PES2MP - Quick Guide

Quantum Dynamics Lab, Indian Institute of Technology Ropar

v. 2025.R2

Section 1

Introduction

1.1 About PES2MP program

PES2MP is a **potential energy surface (PES)** builder that can automate PES building and **augmentation using Neural Networks** (N-dimensional augmentation) or analytical fitting (restricted to radial augmentation only) while also automating **PES fitting** into **Radial Terms** using Legendre polynomials (2D) and Spherical Harmonics (4D) for studying rotational (de-)excitation of any rigid rotor.

The Python-based program automatically creates PES for various collisional pairs e.g. atom - atom (1D), rigid rotor - atom (2D), and rigid rotor - rigid rotor (4D). The program calculates and moves respective Rigid Rotor(s) (RR) to COM and generates PES using Psi4 (for rough estimation). The program also generates input files for external calculations using Molpro/Gaussian/Psi4 along with publication-ready plots with customization (title, axis, etc.) in pdf/eps formats. The users are expected to generate a PES using default coordinates which are sparse but can be changed to make coordinates denser. This sparse PES can then be augmented using an ensemble of Neural Networks (NN) models based on the high-level library of TensorFlow i.e. **Keras** where users can choose multiple models to avoid dependency on a singly trained machine learning (ML) model. The final part of the program does multipole expansion of the PES to obtain radial terms which can be analytically fitted into various functions for MOLSCAT calculations.

The first part of the program will take atoms and bond lengths (optimized or experimental) for each collider along with the charge and multiplicity of the fragments. The users can choose from various input files for generating Molpro or Gaussian input files (CP corrected or CBS) and leave the rest of the parameters to default.

For the NN model, a cutoff for very high energy is required, as the program creates separate models for minima and high-energy regions to preserve the features that are lost when both regions are trained on a single NN model. The program provides a residual plot for the NN-generated PES and also the PES generated after recreating the surface from radial coefficients (multipole (MP) expanded PES). Currently, the program uses Scipy and pyshtools to carry out MP expansion of 2D and 4D PES by calculating Legendre and spherical harmonics terms respectively.

The analytical fitting of these radial terms or 1D plot (optional) can be carried out using either $\sum_i \alpha_i R^{-\beta_i}$, $\sum_i \alpha_i e^{-\beta_i R}$ or mixed functions. The final output of these analytically fitted functions is provided in MOLSCAT readable format.

The subsequent sections cover the installation and usage process (of PES2MP) using Character User Interface (CUI) and Graphical User Interface (GUI). The final section discusses briefly on capabilities of PES2MP.

Section 2

Installation and Usage using CUI

This section covers installation instructions using the Character User Interface. For Graphical User Interface go to the next section.

2.1 Instructions for Installation

There are two options for makefiles :

1. Quick install :: **Filename**: *install_pes2mp_quick.sh* || **Conda environment**: *pes2mp_q*)
2. Conda install :: **Filename**: *install_pes2mp.sh* || **Conda environment**: *pes2mp*)

The quick version uses **python -m pip** command (faster but does not check the compatibility among packages) while the normal version uses **conda** command to install required packages. While the conda installation is recommended, it can sometimes be painfully slow to solve environment conflicts among packages and therefore it is suggested that users install both makefiles in separate terminals. If the conda installation doesn't finish in 24 hours (i.e. conda is stuck in a loop and keeps solving the environment), just cancel the installation and use the quick version. If both install normally, users can use any of the two environments to run the program.

2.1.1 Prerequisite

Anaconda
Install anaconda by visiting anaconda webpage i.e. `https://www.anaconda.com/download`.
It can create environments with specific Python versions and two environments are not affected by any change in the base (root) environment. For our work, we shall create **pes2mp** environment utilizing conda and **pes2mp_q** environment utilizing pip for installation of various packages (**makefile is provided**). The important packages are: Psi4, pyshtools, TensorFlow, SciPy, Matplotlib, tqdm, and many more. å

2.1.2 Installing and Running PES2MP

1. Download the zipped file from GitHub (link)
2. Goto "**makefile**" folder
3. Open **two** terminal windows (one for quick install and the other for conda install) and type the following commands: Users can install either the quick version or the normal version. The instructions are provided for both versions (read above). The quick version installation does not interfere with conda installation. The quick version installs packages without rigorously checking for inter-compatibility and is therefore faster to install. However, conda install ensures that all packages are compatible and are therefore slow to install.

