autoPES Package

User’s Guide

Version 2020.1

Michael P. Metz, Konrad Piszczatowski, and Krzysztof Szalewicz

Department of Physics and Astronomy,
University of Delaware, Newark, Delaware 19716

December 26, 2019
Contents

1 Introduction 3
  1.1 Monomer flexibility 4

2 Changes since version 2016.1 4
  2.1 New in version 2016.2 4
  2.2 New in version 2020.1 6

3 Installation 6

4 Basic usage 7
  4.1 Control file 8
  4.2 System specification file 8
  4.3 Running the program 14
  4.4 Special run modes 15
  4.5 Using the results
     4.5.1 Compute the fitted interaction energy of a PES 16
     4.5.2 Convert to Cartesian coordinates 17
     4.5.3 Create DL_POLY inputs 17
  4.6 Parallelization 18
  4.7 Basis sets 18

5 Potential Energy Surface 19
  5.1 Rigid Monomer Case 19
  5.2 Flexible Monomer Case 20
  5.3 Intramonomer PES 21

6 Graphical Energy Plotting Utility 21

7 Options 22

8 Structure of the program 29
  8.1 Program flow 29
  8.2 Directory structure 33
  8.3 Internal file formats 33
  8.4 Programs and scripts in the autoPES suite 42
1 Introduction

The autoPES package automates the development of accurate symmetry-adapted perturbation (SAPT) based intermolecular potential energy surfaces (PES), also known as force-fields, of rigid or partially flexible molecules. It is also possible to use the coupled cluster method with single, double, and noniterative triple excitations [CCSD(T)] to compute interaction energies instead of SAPT. It is an implementation of the methodology described in Ref. [1]. The current version autoPES can generate potentials using SAPT based on density functional theory [SAPT(DFT)] [2, 3] or CCSD(T). The PES generation procedure can be broadly divided into six steps:

1. Grid point generation – The generation of a set of dimer configurations which adequately represent the most important regions of intermolecular interaction.

2. Ab initio calculations – Computing interaction energies at each of the generated grid points.

3. Asymptotic calculations – The potentials generated by the autoPES package seamlessly integrate long-range and short-range interactions. The long-range components of the PES require separate ab initio calculations, which are done in this step.

4. Parameterization – The data from the ab initio calculations at grid points and the asymptotic calculations are combined, and the parameters of the PES are optimized.

5. Hole fixing – Due to the functional form of the fit, it is possible that the PES lacks a physically accurate repulsive wall at short-range dimer configurations. This step automatically corrects these ‘holes’ by performing additional ab initio calculations as necessary.

6. Iterative improvement – Based on the results of the parameterization step, the autoPES package automatically determines if any additional grid points are required, and if so performs the additional calculations and fits.

The autoPES package does not itself perform ab initio calculations, but rather interfaces with the SAPT package [4]. The SAPT package in turn interfaces with external front-end programs to perform monomer ab initio calculations. In addition, the autoPES package interfaces directly with these front-end programs to perform certain steps of the PES generation. The current version of autoPES requires both Orca [5] and Dalton [6] to run, in addition to the SAPT package.

Due to the computational nature of PES generation, autoPES is designed to interface with job queuing systems on computer clusters by dynamically submitting jobs as required. This is in contrast to the usual way of manually submitting a single job to the queuing system, although autoPES can also be configured to run in the more typical way. See Sections 3 and 4.3 for details.
1.1 Monomer flexibility

In the simplest case, a dimer can be modeled as two perfectly rigid monomers, which are each kept at some fixed reference monomer geometry while able to rotate and translate relative to each other. In cases where the monomer geometries deform significantly under the effect of intermolecular forces, this becomes a severe approximation. The type of PES handled by autoPES can handle these internal monomer deformations in a sequence of increasingly complex stages.

In the first stage, no additional steps are taken in the development of the PES, but the fit is still treated as flexible. This is possible due to the site-site form of the fit, which is well-defined for arbitrary monomer geometries and inherently encodes some physical information. This approach will typically produce qualitatively reasonable results only for small deviations from the reference monomer geometry, but can be used when very high accuracy is not crucial.

As of autoPES version 2020, the second and third stages of handling monomer flexibility are supported. In these stages, some number of internal degrees of freedom (IDOFs) of the monomer are specified by the user (see Sec. 4.2). In the second stage, the same functional form of the fit is used as in the first stage (i.e. there is no explicit functional dependence on the IDOFs). However, the data set to which the PES is fitted is generalized to include dimer configurations in which the monomer is deformed according to the specified IDOFs. This second stage of handling monomer flexibility allows the fit to remain qualitatively accurate for configurations in which the monomers are deformed significantly from the reference geometry. However, due to the simplicity of the functional form, the overall fitting error can be expected to increase relative to the rigid monomer case.

The third stage of handling monomer flexibility uses the generalized data set of stage two, while additionally including explicit dependence of the functional form of the fit on the IDOFs (see Sec. 5.2). This approach is the most complex, but it allows for highly accurate PESs throughout the entire specified range of IDOFs.

2 Changes since version 2016.1

2.1 New in version 2016.2

• Support for explicit IDOFs added and the functional form of the fit has been extended accordingly.

• Introduced support for MOLPRO interface for supermolecular calculations.

• Introduced complete basis set extrapolation (CBS) option for supermolecular calculations.

• The formula used to determine the default number of grid points has been changed to
account for cases where the monomers are large, but have a very high degree of permutational symmetry. The formula had been changed from \( N_{\text{grid}} = 6N_{\text{FP}} \) to \( N_{\text{grid}} = 5N_{\text{FP}} + 5(N_{\text{atomA}} + N_{\text{atomB}}) \), where \( N_{\text{FP}} \) is the number of free parameters in the functional form of the fit and \( N_{\text{atomA}} \) and \( N_{\text{atomB}} \) are the numbers of atoms in the monomers.

- Dependence of the fit weight on the value of the fit (see Eq. 19 of Ref. [1]) has been removed due to convergence problems with some systems.

- A penalty function based on CHELPG\([\ast]\) has been added to the partial charge fitting procedure to constrain the charges to have more physically reasonable values.

- Grid point placement near local minima now rejects points which are close to existing dimer configurations (this was always done for the main grid generation).

- Grid point distribution now includes a factor based on the dimer geometry in addition to the existing factor which depends on the interaction energy. This is so that the initial iteration will produce a reasonable distribution of grid points even in the case that no guiding potential is specified (i.e. if all \( t_{Ai} \) are set to 0).

- Default value of the TEST.PCT option increased from 20% to 30%.

- Hole searching criterion for the case that the PES value is found to be less than 1.1 times the lowest \textit{ab initio} value has been changed. The point will now only be considered a hole if the value is less than 1.3 times the lowest \textit{ab initio} value (see Sec. VII of Ref. [1]).

- The density-fitting approximation has been disabled by default when using the ORCA interface with supermolecular calculations.

- Overall monomer polarizability is no longer computed using ORCA. The overall monomer polarizability is now unconstrained, unless a value is given using the POLARIZABILITY.A or POLARIZABILITY.B options.

- An incorrect column size in the generated DL.POLY input files has been corrected.

- The iteration summary table now includes more information.

- PESWalk minimum search replaced by genetic algorithm based on a modified version of Pikaia\([7]\).

- Various minor bug fixes.
2.2 New in version 2020.1

- Implemented rotational and translational IDOFs with appropriate extensions to the functional form of the intermolecular PES.
- Implemented automatic generation of intramolecular PESs.
- Implemented Iterative Variance Minimizing Grid (IVMG) method.
- Inclusion of Dimerplot graphical utility.
- The basis set repository now lives in the SAPT package rather than in autoPES.

3 Installation

Installation of autoPES requires the following steps:

1. The following software must first be installed on the system:
   - Orca 3.0.1 [5]
   - Dalton 2.0 [6] (only if COM-COM asymptotics is used)
   - MOLPRO 2010 or newer [8] (only if MOLPRO interface is used)
   - SAPT2020 [9]

   Note that Dalton 2.0 must be patched using the script included in the SAPT package in order to use GRAC asymptotic correction, see the SAPT manual.

   It is recommended that the SAPT installation be tested with both ORCA and Dalton interfaces using the included examples before attempting to use autoPES.

2. To compile autoPES, first edit the $PATH/compall script and set the variables BLAS, TARGET, and OMP. Here $PATH is the autoPES installation directory, which is extracted from the archive file available on the web.

   - TARGET specifies the FORTRAN 2003 compatible compiler to use. The autoPES package has been tested with the Intel ifort version 18.0 and GNU gfortran version 8.2.0 compilers.
   - BLAS is a string of compiler arguments specifying an appropriate Basic Linear Algebra Subprograms library. This BLAS should be configured to run in a serial (single-threaded) mode, and its performance is not crucial to the overall performance of PES generation.
OMP specifies whether autoPES will be compiled with OpenMP parallelization enabled. See Sec. 4.6 for details. OMP should be set to either ‘YES’ or ‘NO’.

After these variables are specified, simply run the compall script from the autoPES root directory. The clean script may be used to remove compiled files in order to perform a clean installation.

3. The file $PATH/bin/vars contains paths to the installations of Orca, MOLPRO, and SAPT installed on the system. These variables must be set manually before running autoPES.

4. Finally, the file $PATH/bin/submit.sh must be modified to interface with the job submission system present on the specific computer system. This script must ensure that the specified resources are available, copy the specified input files to an appropriate scratch directory, perform the specified work, and finally copy the specified output files back. Details of each step are given in the comments of the provided example submit.sh.EXAMPLE file.

On most large computer systems, running a job requires one to create a job script, then submit it to a queuing system with a command such as qsub. Because of the complexity and large computational cost of PES generation, it is often undesirable to use autoPES directly in this way. Rather, the submit.sh file dynamically creates the job scripts, and autoPES submits multiple jobs as necessary.

To perform this step of installation, the user should first ensure that he/she knows how to properly run jobs on the given computer system. The $PATH/bin/submit.sh.EXAMPLE file should then be copied to $PATH/bin/submit.sh, and the latter file should be modified to run the required jobs in a way appropriate for the system. In addition to performing the required work, the submit.sh file must create appropriate ‘marker’ files in order for autoPES to know when a job has started or completed running. Additionally, depending on the computer system, the submit.sh file may need to transfer files to an appropriate scratch directory and/or provide commands to the queuing system specifying required resources. These steps are outlined and described in the comments of the provided submit.sh.EXAMPLE file.

