e-10
Energy = -2.900 Transmission = 8.28204484373e-11
Energy = -2.800 Transmission = 5.75690142338e-10
```

[...]
Energy = 0.000 Transmission = 1.52076993493e-12
Energy = 0.100 Transmission = 4.20679561841e-13
Energy = 0.200 Transmission = 8.66903777708e-12
Energy = 0.300 Transmission = 7.61513015853e-05
Energy = 0.400 Transmission = 0.582636114854
Energy = 0.500 Transmission = 1.97684732224
Energy = 0.600 Transmission = 1.45024215504
Energy = 0.700 Transmission = 1.79466938788
Energy = 0.800 Transmission = 2.73848686458
Energy = 0.900 Transmission = 2.0731772229
Energy = 1.000 Transmission = 0.700977426521
Energy = 1.100 Transmission = 0.00580269561026
Energy = 1.200 Transmission = 0.002262243844
Energy = 1.300 Transmission = 0.058924088001
Energy = 1.400 Transmission = 0.00593575919353
Energy = 1.500 Transmission = 0.00693362102418
Energy = 1.600 Transmission = 1.07180495472
Energy = 1.700 Transmission = 0.271576591225
Energy = 1.800 Transmission = 0.973361114618
Energy = 1.900 Transmission = 0.765007660261
Energy = 2.000 Transmission = 1.0148580544
Energy = 2.100 Transmission = 0.0518980373159
Energy = 2.200 Transmission = 0.000838132090911
Energy = 2.300 Transmission = -0.000300984518523
Energy = 2.400 Transmission = -0.0022509128318
Energy = 2.500 Transmission = -0.000697484340994
Energy = 2.600 Transmission = -0.000727872114575
Energy = 2.700 Transmission = -7.38330191189e-05
Energy = 2.800 Transmission = -0.00427433096832
Energy = 2.900 Transmission = -3.52556817825e-05
Energy = 3.000 Transmission = -3.04734924671e-06

Figure 4 shows the result of plotting the transmission versus energy with clearly defined transmission bands with a small gap centered around 1.3 eV.

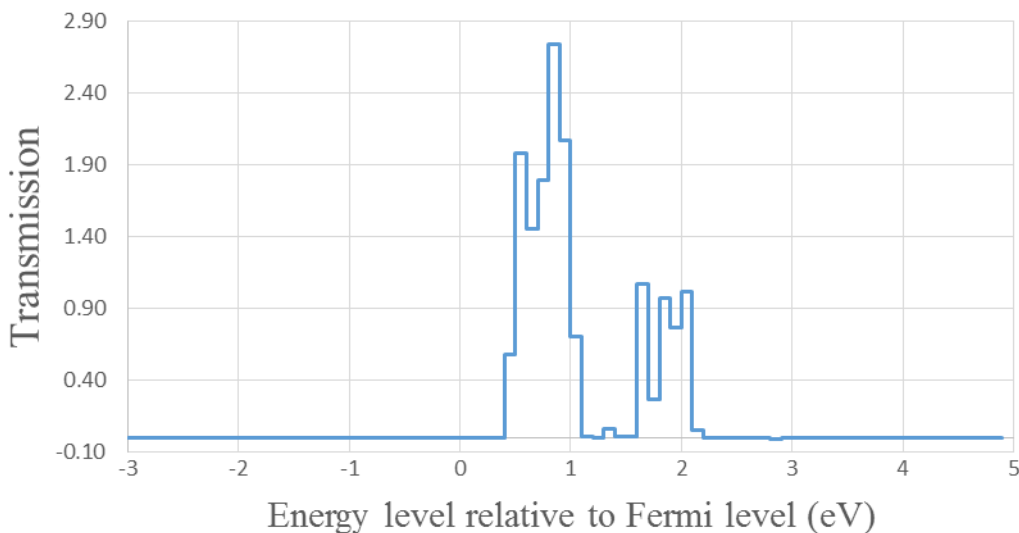


Fig. 4 Transmission spectra across GaN 001/SiC 001 interface with 59.9° twist rotation

5. Conclusion

Computational studies of crystalline chemical systems require a complete description of the atoms in a form that can be used by computational chemistry software. Transire was developed to generate the necessary descriptions for planar interfaces between crystalline solids. Starting from readily available unit cell coordinates, Transire can generate a wide range of interface structures incorporating cut planes, twist rotations, and translations. Integrated support is included for MM and QM calculations using LAMMPS and CP2K, respectively. The ET properties can be calculated using the interface structures generated by Transire using the output from the QM calculations. Transire provides an end-to-end software for generating and analyzing crystalline interface structures with a minimal user input requirement.

Three test cases have been presented to demonstrate the capabilities of Transire: graphene bilayer, copper grain boundaries, and the GaN/SiC heterojunction. Constructing the graphene bilayer structures by twist rotation reproduces qualitative features or previously published analytical solutions^{5,15} with differences caused by noise in the numerical solution. Employing the RAS method for determining the interface energy plot for several pairs of Miller indices over a range of twist angles shows a 5-fold decrease in the computational cost over an exhaustive search of angles in the same range using a set stepsize while producing a lower envelope of the energy plot that closely approximates the lower envelope of the exhaustive search. The RAS method was used to locate low energy interface structures for the GaN/SiC heterojunction that are small enough for QM