   Terminal 1 (Quick Install) || Terminal 2 (Conda Install):

   ```
   $ chmod +x install_pes2mp_quick.sh          ||    $ chmod +x install_pes2mp.sh
   $ ./install_pes2mp_quick.sh                 ||    $ ./install_pes2mp.sh
   ```

4. Adding pes2mp command to bashrc/bash_profile/zshrc (Optional):

   After installation, the command to run the program can be shortened by adding a few lines in bash_profile/zshrc (Mac OS) or bashrc (Ubuntu). This is not essential but makes the program more user-friendly. Modify .bash_profile/zshrc (MacOS) or .bashrc (Most Linux including Ubuntu) in the terminal by typing:

   Mac OS || Ubuntu (Linux):

   ```
   $ open -t ~/.bash_profile                    ||    $ gedit ~/.bashrc
   ```

   Now the GUI text editor will open the bashrc/bash_profile/zshrc. Paste the following lines at the bottom of the text file, save, and exit.

   ```
   pes2mp ()
   {
           python3 pes2mp.py $1
   }
   ```

5. **To run the program on Terminal**:

   (a) **Copy** the **pes2mp.py** and **pes2mp_driver.py** files into a folder of your choice.

   (b) Now **copy the input file(s)** that you want to execute (e.g. pesgen1D.py) in that folder.

   (c) Open the **terminal** at the folder (make sure the conda base environment is showing).

   (d) **Activate** conda **environment** (pes2mp) using either command depending on installation

   ```
   $ conda activate pes2mp        ||     $ conda activate pes2mp_q
   ```

   (e) Now type the command in the terminal (e.g. Input file = pesgen1D.py):

   Step 4 completed || Step 4 incomplete:

   ```
   $ pes2mp pesgen1D                   ||     $ python3 pes2mp.py pesgen1D
   ```

   The first command works only if the bashrc/bash_profile/zshrc is edited. If somehow, the bashrc could not be found/edited, the users can still run the program using the longer script:

   Important: The input file must have the "**.py**" extension e.g. plot2D.py. However, while executing (running) the program, simply type **pes2mp plot2D** without the "**.py**" extension.

6. Users can keep multiple input files (within the same folder) with the same 'Project_name' variable (the variable is set inside the input files) to execute them in sequence such as:

   (a) PES Generation (Internal Psi4 or External Psi4, Molpro or Gaussian)

   (b) Optional: NN Augmentation and PES plotting

   (c) Optional: Fitting PES into a Function (for good fit use the same function as radial terms in the final step)

   (d) Multipole Expansion of PES, and

   (e) Fitting Radial Terms into a function (this automatically gives &POTL file for the functions to be used in MOLSCAT).

   After PES generation, the subsequent files can read either PES or Radial terms to execute the required task. The variable names for PES/Radial-Terms (and other options) are set inside the respective input file.

   The input file examples for GUI execution are provided inside **PES2MP_GUI** folder. The users can simply copy the whole folder depending on the task and rename the same. The environment name, folder location, and Project name are set in the GUI interface. For more details on using GUI, see the next Section.

2.2 PES2MP Input Files *vs* Standalone Python Scripts & Auxiliary Codes

2.2.1 PES2MP Input Files:

As discussed earlier, there are input files (in Python format) that are read by the PES2MP program. To run the PES2MP program, keep the 3 Python files i.e.

```
(1) pes2mp.py, (2) pes2mp_driver.py and (3) $inputfile.py
```

in the same folder. After which follow Step 5 of the previous subsection to activate the conda environment for pes2mp and run the program.

2.2.2 Auxiliary Codes:

For calculating the PES using external programs like Molpro, Psi4, and Gaussian, the batch scripts and provided in **/input_files/aux_scripts/** folder. Similarly, once the calculations are finished, a Python script to collect the results (in the required format) is provided as well.
To run auxiliary scripts use the following commands:

```
Python scripts : $ python3 script.py

Bash scripts   : $ chmod +x script.sh
                 $ ./script.sh
```

- Apart from PES calculations and results, there are similar scripts to generate Molscat input files, run them, collect results (cross-sections), and calculate rate coefficients for excitation and de-excitation transitions.
- There is also a Python script to convert energy from Hartree to cm^{-1} (which takes asymptotic energy of each angle to remove size-consistency error as faced by CI and F12 methods).