The submit.sh file may also be configured to run jobs directly, rather than submitting to a queuing system. This may be useful for generating potentials on a workstation or other small computer system, but this is only recommended for small dimers.

4 Basic usage

The autoPES program requires two input files. The first, NAME.input, specifies the molecular structure of each monomer. The second, NAME.ctrl, specifies various options of the desired PES
and computation control parameters. Here NAME is the name of the system, and can be any string up to 20 characters long.

4.1 Control file

The NAME.ctrl file should be prepared in the standard FORTRAN name-list format. The simplest NAME.ctrl file would look like

```
&GENERAL
/
&AB_INITIO
/
&GRID_GENERATION
/
&ASYMPTOTICS
/
&MAIN_FIT
/
&HOLE_FIXING
/
&SYS_SETTINGS
/
```

In this case, default values are used for all options in all seven categories. All available options are described in Sec. 7. In addition, the examples provided with autoPES may be used as a template.

4.2 System specification file

The NAME.input file has the following structure

```
*MONOA
N_A q_A
a_1 x_{A_1} y_{A_1} z_{A_1} q_{A_1} s_{A_1} t_{A_1}
a_2 x_{A_2} y_{A_2} z_{A_2} q_{A_2} s_{A_2} t_{A_2}
...
**CHIRAL
**IDOF
N_{A_{IDOF}}
```

8
The **NAME**.input file is organized into sections, with the start of each section indicated by an asterisk followed by the section label. Additionally, sections may contain subsections which are indicated by two asterisks followed by the subsection label. The sections **MONOA** and **MONOB** are required, while all other sections and subsections are optional. The placement of the sections within the input file is arbitrary.

- The format of section **MONOA** is organized as follows: the first line contains two whole numbers: $N_A$, specifying the number of atoms in monomer A, and $q_A$, specifying the charge of monomer A. In the following $N_A$ lines the information about the molecule A should be given. All
columns are free-form, i.e., not sensitive to the number of digits or spaces between numbers. The first column contains a string of characters $a_i$, which is the name of the given atom or off-atomic site in the molecule $A$. The chemical or any other symbol up to 5 characters can be used, however a symbol for each atom in a given molecule has to be unique. So, for example, if monomer $A$ is a water molecule, then the hydrogen atom can be denoted by its chemical symbol ‘O’, but the two hydrogen atoms must have distinctive symbols, e.g., ‘H1’ and ‘H2’.

Columns 2, 3, and 4 contain the Cartesian coordinates of the atom $(x_{Ai}, y_{Ai}, z_{Ai})$.

Column 5 contains the nuclear charge $q_{Ai}$ of the atom, given as a whole number.

Column 6 contains the ‘symmetry equivalence index’ $s_{Ai}$ of the atom. This is a positive integer which should have the same value for those atoms which are equivalent through symmetry. For example, in the water molecule one should assign the same indices to both hydrogen atoms. Note that “symmetrically equivalent” refers not only to the atomic number of an atom but also its position within the molecule. In some cases, where atoms are not exactly symmetrically equivalent but are approximately so, it may be advantageous to assign them the same $s_{Ai}$ value. This will sacrifice some adjustability of the fit, but reduce the number of free parameters and so reduce the number of grid points required.

Finally, column 7 contains an atom type number $t_{Ai}$, which specifies the atom type of the initial guiding force-field. In the current version of autoPES, the user must manually determine the atom type. This may be accomplished by reading the descriptions of atom types given in the file $\$PATH/ff/FORCE\_FIELD$, where $\$PATH$ is the installation directory of autoPES and $\$FF\$FIELD$ is the force-field name specified by the option given in the $NAME.\$ctrl$ file. For example, in the case of the water molecule, if $\$FF\$FIELD$ is set to ‘oplsaa’, then the oxygen atom and hydrogen atoms could use $t_{Ai}$ equal to 63 and 64, respectively. If a value of 0 is given for $t_{Ai}$, then the atom will not contribute to the initial guiding force-field. It is possible to use autoPES without a guiding force-field by setting all $t_{Ai}$ to 0, but this may result in a sub-optimal initial grid point distribution.

- The section MONOB may follow the same format as MONOA. In the case that monomer A and monomer B are identical, the section MONOB should instead consist of a single line with the keyword ‘SAME’. If the monomers are different, the symmetry equivalence indices $s_{Bi}$ have to be kept unique through out the whole dimer. In other words, if the same symmetry equivalence indexes are given to an atom from the monomer A and an atom form the monomer B, they will be treated as symmetry equivalent.

- The subsection CHIRAL may be included in sections MONOA and/or MONOB. It indicates that the monomer is chiral, and that the potential should be generated for both same-handed and
opposite-handed interactions. The CHIRAL subsection contains no additional lines except the subsection label.

- The subsection IDOF may be included in sections MONOA and/or MONOB. It defines any internal degrees of freedom (IDOFs) of the monomer (excluding chirality). The first line contains the number $N_A^{IDOF}$ of IDOFs. The remainder of the IDOF subsection is organized into triples, with each IDOF specified by three lines:
  - The first line for each IDOF contains three or four entries, depending on the type of IDOF. The value $T_{iA}$ is a string defining the type of the $i$th IDOF. It can be either ‘LIN’, indicating a linear translation of atoms along some axis ‘ANG’, indicating a rotation of atoms about some axis, or ‘BOND’, which is an alternative rotational IDOF. The values $a_{i1}^{\text{vect}}$ and $a_{i2}^{\text{vect}}$ define the axis of translation or rotation to be the unit normal vector in the direction from the atom $a_1$ to the atom $a_2$. These are given as the atom labels as defined in the MONOA section. In the case of ‘BOND’ IDOF, the three atoms $a_{i1}^{\text{vect}}$, $a_{i2}^{\text{vect}}$, and $a_{i3}^{\text{vect}}$ define a plane, with the axis of rotation going through atom $a_{i2}^{\text{vect}}$ and normal to the plane.
  - The second line for each IDOF contains three entries. The value $N_{iA}^{atom}$ indicates the number of atoms to be translated or rotated in the $i$th IDOF. The values $X_{A,\text{min}}^2$ and $X_{A,\text{max}}^2$ give the minimum and maximum of the range of values that the IDOF can take, in degrees or angstroms. If both $X_{A,\text{min}}^2$ and $X_{A,\text{max}}^2$ are set to 0, an automatic default procedure is used to find the limits, see Sec. II of Ref. [?].
  - The third line for each IDOF contains $N_{iA}^{atom}$ entries, $a_{i1}^{\text{set}}$ through $a_{iN_{iA}^{atom}}^{\text{set}}$, which define which atoms are in the set.

- The ATOMFOLLOW subsection specifies atom following modes for off-atomic sites in flexible monomers. The first line contains the number $N_A^{AF}$ of off-atomic sites which are bound to atom motion, with each remaining line specifying one such site.
  Each atom-following site is defined by 4 or 5 entries, depending on the atom following mode. The label $a_{a}^{\text{oa}}$ specifies the off-atomic site. The value $F_{iA}$ is a string defining the type of the $i$th atom following site. It can be either ‘PLANE’, indicating a planar atom following mode, or ‘BOND’, indicating a bond atom following mode (see Ref. [?] for details of these modes). In the bond case, the values $a_{at1}^{\text{at}}$ and $a_{at2}^{\text{at}}$ specify the two atoms that the off-atomic site remains between. In the planar case, the three values $a_{at1}^{\text{at}}$, $a_{at2}^{\text{at}}$ and $a_{at3}^{\text{at}}$ specify three atoms that form a plane, with the distance from the off-atomic site to the plane kept fixed.

- The UNIT section specifies what units of length are used to specify the geometry in the MONOA
and MONOB sections. This section should consist of a single line, containing either the keyword ‘angstrom’ or ‘bohr’. If the UNIT section is omitted, units of Angstrom are assumed.

- The COMPONENTS section is optional and specifies which components of the complete functional form are to be included for each atom type. Each line of this section corresponds to a single symmetry equivalence index \( s_i \). The electrostatic, polarization, induction plus dispersion, and exponential components of atom type \( s_i \) are switched on or off by setting the respective logical values \( P_{e,s}^i \), \( P_{p,s}^i \), \( P_{i+d}^i \), and \( P_{e,x}^i \) (the \( r^{-12} \) term, if used, is considered part of the exponential component). If this section is omitted, all values will default to true, except for polarization and induction plus dispersion components of off-atomic sites (see below). In order to include the polarization model, the option POLARIZABLE must also be enabled in the NAME.ctrl file.

Below are examples of the NAME.input files for two different rigid-body systems, one with the same monomers, and the other one with different.

**H2O2.input:**

```
1  ∗UNIT
2    bohr
3  ∗MONOA
4    3 0
5  O    0.0000  0.0000  0.1246  8 1 63
6  H1   −1.4365  0.0000 −0.9971  1 2 64
7  H2   1.4365  0.0000 −0.9971  1 2 64
8  ∗MONOB
9    SAME
10 ∗COMPONENTS
11  1 T T T T
12  2 T F T T
```

**H2ONH3.input:**

```
1  ∗MONOA
2    3 0
3  O    0.0000  0.0000  0.1246  8 1 63
4  H1   −1.4365  0.0000 −0.9971  1 2 64
5  H2   1.4365  0.0000 −0.9971  1 2 64
6  ∗MONOB
7    4 0
8  N    0.0000  0.0000  0.0000  7 3 78
9  H1   0.0000 −1.7720 −0.7211  1 4 79
10 H2   1.5346  0.8861 −0.7211  1 4 79
11 H3   −1.5346  0.8861 −0.7211  1 4 79
12  ∗UNIT
13    bohr
```

In addition to atoms, it is possible to add off-atomic sites to the system geometry. Such sites do not affect ab-initio calculations, but generalize the functional form of the PES. These are
typically used on small systems where the atom-atom separations alone are insufficient to represent the PES. To include an off-atomic site, the same syntax should be used as for an atom, but with the values of \( q_i \) and \( t_i \) set to zero. The symmetry index \( s_i \) must still be specified as with atoms. Off-atomic sites must always be listed at the end, after the atoms. In the current version of autoPES, the locations of off-atomic sites are not optimized automatically, but must be specified by the user.

Below is an example NAME.input file for a fully flexible water molecule interacting with a partially flexible ethylene glycol molecule.