calculations. In addition, the transmission spectrum for ET across the interface has been calculated for one of the interface structures generated by Transire.

Transire simplifies the process of generating and evaluating chemical systems with a planar interface by combining simple input requirements and powerful tools. Transire is designed to continue developing as new tools are added and the ET-related capabilities are expanded.

6. References

1. Tschopp MA, McDowell DL. Structures and energies of Σ 3 asymmetric tilt grain boundaries in copper and aluminum. *Philosophical Magazine*. 2007;87:3147–3173.
2. Tschopp MA, McDowell DL. Atomic scale investigation of grain boundary structure role on deformation and crack growth dynamics in aluminum. *Philosophical Magazine*. 2007;87:3871–3892.
3. Wimmer E, Christensen M, Eyert V, Wolf W, Reith D, Rozanska X, Freeman C, Saxe P. Computational materials engineering: recent applications of VASP in the MedeA[®] software environment. *J Korean Ceram Soc*. 2016;53(3):263–272.
4. MedeA. Ver. 2.20 Angel Fire (NM): Materials Design, Inc. 2016. [accessed 2017 Aug 30]. <http://www.materialsdesign.com/medea>.
5. Shallcross S, Sharma S, Kandelaki E, Pankratov OA. Electronic structure of turbostratic graphene. *Phys Rev B*. 2010;81(16):165105.
6. Taylor J, Guo H, Wang J. *Ab initio* modeling of open systems: Charge transfer, electron conduction, and molecular switching of a C₆₀ device. *Phys Rev B*. 2001;63(12):121104.
7. Taylor J, Guo H, Wang J. *Ab initio* modeling of quantum transport properties of molecular electronic devices. *Phys Rev B*. 2001;63(24):245407.
8. Liu Q, Li L, Li Y, Gao Z, Chen Z, Lu J. Tuning electronic structure of bilayer MoS₂ by vertical electric field: a first-principles investigation. *J Phys Chem C*. 2012;116:21556–21562.
9. An Y, Zhang M, Wu D, Fu Z, Wang K. The electronic transport properties of transition-metal dichalcogenide lateral heterojunctions. *J Mater Chem C*. 2016;4:10962-10966.
10. Nocedal J, Wright S. Numerical optimization. New York (NY): Springer-Verlag; 1999. p. 198.
11. Larsen AH, et al. The atomic simulation environment—a Python library for working with atoms. *J Phys Condens Matter*. 2017;29:273002.
12. File input and output. Ver. 3.14.1. ASE-developers. 2017 [accessed 2017 Aug 31]. <https://wiki.fysik.dtu.dk/ase/ase/io/io.html>.

13. Hutter J, Iannuzzi M, Schiffmann F, VandeVondele J. CP2K: Atomistic simulations of condensed matter systems. *Wiley Interdiscip Rev: Comput Mol Sci.* 2014;4:15–25.
14. Plimpton S. Fast parallel algorithms for short-range molecular dynamics. *J Comp Phys.* 1995;117:1–19.
15. Campanera JM, Savini G, Suarez-Martinez I, Heggie MI. Density functional calculations on the intricacies of Moiré patterns on graphite. *Phys Rev B.* 2007;75(23):235449.

Appendix. Sample pycp2k Input File

A sample pycp2k input file for CP2K calculations involving Ga, N, Si, and C is presented here. The simplest usage is to edit a preexisting pycp2k file or to start with the initial declarations and add in CP2K input keywords.