∗ For more information on input/output files, refer to the programmer's manual.

Section 3

Installation and Usage using GUI

3.1 Instructions for Installation

After downloading and extracting the zip file for PES2MP from GitHub, run the following command to open the PES2MP installer.

```
$ python3 installer_gui.py
```

This will open the following window:

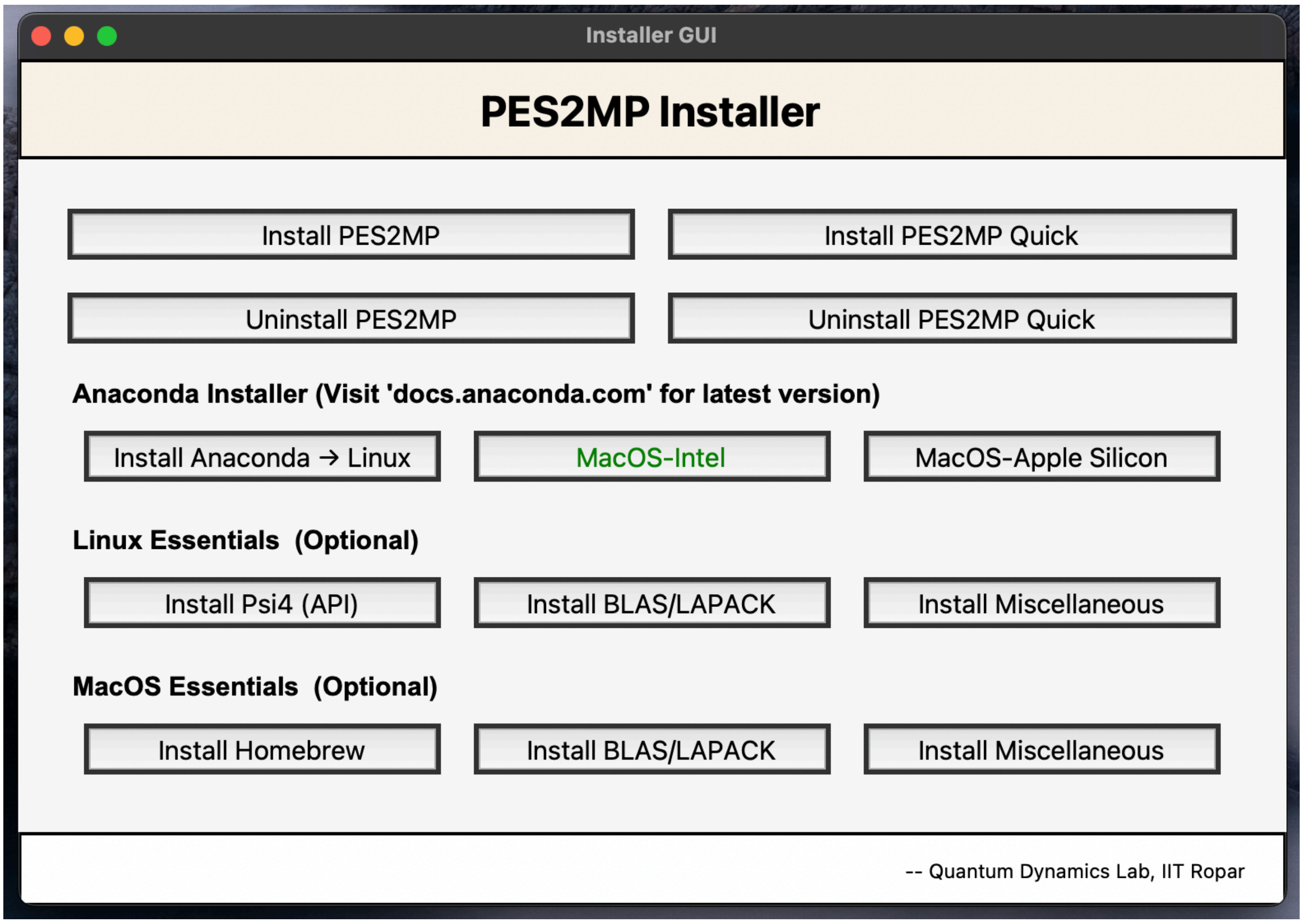


Figure 3.1: GUI Installer

- Select the 'Install Anaconda $->$' option to begin the installation process. Choose either Linux, MacOS (Intel), or MacOS (Apple Silicon) based on your hardware and OS. Once clicked, a new terminal will open and install Anaconda. Follow the on-screen instructions and select the appropriate options to finish the installation. Once installation is done, close the terminal.