**EG-WAT.input:**

```
*UNIT
bohr
*MOL A
12 0
C1 -1.43586075 0.00000000 0.00000000 6 1 99
C2 1.43586075 0.00000000 0.00000000 6 1 99
O1 -2.44150180 2.47139335 0.00000000 8 2 96
O2 2.44150180 -2.47139335 0.00000000 8 2 96
H1 -2.13070300 -1.00916300 1.66332700 1 3 98
H2 -2.13070300 -1.00916300 -1.66332700 1 3 98
H3 2.13070300 1.00916300 1.66332700 1 3 98
H4 2.13070300 1.00916300 -1.66332700 1 3 98
HO2 2.13857197 -3.21066850 -1.62520090 1 4 97
HO1 -2.13857197 3.21066850 1.62520090 1 4 97
XC1O -0.81383390 0.82379780 0.00000000 0 5 0
XC2O 0.81383390 -0.82379780 0.00000000 0 5 0
*IDOF
3
*ANG C1 C2
1 -180.0 180.0
O2 H3 H4 HO2
*ANG C1 O1
1 0.0 0.0
HO1
*ANG C2 O2
1 0.0 0.0
HO2
*ATOMFOLLOW
2
XC1O PLANE O1 C1 C2
XC2O PLANE O2 C2 C1
*MOL B
8 0
O1 0.00000000 0.00000000 0.11346859 8 6 63
H1 -1.43578546 0.00000000 -0.99147497 1 7 64
H2 1.43578546 0.00000000 -0.99147497 1 7 64
D11 0.00000000 0.20672130 -0.24625894 0 8 0
D12 0.00000000 -0.20672130 -0.24625894 0 8 0
D21 0.00000000 0.20000000 0.25090156 0 9 0
```
4.3 Running the program

Both input files should be located in an otherwise empty directory and the program called with command

```
$PATH/bin/autoPES NAME
```

where PATH is a path to the main autoPES directory. The script will generate several files and subdirectories containing outputs of various parts of the PES development. These can be expected to total about 500 MB to 5 GB in size, depending on the size of the system and number of grid points and iterations used.

The autoPES program may simply be run once, and remain running for the duration of the PES generation. However, because this may take days or even weeks depending on the size of the dimer and the amount of computer resources available, it may be impractical to use autoPES in this way. Therefore, it is possible to terminate the program midway through PES generation and resume running it at a later time. When the autoPES program is called, it will identify the appropriate next step to take based on the contents of the working directory from which it was called. It will then submit the appropriate job, or, if a job is in progress, wait until that job completes. When waiting on a job to complete, autoPES will print a line starting with the word ‘Awaiting’. This indicates that it is safe to kill the process and restart it at a later time. If the script is terminated when the last line printed does not begin with ‘Awaiting’, there is a possibility of corrupting some part of the PES generation process, and in this case manual user action may need to be taken before restarting autoPES.
Each first-level subdirectory in the autoPES working directory is associated with a slave script, which handles all computations related to that task. When a given task ‘X’ is complete, the slave script creates a SLAVE_X_DONE file in that subdirectory. Additionally, in order to interface with a job queuing system, autoPES is signaled that a given job ‘X’ is running or complete by the creation of ‘marker’ files X_RUNNING or X_DONE. For example, once the asymptotics task is complete, the autoPES working directory will contain a subdirectory ASYM, which contains the files SLAVE_ASYM_DONE, ASYM_DONE, and ASYMFIT_DONE. If a job is terminated before it completes, autoPES will think that the job is still running. In this case, manual action must be taken to re-run the appropriate job, see below.

Individual components of the PES calculation may be recomputed by removing or renaming the appropriate directory within the main running directory and then restarting autoPES. These directories are described in detail in Sec. 8. In addition, an individual SAPT calculation $i$ may be recomputed by removing the corresponding $\text{SAPT}_n/\text{PART}_i$ subdirectory. In the case that the entire batch $n$ of SAPT calculations has already completed, the file SAPT_n/SLAVE_SAPT_DONE must also be removed. See also the ‘clean’ option described in Sec. 4.4.

The user may wish to alter various options part way through the PES generation. This may in general be done by exiting autoPES as described above, then making the appropriate changes in the NAME.ctrl file. The changes will take effect at the next part of the PES generation which they affect. In order to alter some previously-computed part of the PES generation, the user must remove that directory in addition to making changes to the NAME.ctrl file. Specific options may require additional precautions when changed part way through a PES generation. Any such precautions are described in Sec 7.

The user may want to modify the functional form of the fit in NAME.input file midway through a PES generation in order to make changes to the fit. While the atomic coordinates should never be altered, the values $s_i$ and $t_i$, as well as off-atomic sites may be changed. If any of the $s_i$ values or off-atomic sites are altered, the asymptotic fitting must be re-run. This may be accomplished by removing the files ASYM/SLAVE_ASYM_DONE and ASYM/ASYMFIT_DONE.

### 4.4 Special run modes

In addition to the normal operation outlined above, it is possible to run autoPES in special modes which may be useful. This may be done with the command

```
$PATH/bin/autoPES NAME MODE
```

where MODE is one of the following:

1. **clean** – When run in ‘clean’ mode, autoPES will, when encountering an incomplete batch of SAPT calculations, first prompt the user before removing all incomplete or failed calculations
in that batch. If a batch is already completed and the user wants to re-run failed calculations, he/she should first remove the corresponding SAPTn/SLAVE_SAPT_DONE file. Note that this will retain any calculations which have already been marked as complete, and will only remove failed or incomplete calculations.

2. **compress** – When run in ‘compress’ mode, autoPES will, when encountering a completed iteration of *ab initio* calculations (i.e., a SAPTn or DELTAHFn directory containing the file SLAVE_SAPT_DONE), place the output files and directories from those calculations into compressed archives and remove the original, uncompressed files. This option is useful if the PES files are taking an excessive amount of disk space. Using the ‘compress’ feature does not compress the grid point coordinates or interaction energies which are used by autoPES and will have no effect on the functioning of autoPES. However, if the user later wishes to view the *ab initio* output files, they must first manually decompress the appropriate SAPTn/archive.tar.gz file.

3. **xyz** – When run in ‘xyz’ mode, autoPES will attempt to add geometries from .xyz files to the set of grid points. Before running this, the user should create a directory XYZ in the autoPES running directory, and place the desired .xyz files there. Each unique dimer geometry should be in its own .xyz file in the standard format, with the number of atoms equal to the number of atoms in the dimer. The order of atoms in the file is arbitrary. The program will report which files were successfully identified as valid dimer geometries and create a new SAPTn directory containing the new points. autoPES must be re-run as normal (i.e., without the ‘xyz’ run mode) to begin the actual SAPT calculations. After completing the SAPT calculations, autoPES will then re-fit the PES.

### 4.5 Using the results

The results of each fit may be found in the FITn directories. The directory with the largest n is typically the most recent fit. The file FITn/fit_report.out contains a description of the fit in human-friendly and LaTeX formats. The file FITn/fit_inter.dat contains all intermolecular fit parameters in a machine-friendly format. The structure of the fit_inter.dat file is described in Sec. 8.3.

#### 4.5.1 Compute the fitted interaction energy of a PES

The program eval_pes can be used to evaluate the value of the potential at a given geometry. To use eval_pes, first prepare a file in .xyz format containing the dimer geometry, with the atoms appearing in the same order that they do in the NAME.input file, with monomer A first and monomer B second. The program may then be used with the command
$PATH/bin/eval_pes $PARM_FILE < $GEO_FILE

where $PARM_FILE is the generated fit_inter.dat file and $GEO_FILE is the .xyz geometry file. The result is sent to the standard output.

4.5.2 Convert to Cartesian coordinates

The program toxyz can be used to convert from the $R$ and Euler angle coordinates used by autoPES to Cartesian coordinates in the standard .xyz format. The toxyz program must be run from a directory containing the NAME.input_long file (see Sec. 8.3). The coordinates are given in the same format as the NAME.ener file (see Sec. 8.3) to the standard input. In the rigid-monomer case, there are six coordinates. The program may be run with the command

```
echo '5 0 0 0 0 0' | $PATH/bin/toxyz $NAME
```

where $NAME is the name of the system. In this example, the COM-COM distance is 5.0 Å and all Euler angles are set to zero degrees, meaning neither monomer is rotated from its starting orientation. The result is sent to the standard output.

In order to evaluate the PES at a configuration specified by $R$ and Euler angles, the eval_pes and toxyz programs may be chained together like so:

```
echo '5 0 0 0 0 0' | $PATH/bin/toxyz $NAME | $PATH/bin/eval_pes $PARM_FILE
```

In the flexible-monomer case, the eval_pes program takes a second argument $INTRA_FILE containing the parameters of the intramolecular fits.

4.5.3 Create DL_POLY inputs

The program make_dlpoly_inp can be used to create input files for the DL_POLY molecular dynamics simulation package [10]. Currently, this option does not support the polarization model. The make_dlpoly_inp program creates two input files: TABLE and FIELD. The TABLE file contains a description of the PES, excluding electrostatics and polarization components, in a tabulated form. The FIELD file contains molecular geometries and partial charges. The TABLE file generated by make_dlpoly_inp is typically usable as-is, but the FIELD file should be treated as a template only and will generally require some modification depending on the intended application. For a full description of these files, see the DL_POLY user manual [11]. The program may be run with the command

```
$PATH/bin/make_dlpoly_inp $NAME $PARM_FILE $R_CUTOFF
```

where $NAME is the name of the system, $PARM_FILE is the generated fit_inter.dat file, and $R_CUTOFF is the potential cutoff radius used by DL_POLY, specified in Angstrom.
4.6 Parallelization

To enable OpenMP parallelization of certain processor intensive programs in autoPES, the OMP option in the \$PATH/compall script should be set to ‘YES’. This does not affect the SAPT dimer calculations, which consume the vast majority of the total CPU time required for PES generation. However, because the SAPT dimer calculations may in general be run simultaneously, enabling this option will typically result in significantly reduced wall time. If autoPES is compiled with the OMP option enabled, the NCORES option is used to set the number of parallel threads (see Sec. 7).

In addition to the OpenMP option mentioned above, the PES generation procedure naturally allows for a large degree of parallel processing because the SAPT calculations for each grid point are independent. This is typically done through the computer’s job queuing system using the options MAX_SIM_PT and NPART_PER_JOB (see Sec. 7). However, it may be desirable on certain systems to run each individual SAPT calculation on multiple cores. This is accomplished through a combination of parallel BLAS libraries and Message Passing Interface (MPI), and requires both the SAPT package and front-end programs to be properly configured. See appropriate manuals for details. If the SAPT package and interface programs are properly configured, the NCORES_SCF option may be used to specify the number of cores to use in each SAPT dimer calculation. The NCORES_SCF option also affects supermolecular calculations, if the interface program used is properly configured.