```
#===== initial declarations
CP2K_INPUT = calc.CP2K_INPUT
GLOBAL = CP2K_INPUT.GLOBAL
FORCE_EVAL = CP2K_INPUT.FORCE_EVAL_add()
SUBSYS = FORCE_EVAL.SUBSYS
DFT = FORCE_EVAL.DFT
SCF = DFT.SCF
MOTION = CP2K_INPUT.MOTION
CONSTRAINT = MOTION.CONSTRAINT
#===== inputs

GLOBAL.Run_type = "ENERGY"
GLOBAL.Print_level = "MEDIUM"
GLOBAL.PRINT.Add_last = "NUMERIC"
GLOBAL.Extended_fft_lengths = ".True."
GLOBAL.Walltime = "08:00:00"
calc.create_cell(SUBSYS, self.config.atom)
calc.create_coord(SUBSYS, self.config.atom)

FORCE_EVAL.Method = "Quickstep"
FORCE_EVAL.PRINT.FORCES.Section_parameters = "OFF"
FORCE_EVAL.PRINT.FORCES.Filename = "forces"
FORCE_EVAL.PRINT.FORCES.Section_parameters = "OFF"

#===== MNDO

DFT.QS.Method = "GPW"
DFT.QS.Eps_default = 1.0E-9
DFT.SCF.Scf_guess = "ATOMIC"
DFT.SCF.OT.Section_parameters = ".TRUE."
DFT.SCF.OUTER_SCF.Section_parameters = ".TRUE."
DFT.SCF.OT.Linesearch = "2PNT"
DFT.SCF.OT.Minimizer = "DIIS"
DFT.LS_SCF.Dynamic_threshold = ".TRUE."
DFT.LS_SCF.Ls_diis = ".TRUE."
DFT.SCF.OUTER_SCF.Max_scf = "200"
DFT.POISSON.EWALD.Ewald_type = "SPME"
DFT.POISSON.EWALD.Gmax = 30
DFT.Basis_set_file_name = "BASIS_MOLOPT"
DFT.Potential_file_name = "GTH_POTENTIALS"
DFT.MGRID.Ngrids = 4
DFT.MGRID.Cutoff = 60
DFT.MGRID.Rel_cutoff = 30
DFT.XC.XC_FUNCTIONAL.Section_parameters = "PADE"
DFT.XC.Density_smooth_cutoff_range = 1.0E-5
DFT.XC.XC_GRID.Xc_smooth_rho = "NN10"
```

Approved for public release; distribution is unlimited.

```

DFT.XC.XC_GRID.Xc_deriv = "SPLINE2"
DFT.XC.XC_GRID.Use_finer_grid = ".True."

#===== Necessary lines for any ET
calculations
DFT.PRINT.AO_MATRICES.Core_hamiltonian = ".True."
DFT.PRINT.AO_MATRICES.Overlap = ".True."
DFT.PRINT.AO_MATRICES.Filename = "hamiltonian"

#===== Elements
KIND = SUBSYS.KIND_add("Si") # Section_parameters can be
provided as argument.
KIND.Basis_set = "SZV-MOLOPT-SR-GTH"
KIND.Potential = "GTH-PADE-q4"

KIND = SUBSYS.KIND_add("C")
KIND.Basis_set = "SZV-MOLOPT-GTH"
KIND.Potential = "GTH-PADE-q4"

KIND = SUBSYS.KIND_add("Ga")
#KIND.Basis_set = "def2-SVP"
KIND.Basis_set = "SZV-MOLOPT-SR-GTH"
KIND.Potential = "GTH-PADE-q13"

KIND = SUBSYS.KIND_add("N")
KIND.Basis_set = "SZV-MOLOPT-GTH"
KIND.Potential = "GTH-PADE-q5"

```

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
5DOF	5 degrees of freedom
ARL	US Army Research Laboratory
ASE	Atomic Simulation Environment
ET	electron transport
GaN	gallium nitride
MM	molecular mechanics
NEGF	non-equilibrium Green's function
QM	quantum mechanics
RAS	reducing angle search
SiC	silicon carbide

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIR ARL
(PDF) RDRL CIO L
IMAL HRA MAIL & RECORDS
MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 ARL
(PDF) RDRL WMM G
C CARLIN

INTENTIONALLY LEFT BLANK.