The scripts for anaconda installs are present in *'/makefiles/anaconda_install'* folder. Users can also update the script for installing any newer version of Anaconda if the one present in the script is no longer maintained.

- Once anaconda is installed, click on either the 'Install PES2MP' or 'Install PES2MP Quick' option. A new terminal will open and the scripts automatically create environments and install the required Python packages needed for the PES2MP program. Follow on-screen commands and once the installation is finished, close the terminal. *To learn more about the difference between the two install options, see CUI installation.*

  `(OPTIONAL INSTALLS)`

- Linux users can also install Psi4 (Standalone API for external calculations) directly from here. For MacOS visit Psi4 website.

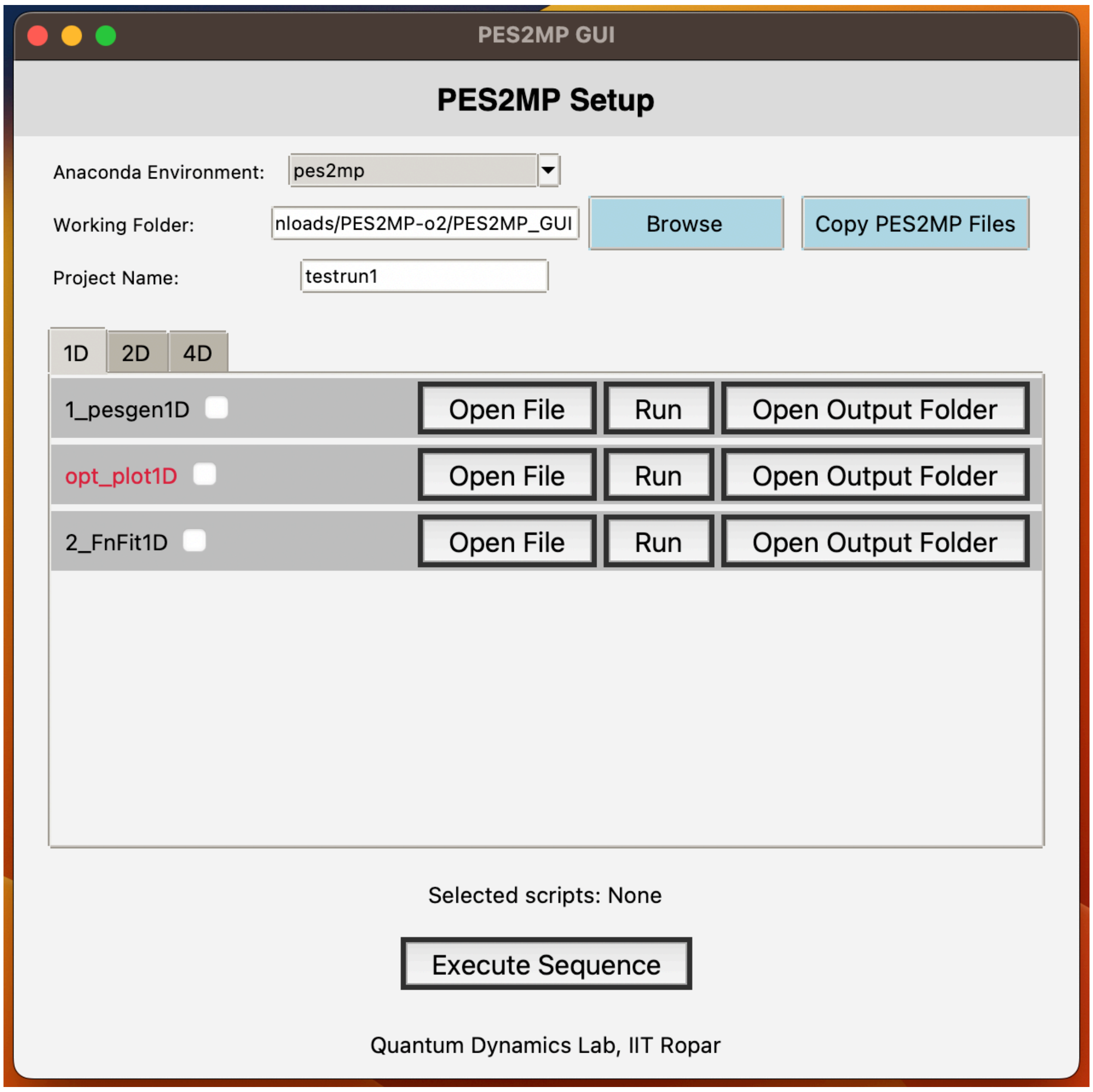
- The BLAS and LAPACK libraries (along with gfortran for Linux and gcc for MacOS) can be installed using the on-screen command. They are needed for installing MOLSCAT program. *(* **For MacOS users only** :: Homebrew an essential package manager for installing any library in MacOS. Therefore MacOS users must install Homebrew before installing BLAS/LAPACK or Miscellaneous!)*