4.7 Basis sets

The autoPES package makes use of the basis set repository included with the SAPT package, which includes the cc-pVDZ, cc-pVTZ, cc-pVQZ, aug-cc-pVDZ, aug-cc-pVTZ, aug-cc-pVQZ basis sets \cite{12} and others for the first, second, and third row atoms. More basis sets can be added by the user by placing an appropriately named file in the SAPT2020/basis sets directory, in the Gaussian94 format. The main basis set may be chosen using the BASIS option in the NAME.ctrl file as specified in Sec. 7.

Auxiliary basis sets of Weigend \cite{13} corresponding to the main basis sets are also included. As with the main basis sets, additional auxiliary basis sets may be added to the SAPT depository in the Gaussian94 format. By default, the auxiliary basis corresponding to the chosen main basis is used. A different auxiliary basis may be chosen using the AUXBASIS option.

Midbond functions are by default the set of spdf functions and corresponding auxiliary basis functions taken from Ref. \cite{14}.

The MC+BS form of the basis sets\cite{15} is used for SAPT(DFT) calculations. There is currently no option to use the full dimer centered basis set, except in the case of supermolecular calculations.
5 Potential Energy Surface

5.1 Rigid Monomer Case

When no IDOF are specified, the PES has the functional form (see Ref. [1], Sec. IV)

\[ V = V_{\text{ind}}(A, B) + \sum_{a \in A} \sum_{b \in B} u_{ab}(r_{ab}). \] (1)

The first term is a polarization model. The second term is a sum of site-site functions, where one site belongs to monomer A and the other to monomer B. The set of sites includes all atoms of both monomers, and may also include off-atomic sites if additional accuracy is required.

\[ u_{ab}(r_{ab}) = A_{12}^{ab} P_{ab}(r_{ab}) e^{-\alpha^{ab} r_{ab}} + f_1(\delta^{ab}, r_{ab}) \frac{q_a q_b}{r_{ab}} - \sum_{n=6,8,10} f_n(\delta^{ab}, r_{ab}) \frac{C_n^{ab}}{(r_{ab})^n}. \] (2)

Here \( r_{ab} \) is the distance between sites, \( q_x \) are partial charges on each site, \( C_n^{ab} \) are distributed induction plus dispersion coefficients, and \( f_n \) are Tang-Toennies damping functions \[16\]. The coefficients \( A_{12}^{ab} \) are constrained to be positive, and are included to ensure the correct repulsive behavior of the potential at very close range. The factor \( A \) is needed for consistency of energy units, and equal to 1 kcal/mol in the case that all other parameters have the units specified in Sec. 8.3. Any of the sites in the fit may optionally exclude electrostatic, polarization, dispersion, or exponential components. In the case that one or both monomers has nonzero charge, the last summation in Eq. 1 starts with \( n = 4 \) rather than \( n = 6 \) in order to reproduce the correct asymptotic behavior of the induction energy.

The function \( P_{ab}(r) \) is a polynomial in \( r \) given by

\[ P_{ab}(r) = 1 + \sum_{i=1}^{k} a_{i}^{ab} r^i, \] (3)

where \( a_{i}^{ab} \) are free parameters of the fit.

The polarization model defining \( V_{\text{ind}} \) consists of a set of induced point-dipoles \( \mu_{a}^{\text{ind}} \) on each site \( a \in A \) (and analogously for \( b \in B \)), defined as

\[ \mu_{a}^{\text{ind}} = \alpha_a \left[ \mathcal{E}_a + \sum_{b \in B} T_{ab} \mu_{b}^{\text{ind}} \right], \] (4)

where \( \alpha_a \) is an isotropic polarizability assigned to a site \( a \), \( \mathcal{E}_a \) is the damped electric field at point \( a \) due to all permanent point charges \( q_b \) of monomer B

\[ \mathcal{E}_a = \sum_{b \in B} f_1(\delta^{ab}, r_{ab}) \frac{q_b r_{ab}}{r_{ab}^3}, \] (5)

and \( T_{ab} \) is the damped dipole-dipole 3×3 interaction tensor

\[ T_{ab} = f_3(\delta^{ab}, r_{ab}) \left[ 3 \frac{r_{ab} \otimes r_{ab}}{r_{ab}^5} - \frac{1}{r_{ab}^3} \right]. \] (6)
with \( \mathbf{r}_{ab} \) denoting the displacement vector from site \( a \) to site \( b \) and \( \otimes \) denoting the Kronecker product. The summations in Eqs. (4) and (5) are over atoms in monomers other than the one in which the atom \( c \) resides. After the set of equations is solved iteratively, the energy contribution from the polarization model is given by

\[
V_{\text{ind}} = -\frac{1}{2} \sum_{c} \mathbf{E}_{c} \cdot \mathbf{\mu}_{c}^{\text{ind}},
\]

where the sum is over all atoms \( c \) in both monomers.

The two-body implementation of the polarization model can easily be extended to more bodies by making the summations in Eqs. 5 and 6 over atoms in monomers other than the one in which the atom \( c \) resides and summing Eq. 7 over atoms in all monomers.

### 5.2 Flexible Monomer Case

In the flexible-monomer case, the terms in Eq. 2 are generalized. The polynomial \( P_{ab}(\mathbf{r}) \) is generalized to \( P_{ab}(\mathbf{r}, (r_{ab_1}), (r_{a_1b_2}), (r_{a_2b_2})) \), where the tuple \( (r_{ab_1}) \) denotes the site-site distances, including off-atomic sites, between site \( a \) and any site \( b_1 \) in monomer \( B \), and similarly for \( (r_{a_1b_2}) \). The tuples \( (r_{a_1a_2}) \) and \( (r_{b_1b_2}) \) denote all the site-site distances in the respective monomer.

\[
P_{ab}(\mathbf{r}, (r_{ab_1}), (r_{a_1b_2}), (r_{a_2b_2})) = P_{ab}(\mathbf{r})
+ \sum_{b_2 \neq b} P_{k',ab_2}^{ab}(r_{ab_2}) + \sum_{a_2 \neq a} P_{k',a_2b}^{ab}(r_{a_2b})
+ \sum_{a_1a_2 \in A_2} P_{k',a_1a_2}^{ab}(r_{a_1a_2}) + \sum_{b_1b_2 \in B_2} P_{k',b_1b_2}^{ab}(r_{b_1b_2})
\]

where \( A_2 \) and \( B_2 \) refer to the sets of all unique pairs of sites in the respective monomers and the polynomials are defined as

\[
P_{k',ab_2}^{ab}(r_{ab_2}) = \sum_{i=1}^{k'} b_{i,ab_2}^{ab} \mathbf{r}_{ab_2}
\]

and analogously for \( P_{k',a_1a_2}^{ab}(r_{a_1a_2}) \), and

\[
P_{k',a_1a_2}^{ab}(r_{a_1a_2}) = \sum_{i=1}^{k'} b_{i,a_1a_2}^{ab} [r_{a_1a_2} - \bar{r}_{a_1a_2}]^i
\]

where \( \bar{r}_{a_1a_2} \) is defined in the same way as in Eq. 14, and analogously for \( P_{k',b_1b_2}^{ab}(r_{b_1b_2}) \). The distance \( \bar{r}_{a_1a_2} \) denotes the arithmetic mean of all site-site distances \( r_{aa'} \) in the reference monomer, where \( aa' \) is the same site-pair type as \( a_1a_2 \).

The generalized partial charges are given by

\[
q_a((r_{a_1a_2})) = q_a^0 + \sum_{(a_1,a_2) \in A^2} \sum_{i=1}^{k_a} q_i^{a_1a_2} [r_{a_1a_2} - \bar{r}_{a_1a_2}]^i,
\]
and similarly for $q_b$. The generalized induction plus dispersion coefficients $C_{ab}^n$ always make use of the combination rule $C_{ab}^n = \sqrt{C_{an}^a C_{bn}^b}$, where
\[
C_{ab}^n(r_{a_1 a_2}) = C_{ab}^0 + \sum_{(a_1, a_2) \in A^2} C_{a_1 a_2}^n (r_{a_1 a_2} - \bar{r}_{a_1 a_2}),
\]
and similarly for $C_{bn}^b$.

5.3 Intramonomer PES

The intramonomer components $V_A(R^A)$ are fitted using the same feature space of atom-atom distances, restricted to those atoms within monomer A.
\[
V_A(R^A) = P_0 + \sum_{a_1 \in A} \sum_{a_2 \in A} P_{i}^{a_1 a_2} (r_{a_1 a_2} - \bar{r}_{a_1 a_2}) + \sum_{(a_1, a_2, a_3, a_4) \in A^4} P_{l}^{a_1 a_2, a_3 a_4} (r_{a_1 a_2} - \bar{r}_{a_1 a_2}, r_{a_3 a_4} - \bar{r}_{a_3 a_4})
\]

where the symbol $A^4$ denotes the set of all atom-pair-pairs in the monomer. The constant $P_0$ is treated as an additional fitting parameter, and the polynomials $P$ are defined as
\[
P_{i}^{a_1 a_2} (r_{a_1 a_2} - \bar{r}_{a_1 a_2}) = \sum_{i=1}^{l} c_{i}^{a_1 a_2} [r_{a_1 a_2} - \bar{r}_{a_1 a_2}]^i
\]
and
\[
P_{l}^{a_1 a_2, a_3 a_4} (r_{a_1 a_2} - \bar{r}_{a_1 a_2}, r_{a_3 a_4} - \bar{r}_{a_3 a_4}) = \sum_{i=1}^{l'} \sum_{j=1}^{l'} c_{ij}^{a_1 a_2, a_3 a_4} [r_{a_1 a_2} - \bar{r}_{a_1 a_2}]^i [r_{a_3 a_4} - \bar{r}_{a_3 a_4}]^j.
\]

6 Graphical Energy Plotting Utility

The autoPES package includes the “Dimerplot” graphical utility for plotting energy components of the intermolecular PES. It is written in Java and is distributed as both a precompiled JAR file and source code. The Java program calls the external Fortran program dimer_plot to evaluate the PES. Currently, Dimerplot only supports rigid-monomer PESs, and when used with flexible PESs, will only plot configurations in which the monomers are fixed in their reference geometries.

