V1.0.0: Guide to Setting Up and Running the sp3d5s* Tight-Binding Solver on Gadi

Sarinle

21/3/2025

1 Obtain the Code & Input Files

The following are the requirements to run the tight-binding solver.

1.1 Code Files

- You should have the tight-binding executable, named `sdt`.

1.2 User Input File

- You need an XML file which configures your specific simulation parameters. To construct one, a sample is shown below.

1.3 Material Database File

- You will need a material database XML file (e.g. `Jancu_Materials.xml`). It must have the same format as shown in the example and can be hard coded from references.

---

2 Environment Setup on Gadi

Compilation of the code is beyond the scope of this document. However, assuming you are on Gadi, if you runs compiled the libraries that are not available on Gadi.

2.1 Set Environment Variables

Redirect to where you wish to run your code and set the following variables. These allow your exectuable to identify relevant libraries.

```
export SLEPC_DIR=/g/data/ad73/sy1090/AngusTB/libraries/slepc-install
export PETSC_DIR=/g/data/ad73/sy1090/AngusTB/libraries/petsc-3.18.4
export PETSC_ARCH=arch-linux-c-debug
export TRILINOS_DIR=/g/data/ad73/sy1090/AngusTB/libraries/trilinos-install
```

2.2 Load Required Modules

Then load the following modules:

```
module load pbs openmpi/4.0.2 intel-compiler/2020.0.166 intel-mkl/2020.0.166 cmake gsl e
```

This ensures the compiler, MPI, MKL, GSL, and Eigen libraries are properly set up.

---

3 Running the Tight-Binding Solver

Once your environment is set, you can run the solver. An example command is:

```
./bin/sdt tight-binding \
  --user_input="/g/data/ad73/sy1090/AngusTB/bin/user_input_tight_binding.
     xml" \
  --save_dir="/g/data/ad73/sy1090/AngusTB"
```

- `--user_input` specifies the path to your user input XML file.
- `--save_dir` specifies the output directory where results (like `bandstructure.csv`) will be written.

Tip: If you run into file or directory errors, double-check that the user input file path and save directory path exist or can be written to.

---

4 User input file

Manipulating the user-input file is advisable, below is the general example template of the input file. The example file can also be provided directly by Sarinle.

```
<user_inputs>
  <!-- Domain and cell specifications -->
  <domain>
    <dom_x>1</dom_x>
    <dom_y>1</dom_y>
    <dom_z>1</dom_z>
  </domain>
  <cell_name>zincblende</cell_name>

  <!-- Field parameters -->
  <fields>
    <mag_field>0.0, 0.0, 0.0</mag_field>
    <elec_field>0.0, 0.0, 0.0</elec_field>
  </fields>

  <!-- Material and k-space settings -->
  <material>Silicon</material>
  <kvec></kvec>
  <material_db_path>/g/data/ad73/sy1090/AngusTB/material_database/
     Jancu_Materials.xml</material_db_path>


  <!-- Various flags -->
  <flags>
    <print_cells>0</print_cells>
    <print_materials>0</print_materials>
    <dump_ham>0</dump_ham>
    <dump_prob>0</dump_prob>
    <dump_pdb>0</dump_pdb>
    <dump_csv>1</dump_csv>
  </flags>

  <!-- Plotting and passivation settings -->
  <plot_steps>10000</plot_steps>
  <passivation>
    <pass_x>0</pass_x>
    <pass_y>0</pass_y>
    <pass_z>0</pass_z>
  </passivation>

  <!-- High-symmetry point definitions -->
  <high_symmetry>
    <!-- Fractional coordinates for high-symmetry points -->
    <L_frac>0.5, 0.5, 0.5</L_frac>
    <G_frac>0.0, 0.0, 0.0</G_frac>
    <X_frac>1.0, 0.0, 0.0</X_frac>
    <W_frac>1.0, 0.5, 0.0</W_frac>
    <K_frac>0.75, 0.75, 0.0</K_frac>
    <U_frac>1.0, 0.25, 0.25</U_frac>
```

```
    <!-- Base number of interpolation steps for the k-path -->
    <base_steps>10000</base_steps>

    <!-- Define the desired k-path with corresponding high-symmetry labels
        -->
    <path>
      <point label="L">L</point>
      <point label="  ">  </point>
      <point label="X">X</point>
      <point label="W">W</point>
      <point label="K">K</point>
      <point label="L">L</point>
      <point label="W">W</point>
      <point label="X">X</point>
      <point label="K">K</point>
      <point label="  ">  </point>
    </path>
  </high_symmetry>
</user_inputs>
```

4.0.1 Overview of parameters in user-input file

1. `<domain>`
   Purpose: Specifies the domain replication parameters.
   Details:

   - `<dom_x>`, `<dom_y>`, `<dom_z>`: Define how many times the primitive cell is repeated along the x, y, and z directions, respectively.
   - For a single unit simulation, all values should be set to 1.