- The miscellaneous packages contain several well-known packages like htop, cmake, etc. Users can open the *'/makefiles/macos(linux)_essentials'* folder to check the packages in *install_etc.sh* file.

Once PES2MP is installed, close the GUI installer. Type the script for opening the PES2MP GUI program.

```
$ python3 pes2mp_gui.py
```

This will open the following window:


PES2MP GUI

PES2MP Setup

Anaconda Environment: pes2mp

Working Folder: nloads/PES2MP-o2/PES2MP_GUI

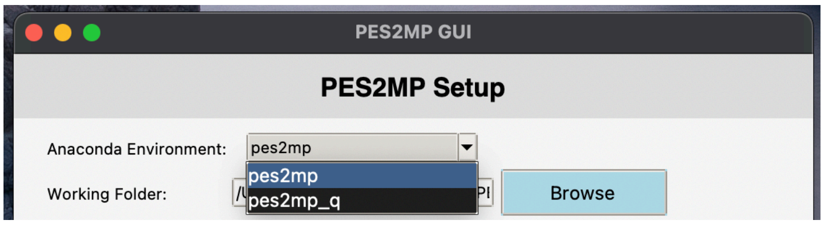
Project Name: testrun1

Selected scripts: None

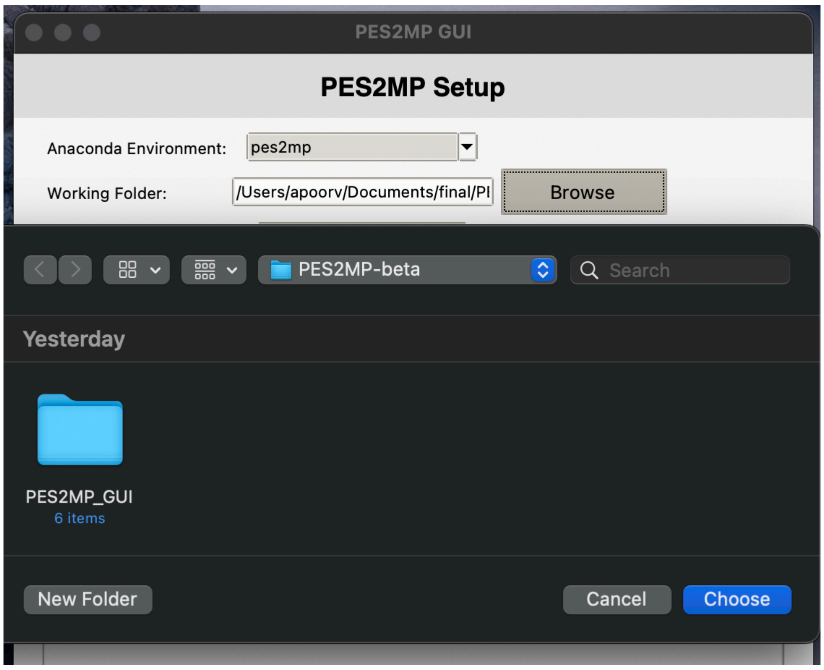
Quantum Dynamics Lab, IIT Ropar

Figure 3.2: GUI Installer

- Select the Anaconda Environment $- >$ pes2mp or pes2mp_q (for quick install version).