Fig. 1 shows a screenshot of Dimerplot for the water dimer near the minimum configuration. The Euler angles of monomers A and B are fixed at user supplied values, and the COM of monomer A is fixed at the origin. The interaction energy is plotted with respect to translations of the COM of monomer B in a fixed plane, with the vertical axis corresponding to the Z direction. Note that the fit breaks down at extremely close monomer separation.

To use Dimerplot, Java version 8 or newer must be installed, and a graphical environment must be enabled. Navigate to the dimerplot subdirectory, and run the command.
Figure 1: A plot of interaction energy for the water dimer near the minimum configuration made using the Dimerplot utility.

```
java -jar dimerplot.jar
```

A dialog option will open, and the directory containing the `fit_inter.dat` and `NAME.input_long` files should be selected.

To run Dimerplot from an autoPES installation on a remote Linux machine, an X Window environment must be enabled on the remote machine, and the connection must support X11 forwarding. If the local machine is running Linux, this can be done by logging into the remote machine via “ssh -X”. If using PuTTY from a Windows machine, the option “Enable X11 Forwarding” must be enabled in PuTTY. Additionally, if running remotely from a Windows machine, an X Window server such as Xming must be installed and running on Windows.

7 Options

The autoPES package allows user to control the following settings via the `NAME.ctrl` file, which should be prepared in the FORTRAN namelist format (see Sec. 4.1).

1. Namelist: GENERAL

   • MIN_GRIDGEN_ITER: integer – This is the minimum number of iterations (excluding hole-fixing iterations) to be performed beyond the initial one, regardless of whether or not the convergence criteria is met. It is recommended to use at least one iteration to ensure accuracy around the minima of the PES. Default value is 1.
   
   • TEST_PCT: integer – Percentage of the grid points to be used as a test set for evaluating
the quality of the potential. Once the convergence criterion is met, all of the grid points are used for the final fit. Default value is 30.

- **TEST\_CONV**: real – The maximum difference in error between fit data and test data to use in PES convergence criteria. The PES will be considered converged, and iterations will no longer be performed (excluding hole-fixing iterations), when \((\text{testRMSE}) < (1 + \text{TEST\_CONV}) \ast (\text{fitRMSE})\). Default value is 0.2.

2. Namelist: **AB\_INITIO**

- **METHOD**: string – The name of the \textit{ab initio} method to use. Currently supported values are ‘SAPTDFT’, indicating SAPT(DFT) calculation, and ‘CCSDT’, indicating counterpoise-corrected supermolecular CCSD(T) calculation. Default value is ‘SAPTDFT’.

- **INTERFACE\_PROGRAM**: string – Interface program to use for supermolecular calculations. Currently supported interfaces are ‘ORCA’ and ‘MOLPRO’. Default value is ‘ORCA’.

- **BASIS**: string – Name of the basis set which will be used for all \textit{ab initio} calculations. The program does not allow to use different basis sets in different parts of the computations, hence the ionization potential calculations, the asymptotic calculations and the main \textit{ab initio} calculations use the same basis set. Midbond functions are enabled by default and may be disabled using the MB\_CUTOFF option. The aug-cc-pVDZ and aug-cc-pVTZ basis sets are distributed with autoPES, and others can be added, see Sec. 4.7. Default value is ‘aug-cc-pVDZ’.

- **AUXBASIS**: string – The name of the auxiliary basis used for SAPT density fitting calculations. Default value is BASIS-RI, where BASIS is the value of the above option. See Sec. 4.7 for more details.

- **ASYM\_BASIS**: string – The name of the basis set used for the asymptotic expansion. Default value is the same as BASIS.

- **ASYM\_AUXBASIS**: string – The name of the auxiliary basis used for the distributed asymptotic expansion. If the COM-COM expansion is used, this is unused. Default value is ASYM\_BASIS-RI, where ASYM\_BASIS is the value of the above option.

- **CBS\_TZ\_BASIS**: If triple-quadruple CBS extrapolation is being used, the BASIS variable should be set to a QZ-quality basis, and this option should be set to the corresponding TZ-quality basis. If no CBS extrapolation is desired, this should be set to ‘NO’. Default value is ‘NO’.
• HYBRID: logical – Specifies whether the hybrid PBE0 functional (T) or non-hybrid PBE functional (F) is used for SAPT(DFT) calculations. Default value is false.

• DELTAHF: string/logical – Specifies whether or not to include the $\delta E_{\text{int,resp}}^{\text{HF}}$ correction for SAPT(DFT) runs. This correction typically increases the computer time required for each grid point by about 60%, and is important for highly polar systems such as water. See the SAPT manual for details. Valid values of DELTAHF are ‘T’, ‘F’, or ‘AUTO’. The AUTO setting determines whether to include $\delta E_{\text{int,resp}}^{\text{HF}}$ based on the criteria described in Ref. [1]. Default value is AUTO.

• MB_CUTOFF: real – For any grid point such that midbond functions placed using the $1/r^6$ weighting method results in a midbond center which is too close to an atom, the midbond functions are removed. This is to prevent divergence of SAPT at close distance. The midbond functions are removed if the distance between any atom and the midbond center is less than MB_CUTOFF times the covalent radius of that atom. This option may also be used to remove midbond functions entirely by setting it to a large value, e.g. 1.0E10. Default value is 0.8.

• FROZEN_CORE: logical – Sets whether frozen core approximation should be used for supermolecular calculations. Default value is false.

3. Namelist: GRID_GENERATION

• FORCE_FIELD: string of characters, maximal length 20 – name of the molecular mechanics force field which will be used as the initial guiding function in the grid generation process. The default value of this variable is oplsaa, which refers to the all atoms OPLS force field [17]. This is the only force field which is included with the autoPES package installation. However, users can add other force fields as required. Such force fields should be placed in the ff subdirectory of the autoPES installation directory, and the FORCE_FIELD option should correspond to the file name.

• GRID_SIZE: integer – the size of the initial grid for the ab initio calculations. Additional points will be added during the iteration stages. If set to 0, the initial grid size will be set to $\text{GRID\_SIZE} = N_{\text{FP}} \ast 6 \ast 100/(100 - \text{TEST\_PCT})$, where $N_{\text{FP}}$ is the number of free parameters in the main PES fitting stage. Default value is 0.

• GRID_ADD_PCT: integer – The number of points to add in each iteration (excluding hole-fixing iterations), expressed as a percentage of GRID_SIZE. Default value is 20.

• GRID_WEIGHT_SCALE: real – Specifies the strength of the exponential weighting used for the intramolecular energy component of the guiding PES in the generation of grid points. A smaller value will favor sampling of lower energy configurations more, while
a larger value will give more uniform sampling. See Eq. 4 of Ref. \[1\]. Default value is 2.0.

- GRID_WEIGHT_SCALE_INTRA: real – Specifies the strength of the exponential weighting used for the intramolecular energy component of the guiding PES in the generation of grid points. See Sec. IIB in Ref. \[\?\]. If a value of 0 is given, it will be set to $2^*\text{GRID\_WEIGHT\_SCALE}$. Default value is 0.

- GRID_MINIMA_PCT: integer – This percentage of points in the initial grid will be generated around the minima of the guiding potential, with the remainder generated throughout the configuration space using a Monte Carlo method. Default value is 10 (meaning 10%).

- GRID_MINIMA_PCT_ADD: integer – Same as GRID_MINIMA_PCT, but applying to second and later iterations rather than to the initial grid. It is recommended that this value is set higher than GRID_MINIMA_PCT to ensure accuracy around the true minima of the PES. Default value is 20 (meaning 20%).

- GRID_MODE: string – Which algorithm to use for the main grid generation. Available values are ‘metric’, ‘sobol’, and ‘rand’. The value ‘metric’ indicates that the $\epsilon_{mn}$ metric will be used to reject grid points which are similar to existing ones, as described in Ref. \[1\]. Default value is ‘metric’.

- FRAC_R0: real – If IDOFs are used, this specifies what fraction of the grid points use the reference monomer geometries. The remaining grid points will include deformations. Default value is 0.3.

- IVMG: logical – Whether to use the Iterative Variance Minimizing Grid (IVMG) method for grid iterations after the first (other than hole-fixing iterations).

4. Namelist: ASYMPTOTICS

- MAX_CN: integer – the highest power used in the distributed asymptotic induction plus dispersion expansion. The minimum value is set automatically to 6 in the case that neither monomer is charged, and 4 in the case that one or both monomers are charged. Available values are 6, 8 and 10. Default value is 6.

- COM_RULE_CN: integer – the type of combination rules to use for the distributed induction plus dispersion parameters. A value of 0 indicates a separate free parameter for each site-pair type, and a value of 1 indicates a single free parameter for each site type, with values between site par types determined using a geometric combination rule.

- POLARIZABLE: logical – this variable indicates whether the polarizable model should be used. Default value is F.
• **DISTR**: logical – Specifies whether COM (F) or distributed (T) asymptotic expansion should be used. Default value is F.

• **ELS_PEN_STR**: real – Determines the strength of the electrostatic fitting penalty function. A value of 0 indicates no penalty function. Default value is 1.0.

• **POLARIZABILITY_A**: real – Specifies the total polarizability of monomer A in atomic units. If a value of 0.0 is given, then no constraint is used for the fitting of atomic polarizabilities to asymptotic induction energies. If the polarization model is not used, this option has no effect. Default value is 0.0.

• **POLARIZABILITY_B**: real – Similar to **POLARIZABILITY_A**, but for monomer B. Default value is 0.0.

• **POLY_ORDER_ELS**: real – Order of the polynomial expansion of the partial charges in terms of intramolecular distances. Unused in rigid-monomer case. Possible values are 0, 1, or 2. Default value is 2.

• **POLY_ORDER_CN**: real – Order of the polynomial expansion of the induction+dispersion coefficients in terms of intramolecular distances. Unused in rigid-monomer case. Possible values are 0 or 1. Default value is 1.

5. Namelist: MAIN_FIT

• **POLY_ORDER**: integer – Order of the linear polynomial factor in the exponential term of the PES functional form. This choice reflects a trade-off between the accuracy of the fit and the number of grid points required. It is always recommended to use at least a first order polynomial if possible. A second order polynomial usually improves accuracy significantly over a first order one. A third order polynomial can sometimes be useful for small systems, but usually does not have a large effect. The use of polynomial order greater than 3 is generally not beneficial. Valid range is 0 to 5, with default value of 2.

• **POLY_ORDER_MIXED_INTRA**: integer – Determines the value of the summation limit $k'$ in Eq. [9]. This has no effect unless IDOFs are used. Default value is 0.