2. `<cell_name>`
   Purpose: Specifies the crystal structure to use.
   Details:

   - Example: `zincblende` (also covers diamond structures in the code).
   - Currently, only `zincblende` is fully supported.

3. `<fields>`
   Purpose: Defines external fields applied in the simulation.
   Details:

   - `<mag_field>`: Sets a magnetic field (B_x, B_y, B_z).
   - `<elec_field>`: Sets an electric field (E_x, E_y, E_z).
   - The current version is only tested with both fields set to `0.0, 0.0, 0.0`.

4. `<material>`
   Purpose: Specifies the material to simulate.
   Details: The provided string must match the `name` attribute of a corresponding entry in your material database.

5. `<kvec>`
   Purpose: Optionally sets a single wavevector (k_x, k_y, k_z) at which to solve the band structure.
   Details:

   - If empty: The code computes the full k-path as defined in the `<high_symmetry>` section.
   - If provided: Eigenvalues are calculated only at the specified $\mathbf{k}$.

6. `<material_db_path>`
   Purpose: Provides a file path (string) to the XML database containing material parameters (e.g., Slater–Koster data).
   Details:

   - Example: `Jancu_Materials.xml`.
   - See "Section 5" of the documentation for details on adding or modifying materials.

7. <flags>
   Purpose: Contains various toggle parameters for debugging or output options.
   Details:

   - <print_cells> and <print_materials>: Dump geometry or material info to the screen.
   - <dump_ham>: Writes out the Hamiltonian matrix (in Matrix Market format) for debugging.
   - <dump_prob>: Writes wavefunction probability data to wf_prob.txt.
   - <dump_pdb>: Exports the constructed geometry as a .pdb file.
   - <dump_csv>: Produces bandstructure.csv containing energies at each **k** along the path.

8. <plot_steps>
   Purpose: Currently unused.
   Details: Intended for future features such as controlling the number of k-points or plotting intervals in real space.

9. <passivation>
   Purpose: Controls boundary passivation, currently untested in version.
   Details:

   - <pass_x>, <pass_y>, <pass_z>: Each can be set to 0 or 1.
   - If set to 0: The code uses periodic boundaries in that direction.
   - If set to 1: The code attempts to passivate the boundary.

10. <high_symmetry>
    Purpose: Specifies high-symmetry **k**-points and defines the interpolation path for a band-structure sweep.
    Details:

    - Fractional Coordinates: <L_frac>, <G_frac>, <X_frac>, <W_frac>, <K_frac>, <U_frac> define the fractional coordinates in reciprocal space.
    - <base_steps>: Sets the number of interpolation points for the first segment (subsequent segments scale accordingly).
    - <path>: Contains a sequence of <point> elements, each with a label (e.g., L, Γ, X, etc.) that corresponds to one of the fractional coordinates. The code interpolates between these points to form the full band path.

---

5 The Material Database XML

The materials database stores all the parameters required for running the tight-binding simulation using the sp3d5s* format; see Jancu's [1] paper for more details. Note that for diamond structures, the cation-to-anion and anion-to-cation inputs are identical, resulting in duplicate entries as shown below.

```
<materials>
        <material name="Silicon">
                <!-- Lattice constant and reference -->
                <a val="5.430" />
                <E_100 val="5.1016" />
                <!-- Reference not given -->
                <!-- Onsite energies -->
                <E_s_a val="-2.0196" />
                <E_s_c val="-2.0196" />
                <E_p_a val="4.5448" />
                <E_p_c val="4.5448" />
                <E_d val="14.1836" />
                <E_sstar val="19.6748" />
                <!-- Two-center integrals -->
                <ss_sigma val="-1.9413" />
                <sstar_sstar_sigma val="-3.3081" />
                <sa_star_sc_sigma val="-1.6933" />
                <sa_sc_star_sigma val="-1.6933" />
                <sa_pc_sigma val="2.7836" />
                <sc_pa_sigma val="2.7836" />
                <sa_star_pc_sigma val="2.8428" />
                <sc_star_pa_sigma val="2.8428" />
                <sa_dc_sigma val="-2.7998" />
                <sc_da_sigma val="-2.7998" />
                <sa_star_dc_sigma val="-0.7003" />
                <sc_star_da_sigma val="-0.7003" />
                <pp_sigma val="4.1068" />
                <pp_pi val="-1.5934" />
                <pa_dc_sigma val="-2.1073" />
                <pc_da_sigma val="-2.1073" />
                <pa_dc_pi val="1.9977" />
                <pc_da_pi val="1.9977" />
                <dd_sigma val="-1.2327" />
                <dd_pi val="2.5145" />
                <dd_delta val="-2.4734" />
                <!-- Spin-orbit parameters -->
                <Delta_a_over_3 val="0.0195" />
                <Delta_c_over_3 val="0.0195" />
        </material>
</materials>
```