- Select the folder where input files for each module (1_PESGen1D.py, 2_FnFit1D.py, etc.) are located. * *The default directory is /PES2MP_GUI/.*



- Click on '*Copy PES2MP Files*' to copy *pes2mp.py* and *pes2mp_driver.py* files in the selected folder.
- Enter a Project Name such as '*testrun1*'. The separate project name ensures that output files from different runs are not overwritten. The output files are located inside '*/$Working_Folder/Projects/$Project_Name/*'.
- Users can open the input file by clicking on '*Open File*' to make changes in the system or associated parameters. To see output files, users can click on '*Open Output Folder*'

—————————————— How to use custom input files with GUI ——————————————

The input file names are tied to the GUI program i.e. they cannot be changed (as the GUI uses these names to distinguish between different input files). The example files for each 1D, 2D, and 4D collision with their numbered (1_PESGen1D.py) and optional files (opt_plot2D.py) are provided inside */PES2MP_GUI/* folder.

While running the program using CUI takes the project name from the input file, the GUI on the other hand, takes the project name in its interface.

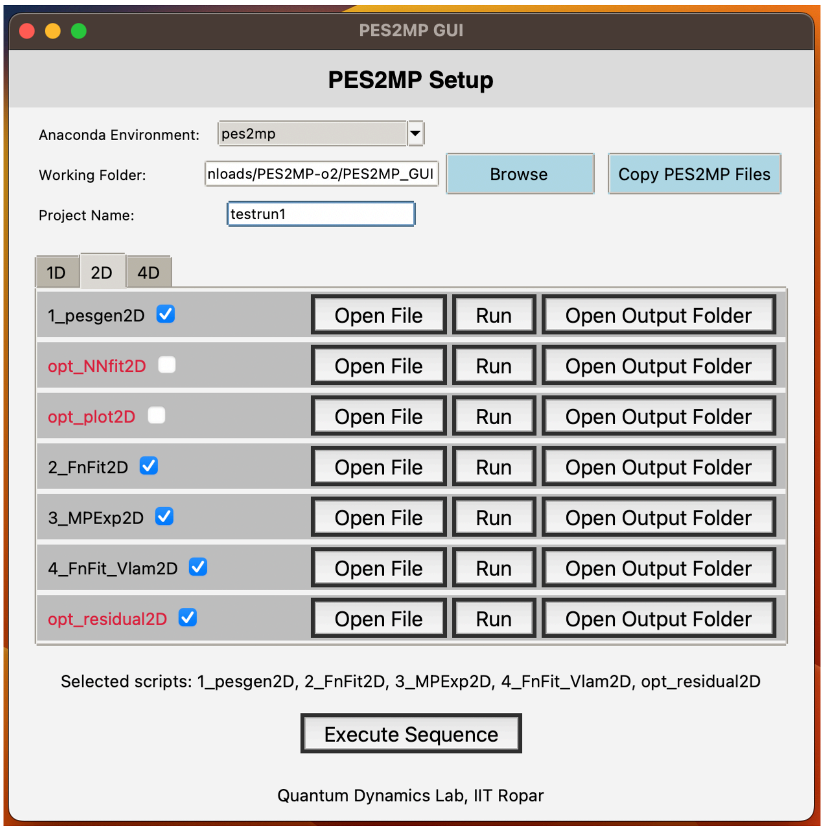
However, users who want to use input files provided inside the '*/input_files/*' folder with GUI, must keep in mind the following changes:

(a) Users can simply copy and paste the contents of the required input file into the GUI input file.

(b) The only change inside the GUI input file is the project name:

```
# CUI Project name CUI simply takes the project name in the input file
Proj_name  =  'testrun1_C2_He'   # This line has to be replaced!):
#-----------------------------------------------------------------------------------#
# Replace the above code with these two lines (Project name is read from GUI interface):
```

```python
    import os
    Proj_name  =  os.getenv("Proj_name", "default_project_name")
    #-----------------------------------------------------------------------------------------#
```



Anaconda Environment: pes2mp
Working Folder: nloads/PES2MP-o2/PES2MP_GUI
Project Name: testrun1

1_pesgen2D ✓
opt_NNfit2D
opt_plot2D
2_FnFit2D ✓
3_MPExp2D ✓
4_FnFit_Vlam2D ✓
opt_residual2D ✓

Selected scripts: 1_pesgen2D, 2_FnFit2D, 3_MPExp2D, 4_FnFit_Vlam2D, opt_residual2D

- Users can run the individual module using the ‘Run’ command or by selecting the tick option and running multiple modules in sequence. The sequence is shown above the ‘Execute Sequence’ button. If nothing is selected, the ‘Selected script:’ shows ‘None’.
- Clicking on different tabs selects different collision types, e.g. 2D and 4D collisions. The files for different collisions (1D/2D) can be placed in the same folder as different dimension files use different extensions i.e. 1D, 2D, and 4D.