• **POLY_ORDER_MIXED_INTER**: integer – Determines the value of the summation limit $k''$ in Eq. [10]. Default value is 0.

• **FIT_WEIGHT_SCALE**: real – Value used in setting of fitting weights. The weight of a point with interaction energy $E_i$ is set to $w_i = \exp(-\text{FIT_WEIGHT_SCALE} \times W)$ if $W < 1$ and $w_i = \exp(-\text{FIT_WEIGHT_SCALE})/W^3$ when $W > 1$, where $W = (E_i - E_{\text{min}})/(|E_{\text{min}}| + 4\text{kcal/mol})$ and $E_{\text{min}}$ is the global minimum energy of the PES. A smaller value results in stronger weighting of more attractive regions, while a larger value gives more even weighting. Default value is 5.0.
• **COM\_RULE\_EXP**: integer – Indicates the type of combination rule to use for the exponential $\alpha_{ab}$ and $\beta_{ab}$ fit parameters. A value of 0 indicates a separate value for each site-pair, and 1 indicates a single value per site, with values between pairs obtained by an arithmetic combination rule. For dimers larger than about 10 atoms, the use of separate parameters for each site-pair typically does not yield a significant increase in accuracy in comparison to the increased number of grid points needed to fit the extra free parameters. However, for smaller dimers, especially if off-atomic sites are used, it may be beneficial to set this option to 0. Default value is 1.

• **COM\_RULE\_LIN**: integer – Indicates the type of combination rule to use for the linear $a$ fit parameters. A value of 0 indicates a separate value for each site-pair, and 1 indicates a single value per site, with values between pairs obtained by an arithmetic combination rule. Default value is 0.

• **COM\_RULE\_DAMP**: integer – Indicates the type of combination rule to use for damping parameters in the distributed electrostatics and induction plus dispersion expansions. A value of 0 indicates a separate value for each site-pair, 1 indicates a single value per site with values between pairs obtained by a geometric combination rule, and 2 indicates a single value for all damping parameters. Default value is 1.

• **ELS\_DMP**: logical – Sets whether or not Tang-Toennies damping will be used for the electrostatic component of the PES. Default value is F (meaning damping is not used).

• **CN\_DMP**: logical – Sets whether or not Tang-Toennies damping will be used for the induction plus dispersion component of the PES. Default value is T (meaning damping is used).

• **FIT\_A12**: logical – Sets whether to use the repulsive $r^{-12}$ term of Eq. (1). Including this term may reduce the number of grid points required for hole fixing. Default value is F.

• **FIT\_AUTO\_REDUCE**: real – The functional form of the fit will be automatically simplified such that the number of fitting points in each stage of the fitting is no less than \text{FIT\_AUTO\_REDUCE} times the number of free parameters in that stage. Simplifications are done by introducing combination rules or reducing the polynomial order. This is enabled by default because if there are not enough data points to fit the full functional form, the resulting potential may be of very poor quality. Note that this feature will only have an effect if the user changes the functional form of the fit to include a larger number of free parameters after the number of grid points is determined, or if the user manually sets the number of grid points to a value which is too small given the functional form. Default value is 2.0.
6. Namelist: HOLE_FIXING

- **WALL_HEIGHT**: real – The minimum required height of the repulsive wall for any orientation, in kcal/mol. See Sec. VII of Ref. [1]. Default value is 15.0.

- **WALL_MIN_DIST**: string/real – Specifies the minimum distance at which hole fixing points are added. Given as a ratio of the distance between the nearest pair of atoms in a given configuration to the sum of the covalent radii of those atoms. A typical value is 1.3, or smaller for strongly attracting systems. If set to ‘AUTO’, the distance will be automatically determined based on energies of previously computed grid points (see Sec. VII of Ref. [1]). Default value is AUTO.

- **HOLE_ADD_PCT**: integer – The maximum number of points to add in each hole-fixing iteration, expressed as a percentage of the initial grid size. Setting this to a higher value may decrease the number of hole-fixing iterations, but may also result in less efficient use of computer resources. Default value is 10 (meaning 10%).

7. Namelist: SYS_SETTINGS

- **NCORES**: integer – The number of parallel cores to be used by all components of autoPES. The autoPES programs which support OpenMP parallelization do not account for a large portion of the overall CPU time of the PES generation, but can significantly increase overall wall time if only one core is used here. Therefore the use of about 4 to 8 cores is recommended if available. Default is 1.

- **NCORES_SCF**: integer – The number of parallel cores to be used in SAPT or supermolecular calculations. Given a properly configured installation, SAPT calculations can be expected to scale reasonably well up to about 8 cores. Default value is 1.

- **NPART_PER_JOB**: integer – The number of SAPT points to calculate in parallel in each job submitted to the queuing system. This option is intended for computer systems which require an entire node to be reserved for each job. If this constraint is not present, this should be left at the default value of 1.

- **MAX_SIM_PT**: integer – The maximum number of SAPT calculations that may be submitted to the queuing system at the same time. If the autoPES program is left running (see Sec. 4.3), it will automatically submit additional grid points as others finish to maintain this many jobs in the queue. On many systems, MAX_SIM_PT may be set to a large value so that autoPES submits all points simultaneously, and the queuing system is left to run jobs appropriately. Default value is 1.

- **SAPT_MEM**: integer – Each SAPT calculation will be limited to this many million 8-byte words of virtual memory. A value of 0 indicates that SAPT will use the amount
of memory required for optimal performance. SAPT can run with less memory with some performance penalty, and this option may be used to reduce the required memory. To determine the minimum amount of memory that SAPT needs to run, the SAPT memory estimator may be used (see the SAPT package manual for details). If CCSD(T) calculations are used in place of SAPT, this value is used to set the ‘maxcore’ option in ORCA (the value in this case is still given in millions of words, and is converted by the program to units appropriate for ORCA). Default value is 0.

- **SAPT_TIMEOUT:** integer – If a grid point begins computing and does not finish after this number of minutes, it will be assumed to have failed and will be discarded. Zero means no timeout. Default value is 0.
- **SAPT_QUEUE_TIMEOUT:** integer – If a grid point calculation is submitted to the queue and does not begin running after this number of minutes, it will be assumed to have failed and will be discarded. Zero means no timeout. Default value is 0.

8. Namelist: INTRA_FIT

- **INC_INTRA:** logical – Whether or not to include the intramolecular PES for flexible-monomer fits. Default value of true.
- **POLY_ORDER_INTRA_BN:** integer – Determines the value of the summation limit \( l \) in Eq. [14]. This has no effect unless IDOFs are used. Default value is 4.
- **POLY_ORDER_INTRA_BMN:** integer – Determines the value of the summation limit \( l' \) in Eq. [15]. This has no effect unless IDOFs are used. Default value is 2.
- **GRID_SIZE_INTRA:** integer – The size of the grid for the intramolecular \textit{ab initio} calculations. If set to 0, the initial grid size will be set to \( \text{GRID\_SIZE\_INTRA} = N_{\text{FP}}^{\text{intra}} \times 20 \), where \( N_{\text{FP}}^{\text{intra}} \) is the number of free parameters in the intramolecular PES. Default value is 0.
- **GEO_IDOF_THRESH:** real – This parameter controls the limits that IDOFs are able to be deformed for the intramonomer fit. See the variable \( \delta_{\text{max}} \) in Sec.IIA of Ref. [?]. If IDOF limits are explicitly specified, this is ignored. Default value is 0.5.

8 Structure of the program

8.1 Program flow

The autoPES package has a modular structure in which different subprograms are called and coordinated by the main script \texttt{master.sh}. The master script is itself called by the front-end
program called autoPES, whose only job is to initialize default values of variables and read in user’s settings. When launched, the master script will identify the appropriate next step of PES generation based on the presence of files and directories within the main running directory. All computationally expensive tasks are associated with their own directory and are handled by a corresponding slave script. This procedure is outlined below.

1. Monomer property calculations – The IP (for ionization potential) directory is created and the slave_ip.sh script is called, which performs the following steps:

   - Input files for \textit{ab initio} calculations are created with the make_ip.inp program.
   - Jobs corresponding to neutral monomer A and ionized monomer A are submitted. In the case of different monomers, two jobs are also submitted for monomer B. The script then waits for all jobs to finish.
   - Ionization potentials, dipole moments, and polarizabilities are extracted from the \textit{ab initio} outputs.

2. Intramonomer \textit{ab initio} calculations – In the case of flexible-monomer PESs, the INTRA directory is created and the slave_sapt.sh script is called, which performs the following steps:

   - Inputs are created for \textit{ab initio} codes. For each point \(i\) in the grid, a subdirectory INTRA/PART\(i\) is created.
   - The script begins creating, submitting, and awaiting the completion of jobs until all \textit{ab initio} calculations are complete. For each job \(j\) that is submitted, a subdirectory INTRA/JOB\(j\) is created, which contains soft links to each of the PART\(i\) directories corresponding to grid points packaged within that job. The number of grid points per job is selected using the option NPART\_PER\_JOB. For each job \(j\) which completes, the file INTRA/JOB\(j\)/FINALIZED is created. For each grid point \(i\) which is packaged into a job, the file INTRA/PART\(i\)/SUBMITTED is created. When the grid point \(i\) calculation completes, one of the files INTRA/PART\(i\)/SUCCESS or INTRA/PART\(i\)/FAILED is created. A job may fail due to bad output or timeout, as described in Sec. 7. Outputs of \textit{ab initio} calculations are stored in the INTRA/PART\(i\)/NAME.log files.
   - When all parts have either succeeded or failed, the results are combined into the file INTRA/NAME.ener.

3. Intramonomer PES fitting – In the case of flexible-monomer PESs, the INTRA\_FIT directory is created and the slave_fit_intra.sh script is called. A job is submitted to run the intramonomer fitting program fit\_inter. The script waits for fitting to complete.
4. Asymptotic calculation – The ASYM directory is created and either the slave_asym.sh script or the slave_asym_idof.sh script is called, which performs the following steps:

- Inputs are generated for the SAPT asymptotic codes and a job is submitted to run the asymptotic calculations using the Dalton 2.0 interface (see the SAPT manual [4] for details). This results in a grid of long-range interaction energies, whose energy components are stored in the file NAME.ener_asym.
- Inputs are generated for the asymptotic fitting codes, and a job is submitted to fit the long-range PES parameters to the grid of energies using the program fit_asymp (see Ref. [1], Sec. V for details).