5.1 Adding a New Material

To simulate a new material (e.g., GaAs, AlAs, or DummyMaterial):

1. Add a new `<material>` block in the same format (with the same set of parameters such as `<E_s_a>`, `<E_s_c>`, `<E_p_a>`, `<E_p_c>`, etc.).
2. Adjust the numerical values to match your parameters, these can be found in papers such as [1].
3. Give it a unique `<material name="...">` attribute.
4. Reference that name in your user-input file (e.g `user_input_tight_binding.xml`) via `<material>YOUR_NEW_MATERIAL</material>`.

Every coefficient in these material entries represents Slater-Koster matrix elements or onsite energies for the sp3d5s* basis. The code internally constructs the Hamiltonian from these integrals.

---

6 Output: `bandstructure.csv`

When `dump_csv` is set to 1, the solver creates a file named `bandstructure.csv` in the directory specified by `--save_dir`.

Each row contains:

```
[label], [k_distance], [kx], [ky], [kz], [E0], [E1], [E2], ...
```

Where:

- `label` is the high-symmetry label for the first k-point of each segment (e.g., Γ, X, etc.). Otherwise, it will be "none" for intermediate points.
- `k_distance` is the cumulative distance along the band-structure path, useful for plotting.
- `kx, ky, kz` are the components of the wavevector in reciprocal space.
- `E0, E1, E2, ...` are the eigenvalues (energies in eV) at that k-point.

Plot the eigenvalues versus `k_distance` to visualize your band structure.

---

7 A Dummy Example

Below is a quick example of how one might run a very simple test with contrived parameters.

7.1 Creating a Dummy Material Entry

Generate your materials database file in your local directory, call it `dummy_materials.xml`.

```
<materials>
<material name="Dummy">
  <a val="5.000" />
  <E_100 val="5.0" />
  <E_s_a val="-2.0" />
  <E_s_c val="-2.0" />
  <E_p_a val="3.0" />
  <E_p_c val="3.0" />
  <E_d val="10.0" />
  <E_sstar val="15.0" />
  <ss_sigma val="-2.0" />
  <!-- fill in the rest with approximate or test values -->
  <Delta_a_over_3 val="0.01" />
  <Delta_c_over_3 val="0.01" />
</material>
</materials>
```

7.2 Modify the User Input File

Copy the provided `user_input_tight_binding.xml` file data into a created `dummy_input.xml` file within your directory. Only update the following parameters in the file.

```
<user_inputs>
  <!-- domain/cell, etc. same as before -->
  <material>Dummy</material>
  <material_db_path>/path/to/dummy_materials.xml</material_db_path>
  <flags>
    <dump_csv>1</dump_csv>
    <!-- etc. -->
  </flags>
</user_inputs>
```

Tip: If you want to know your directory, type `pwd` in Gadi!

7.3 Run the Solver

Execute:

```
.sdt tight-binding --user_input="/path/to/dummy_input.xml" \
   --save_dir="."
```

7.4 Inspect the Output

Check the generated `bandstructure.csv` for your dummy set of parameters.

---

8 Exercises

8.1 Exercise 1: Generating Silicon Band Structure with Jancu's Parameters

Use the given Jancu's tight-binding parameters for `Silicon` to generate its band structure. Compare your results with the experimental values reported in the Ioffe database. **Reference:** Silicon Band Structure – Ioffe Database

8.2 Exercise 2: Generating Band Structures for Germanium

Using Jancu's tight-binding parameters, generate the band structures for `Germanium`. Compare your results to the corresponding experimental data from the Ioffe database. **Reference:** Germanium Band Structure – Ioffe Database

8.3 Exercise 3: Generating Band Structures for GaAs

GaAs is a common III–V compound. Use Jancu's parameters to generate its band structure and compare the results to those reported in the Ioffe database. **Reference:** GaAs Band Structure – Ioffe Database

References

[1] J.-M. Jancu, R. Scholz, F. Beltram, and F. Bassani, *Phys. Rev. B* **57**, 6493 (1998).