Additional Information:

- The GUI program can run the individual modules of PES2MP like PESGen, FnFit, etc. either individually or in sequence as desired.
- If a module in sequence run fails (due to erroneous input), the previous module outputs are preserved. Therefore, the program can be restarted from the failed module in the next run.
- Since the input files are written in Python, users can use its format (e.g. # for comment) and packages (like os and numpy) as needed.
- The Psi4 package used internally is installed with PES2MP (no separate installation is needed).
- The GUI generates a log file inside */Projects/* folder for each run (the filename is the same as the project name with the prefix ‘outv’).

∗ For more information on input/output files, refer to the programmer’s manual.

Section 4

Capabilities

The input files for each are provided with the program.

1. **PES Calculation** :: **1D/2D/4D rigid rotor PES.**
   Simple templates (input_files.py) are provided for Psi4, Gaussian, and Molpro which automatically creates required input files (for each individual coordinate in XYZ format) for respective *ab initio* software packages. Sample input files are provided for Single point, BSSE-corrected, and CBS-extrapolation schemes for Molpro. Gaussian template defaults to BSSE corrected PES, while Psi4 can be used both internally (using pes2mp) or externally (recommended for large calculations).

2. **Plots** :: **Plot PES, residual and curve fits.**
   PES plots are available as 1D (R vs E) and polar (R,θ vs E). For 2D/4D PES, certain angle combinations can be chosen for plots to prevent cluttering. Residual (error) plots can be automatically generated for NN/radial-terms generated PES. The curve fit with various functions can also be visualized along with any extrapolation function (if used). Apart from these, many other relevant plots are also provided for the NN module.

3. **NN augmentation** :: **Create NN model to augment and/or get missing data points.**
   The program can be utilized to separately fit minima and high-energy regions for a good fit. The underlying package for the NN model is TensorFlow, which can learn multiple outputs simultaneously and is not restricted by dimensionality (i.e. data can have N inputs and M output columns). The program also keeps boundary elements (of input features) in the training dataset to prevent boundary errors. There are two models: PES-specific (default) and Generic. The PES-specific model uses Gaussian activation functions (custom-made) and a modified Slater function for high-energy and long-range extrapolation. In the generic model, GELU is the default activation function. For finding the most suitable model (layers/nodes/activation function), Keras-tuner's (package) Bayesian Optimization tuning is used which utilizes the Gaussian process to find optimum parameters within a reasonably defined search space.

4. **Multipole expansion** :: **Calculate radial terms (v_Λ).**
   The radial terms (v_Λ) are obtained by expanding the interaction potential (i.e. 2D/4D PES) in a set of orthogonal functions (Legendre [2D] and two Spherical harmonics [4D]) of the internal coordinates ([R, θ] and [R, ϕ, θ_2, θ_1]). The current implementation takes PES (must not have any coordinate missing) and multiples it with the pseudo-inverse matrix containing Legendre/Spherical harmonics coefficients to get radial terms. Warning: The number of angles must not be less than λ as it will result in poor fit which can be visualized by recreating the PES.

5. **Curve fit** :: **Fit 1D PES and/or radial terms into analytical expression**.
   The radial terms (v_Λ) can be fitted into a series of power/exponential functions provided with the pes2mp package or can be defined in the sample template for a more complex fit. Extrapolation schemes (R^{-6}, R^{-4}, R^{-3}, R^{-x}, etc.) are also available with plots (optional). After fitting the PES into MOLSCAT readable functions, the PES is recreated (with a residual plot) to verify whether a good fit is obtained or not.

6. **MOLSCAT readable output** :: **Available for general-purpose version of POTENL subroutine**
   The curve fit produces an output thatß is MOLSCAT readable for direct utilization in **&potl** block. The feature is available in a general-purpose version of the subroutine POTENL of MOLSCAT 2020. The same can be utilized to calculate cross-sections for rotational (de-)excitation of two colliding species at cold and ultracold temperatures.