5. Grid generation – The GRIDn directory is created, where n is the iteration number, and the slave_gridgen.sh script is called, which performs the following steps:

- A job is submitted to run the script find_minima.sh, which locates the local minima of the PES, and generates grid points around the local minima, stored in the file NAME.geo_min (see Ref. [1], Sec. IIB for details).
- A job is submitted to compute the main grid using the gridgen_ff program, stored in the file NAME.geo_main (see Ref. [1], Sec. IIA for details).
- The script waits for both jobs to complete and the results are combined into a single grid.

6. \textit{ab initio} calculations – The SAPTn directory is created and the slave_sapt.sh script is called, which performs the following steps:

- Inputs are created for the \textit{ab initio} codes. For each point $i$ in the grid, a subdirectory SAPTn/PART$i$ is created.
- The script begins creating, submitting, and awaiting the completion of jobs until all \textit{ab initio} calculations are complete. For each job $j$ that is submitted, a subdirectory SAPTn/JOB$j$ is created, which contains soft links to each of the PART$i$ directories corresponding to grid points packaged within that job. The number of grid points per job is selected using the option NPART_PER_JOB. For each job $j$ which completes, the file SAPTn/JOB$j$/FINALIZED is created. For each grid point $i$ which is packaged into a job, the file SAPTn/PART$i$/SUBMITTED is created. When the grid point $i$ calculation completes, one of the files SAPTn/PART$i$/SUCCESS or SAPTn/PART$i$/FAILED is created. A job may fail due to bad output or timeout, as described in Sec. 7. Outputs of \textit{ab initio} calculations are stored in the SAPTn/PART$i$/NAME.log files.
When all parts have either succeeded or failed, the results are combined into the file 
SAPTn/NAME.ener.

7. Delta-HF calculations – If delta-HF calculations are to be performed, the DELTAHF
n directory is created and the slave_sapt.sh script is called, which performs the following steps:

• Inputs are created for the SAPT delta HF codes, and the calculations are performed as 
with the SAPT calculations described above.
• The program combine_ener is used to combine the results of the main SAPT calculations 
and delta-HF calculations.

8. Main PES fitting – The FITn directory is created and the slave_fit.sh script is called, 
which performs the following steps:

• The input grid is randomly split into fitting and testing components (unless the 
TEST_PCT option is set to 0 or if the final fitting iteration has been reached).
• A job is submitted to run the main fitting program fit_inter (see Ref. [1], Sec. VI for 
details). The script waits for fitting to complete.
• A job to execute the hole scanning program hole_scan is submitted (see Ref. [1], Sec. VII 
for details).
• A job is submitted to run the script find_minima.sh, which locates the local minima 
of the PES.
• The script waits for hole scanning and minima localization to complete, then runs the 
program gen_report to create a human readable report.

9. Additional iterations – Based on the result of the above fitting step, additional iterations 
may be performed:

• If the hole scanning procedure identified any holes, then the next iteration begins with 
the SAPT calculation steps and hole-fixing points are computed.
• If there were no holes found but the accuracy criteria were not met by the previous 
fit, the next iteration begins with the grid generation step and more grid points are 
computed.
• If the accuracy criteria were met and no holes were found, a final fitting stage is begun 
in which there is no test set, i.e., the fit is performed on the entire set of grid points.
• If the final fitting stage is complete and no holes were found, the master script termi-
nates.
After the completion of each step, the autoPES master script identifies the next action to be performed based on the subdirectories present in the main running directory, and the files \texttt{SLAVE\_X\_STARTED} and \texttt{SLAVE\_X\_DONE} in those subdirectories. It is always possible to manually remove a subdirectory when autoPES is not running, and autoPES will re-run the necessary calculations the next time it is launched. One may also, for example, manually create a directory \texttt{FITn} (where \texttt{n} is any positive integer less than 1000 which is not yet used) after altering the \texttt{NAME}\_ctrl file in order to perform an additional fit with a modified functional form.

### 8.2 Directory structure

The autoPES package has the following directory structure when extracted

1. \texttt{bin} – directory containing all scripts and compiled binaries
2. \texttt{dimerplot} – directory containing Java source code and precompiled JAR for the \texttt{dimerplot} utility
3. \texttt{doc} – directory containing documentation
4. \texttt{example} – directory containing example runs
5. \texttt{ff} – directory containing guiding force field parameters
6. \texttt{src} – directory containing all source files for FORTRAN programs

### 8.3 Internal file formats

The most important internal files created by autoPES are described below.

- \texttt{fit\_asymp.dat} – The \texttt{fit\_asymp.dat} file is created by the asymptotics fitting program \texttt{fit\_asymp}, and contains parameter values of the asymptotic part of the fit. It is read in by the complete PES fitting program \texttt{fit\_inter}.

An example \texttt{fit\_asymp.dat} file for the rigid-monomer water dimer is given below. In this example, the distributed induction plus dispersion expansion includes $1/r^6$, $1/r^8$, and $1/r^{10}$ terms, and no polarization model. There are three off-atomic site types, with only the first including the electrostatic interaction, and all three excluding the induction plus dispersion interaction.

```
1  8  -0.343214985865E+00  0.000000000000E+00
2  1  -0.573834190669E+00  0.000000000000E+00
3  0  0.745441683602E+00  0.000000000000E+00
4  0  0.000000000000E+00  0.000000000000E+00
```
The first section of the file contains one line per site type, with three items per line. The first item is an integer giving the atomic number of the site type, with 0 used for off-atomic sites. The next two items are the partial charge $q_i$ and polarizability $\alpha_i$ of the site type in atomic units.

The next line contains three integers. The first specifies the number of terms in the induction plus dispersion expansion. The second is the combination rule used for the induction plus dispersion expansion, with 1 indicating a geometric combination rule and 0 indicating no combination rule. The third gives the starting power of the induction plus dispersion expansion.

The next section contains one line per site type in the case that the combination rule is used for the induction plus dispersion expansion, and one line per site-pair type in the case that no combination rule is used. Each line contains the $C_n$ values ($n = 6, 8, 10$ in this example) in units of kcal/mol × Å$^n$.

- The final two sections give the values of the parameters $C_{n,a,a_1}^a$, $C_{n,b,b_2}^b$, $q_{i,a_1}^a$, and $q_{i,b_1}^b$ of Eqs. (12) and (11). In this case there are no IDOFs, and so the zeros indicate that there are no following lines in the section.

- **fit_inter.dat** – The *fit_inter.dat* file is created by the intermolecular fitting program *fit_inter* after each fitting iteration, and contains all parameter values for the intermolecular fit.

An example *fit_inter.dat* file for the flexible water dimer is given below.
Line 1 contains five integers, specifying the quantities:

* The linear polynomial order $k$, i.e. POLY_ORDER
* The number of terms per site-pair in the distributed induction plus dispersion expansion
* The first power of the distributed induction plus dispersion expansion. In cases where there is more than one term in the induction plus dispersion expansion, powers increase by two starting from the first power.
* The value of $k'$, i.e. POLY_ORDER_MIXED_INTRA
* The value of $k''$, i.e. POLY_ORDER_MIXED_INTER

Line 2 contains two integers, specifying the number of sites in monomer A and in monomer B.

Line 3 specifies site type indices $s_i$ of each site $i$ of monomer A (as given in the NAME.input file). The next line is an analogous one for monomer B.

Line 5 specifies the number of optimized off-atomic site groups. In the current version of autoPES, it should always equal 0.

Lines 6 and 7 specify partial charges and isotropic polarizabilities of each site type, with one line per site type. Both values are in atomic units.

Line 8 contains one integer specifying the number of site-pair types. This corresponds to the number of lines in the next section of the file.

Lines 9 through 11 contain the site-pair parameters, with each line corresponding to one site-pair type. The columns are as follows, with parameters defined in Sec. 5 (note that each two consecutive lines here form one line in the file)

* site type $a$
* site type $b$
* $\beta_{ab}$ (Å$^{-1}$)
* $\alpha_{ab}$ (unitless)
* $\alpha_{ab}^i (\text{Å}^{-i})$: one column for each term in the linear polynomial, $i = 1, \ldots, k$
* $C_n^b$ (kcal/mol $\times$ Å$^n$): one column for each term in the distributed induction plus dispersion expansion
* $A_{12}^{ab}$ (kcal/mol $\times$ Å$^{12}$)
* $\delta_{ab}^1 (\text{Å}^{-1})$
* $\delta_n^{ab} (\text{Å}^{-1})$: one column for each term in the distributed induction plus dispersion expansion
* $\delta_p^{ab} (\text{Å}^{-1})$

Note that the columns are included even if the chosen functional form does not use those parameters. In the case of unused damping parameters, a large value is given, which indicates that the damping factor is equal to unity.

– Line 12 gives the number of lines in the following section, with lines 13 through 14 specifying the values of $r_1(a_1a_2)$ for each site pair type.

– Line 15 given the number of lines in the following section, with lines 16 through 24 specifying the values of the mixed-distance intermolecular polynomial coefficients. Each of these lines contains $4 + k' + k''$ values:
  * Columns 1 and 2 give the site-types of the first site-pair-type.
  * Columns 3 and 4 give the site-types of the second site-pair-type.
  * Columns 5 through $5 + k' - 1$ give the values of the coefficients $b_{i,ab,ab}^2$ (in this example there is only one such value)
  * Columns $5 + k'$ through $6 + k'' - 1$ give the values of the coefficients $b_{i,ab,a_1a_2}$ (in this example there is only one such value)

– Line 25 contains two integers: the first gives the order of the flexible electrostatics expansion POLY_ORDER ELS, i.e. $k_q$. The second gives the number of lines in the following section, with lines 26 through 43 specifying the values of the flexible electrostatics expansion coefficients. Each of these lines contains $4 + k_q$ values:
  * Column 1 gives the monomer of the site $a$.
  * Column 2 gives the index of the site $a$.
  * Columns 3 and 4 give the site indices of the site-pair-type $a_1a_2$.
  * Columns 5 through $5 + k_{eq} - 1$ give the values of the coefficients $q_{i}^{a,a_1a_2}$ (in this example there are two such value)

– Line 44 gives the number of lines in the following section, with lines 45 through 62 specifying the values of the flexible induction+dispersion expansion coefficients. Each of these lines contains 4+MAX_CN values:
• Column 1 gives the monomer of the site $a$.
• Column 2 gives the index of the site $a$.
• Columns 3 and 4 give the site indices of the site-pair $a_1a_2$.
• Columns 5 through 6 give the values of the coefficients $C_{ni}^{a_1a_2}$.

**fit_intra.dat** – The *fit_intra.dat* file is created by the intermolecular fitting program *fit_intra*, and contains all parameter values for the intramolecular fit.

An example *fit_intra.dat* file for the flexible water dimer is given below.

```
1 3 1
2 -0.767925195594298E+02
3 2
4 1 2 1.525077812351310
5 2 2 2.551464654466650
6 2
7 1 2 -0.379151253118782E+03 -0.471028598915133E+03 -0.206117702991755E+03
8 2 2 0.901724903349687E+02 0.580006811201862E+02 0.28748166328728E+01
9 4
10 1 2 1 2 1 0.153678432214908E+02
11 1 2 2 2 1 -0.484175419345742E+02
12 2 2 1 2 1 -0.484175419345742E+02
13 2 2 2 2 1 0.000000000000000E+00
```

• Line 1 contains two integers, which give the polynomial orders $l$ and $l'$, i.e. POLY_ORDER_INTRA_BN and POLY_ORDER_INTRA_BMN.
• Line 2 contains the value $P_0$ from Eq. [13]
• Line 3 gives the number of lines in the following section, with lines 4 through 5 specifying the values of $r_i(a_1a_2)$ for each site pair type.
• Line 6 gives the number of lines in the following section, with lines 7 through 8 specifying the values of the intramolecular polynomial coefficients. Each of these lines contains $2+l$ values:
  * Columns 1 and 2 contain the site-types of sites $a_1$ and $a_2$.
  * Columns 3 through 5 contain values of the polynomial coefficients $c_{i}^{a_1a_2}$.
• Line 9 gives the number of lines in the following section, with lines 10 through 13 specifying the values of the mixed-distance intramolecular polynomial coefficients. Each of these lines contains $4+l'$ values:
  * Columns 1 and 2 give the site-types of the first site-pair-type.
  * Columns 3 and 4 give the site-types of the second site-pair-type.
  * Columns 5 through $5+l'-1$ give the values of the coefficients $c_{i}^{a_1a_2,a_3a_4}$ (in this example there is only one such value)
NAME.ener – Each NAME.ener file contains a set of grid point coordinates and corresponding energies. Multiple NAME.ener files are created at different parts of the PES generation, and are located in different directories. New NAME.ener files are created after each iteration of ab initio calculations, and these files are combined into a new NAME.ener file before each new fit is performed.

The format of the NAME.ener file is given below.

\[
R^1 \beta_A^1 \gamma_A^1 \alpha_B^1 \beta_B^1 \gamma_B^1 \phi_1^1 \ldots \phi_{N_{ID}}^1 E_{tot}^1 E_{clst}^1 E_{exch}^1 E_{ind}^1 E_{exch-ind}^1 E_{disp}^1 E_{exch-disp}^1 E_{\delta_{HF}}^1 E_A^1 E_B^1
\]

\[
R^2 \beta_A^2 \gamma_A^2 \alpha_B^2 \beta_B^2 \gamma_B^2 \phi_1^2 \ldots \phi_{N_{ID}}^2 E_{tot}^2 E_{clst}^2 E_{exch}^2 E_{ind}^2 E_{exch-ind}^2 E_{disp}^2 E_{exch-disp}^2 E_{\delta_{HF}}^2 E_A^2 E_B^2
\]

\ldots

- \( R^i \) is the center-of-mass separation between monomers in dimer \( i \), given in angstroms.
- \( \beta_A^1, \gamma_A^1, \alpha_B^1, \beta_B^1, \) and \( \gamma_B^1 \) are the Euler angles for monomers A and B of dimer \( i \) in the \( zyz \) convention, given in degrees. The angle \( \alpha_A^1 \) is always set to zero. See Ref. [1], Sec. II for more details.
- \( \phi_1^i, \ldots, \phi_{N_{ID}}^i \) are the values of the \( N_{ID} = N_{ID}^A + N_{ID}^B \) internal degrees of freedom, where \( N_{ID}^A \) and \( N_{ID}^B \) are the numbers of internal degrees of freedom in monomers A and B, respectively. The \( \phi_n^i \) values are ordered with those corresponding to IDOFs of monomer A first, and B second. In the case of rigid monomers, the only possible IDOF is chirality, and both \( N_{ID}^A \) and \( N_{ID}^B \) are equal to either 0 or 1. In the case of a homogeneous dimer with chiral monomers, the IDOF is used only for monomer A (without loss of generality). In the case of chirality, a value of \( \phi_n^i = 0.0 \) indicates no change from the undistorted input geometry, while a value of \( \phi_n^i = 1.0 \) indicates that the molecule is mirrored about its center of mass.
- \( E_{tot}^i \) is the total intermolecular interaction energy of dimer \( i \), given in kcal/mol.
- \( E_{clst}^i, E_{exch}^i, E_{ind}^i, E_{exch-ind}^i, E_{disp}^i, E_{exch-disp}^i, \) and \( E_{\delta_{HF}}^i \) are the respective SAPT(DFT) energy components of dimer \( i \), as described in the SAPT program manual [1], given in kcal/mol.
- \( E_A^i, E_B^i \) are the total monomer energies of monomer A and B, respectively, given in Hartrees.

Note that depending on the ab initio method used, some energy components may be equal to zero. If the supermolecular method is used, all components except the total interaction energy and monomer energies are equal to zero.

The files NAME.geo are created by the grid generation codes, and contain geometry coordinates in the same format as NAME.ener, but do not contain energy values.
- NAME.input_long – The NAME.input_long file contains all the information of the NAME.input file, plus some additional information, in a machine-friendly format. It is created by the make_input2 program, and read in by various programs.

The format of the NAME.input_long file is given below.

\[
\begin{align*}
N_A & \quad N_{A,\text{PHYS}}^A \quad N_{A,\text{OA}}^A \quad q_A \quad I_A \quad p_A \quad \alpha_A \\
N_B & \quad N_{B,\text{PHYS}}^B \quad N_{B,\text{OA}}^B \quad q_B \quad I_B \quad p_B \quad \alpha_B \\
a_1 & \quad x_{A1} \quad y_{A1} \quad z_{A1} \quad q_{A1} \quad s_{A1} \quad t_{A1} \\
a_2 & \quad x_{A2} \quad y_{A2} \quad z_{A2} \quad q_{A2} \quad s_{A2} \quad t_{A2} \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
A_{1,1} & \quad \ldots & \quad A_{1,n_{A1}^1} & \quad s_{A1,1}^1 & \quad \ldots & \quad s_{A1,n_{A1}^1} \\
A_{2,1} & \quad \ldots & \quad A_{2,n_{A2}^2} & \quad s_{A2,1}^2 & \quad \ldots & \quad s_{A2,n_{A2}^2} \\
b_1 & \quad x_{B1} \quad y_{B1} \quad z_{B1} \quad q_{B1} \quad s_{B1} \quad t_{B1} \\
b_2 & \quad x_{B2} \quad y_{B2} \quad z_{B2} \quad q_{B2} \quad s_{B2} \quad t_{B2} \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
B_{1,1} & \quad \ldots & \quad B_{1,n_{B1}^1} & \quad s_{B1,1}^1 & \quad \ldots & \quad s_{B1,n_{B1}^1} \\
B_{2,1} & \quad \ldots & \quad B_{2,n_{B2}^2} & \quad s_{B2,1}^2 & \quad \ldots & \quad s_{B2,n_{B2}^2} \\
P_{e1}^1 & \quad P_{e1}^1 \quad P_{e1}^1 \quad P_{e1}^1 \\
P_{e2}^1 & \quad P_{e2}^1 \quad P_{e2}^1 \quad P_{e2}^1 \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots 
\end{align*}
\]

– The first and second lines correspond to monomers A and B, respectively.

* \(N_A\) is the number of sites in monomer A.

* \(N_{A,\text{PHYS}}^A\) is the number of physical IDOFs in monomer A (currently only chirality is supported).

* \(N_{A,\text{OA}}^A\) is the number of off-atomic site IDOFs (not yet supported in the current version).

* \(q_A\) is the charge of monomer A, given as an integer in atomic units.

* \(I_A\) is the ionization potential of monomer A in units of electron volts.
* $p_A$ is the dipole moment of monomer A in units of Debye.
* $\alpha_A$ is the isotropic polarizability of monomer A in atomic units.

And similarly for monomer B.

- The next $N_A$ lines correspond to the sites of monomer A, with one line per site, in the same order as they appear in the NAME.input file.
  * $a_i$ is the site label, as given in the NAME.input file.
  * $x_i$, $y_i$, and $z_i$ are site coordinates in bohrs.
  * $q_i$, $s_i$, and $t_i$ are the atomic number (zero for off-atomic sites), symmetry index, and guiding force-field type, respectively. They have the same meaning as in the NAME.input file, described in Sec. 4.2.

- The next $N_A^{\text{ID}}$ pairs of lines correspond to the IDOFs of monomer A, including monomer chirality (if used).
  * $t_i^{\text{ID}}$ is the type of the IDOF $i$. A value of 1 indicates a LIN type IDOF, a value of 2 indicates an ROT type IDOF, a value of 3 indicates a CHI type IDOF, and a value of 4 indicates a BND type IDOF.
  * The values $w_i^{\text{ID}}$, $v_i^{\text{ID}}$, and $u_i^{\text{ID}}$ give the labels of the sites which define the axis of translation or rotation. In the case of a BND type IDOF, they define the plane that the rotational axis is normal to, while in the ROT or LIN cases the axis goes through the first two sites and the value $u_i^{\text{ID}}$ is unused.
  * The value $n_i^{\text{ID}}$ gives the number of sites that are bound to the IDOF, with each site listed in the next line.
  * The values $g_i^{\text{ID}}$ and $k_i^{\text{ID}}$ give the group index of the IDOF and the scale factor of the IDOF within that group. In the current version of autoPES, these are unused.
  * The values $s_i^{1,1}$ through $s_i^{1,n_i^{\text{ID}}}$ give labels of each of the sites that are bound to IDOF $i$.

- The next $N_B$ lines correspond to the sites of monomer B, with one line per site, in the same order as they appear in the NAME.input file. The variables have similar meanings as for monomer A.

- The next $N_B^{\text{ID}}$ pairs of lines correspond to the IDOFs of monomer B. The variables have similar meanings as for monomer A.

- The final section contains one line per site type, that is, the number of lines is equal to the largest $s_i$ value. The meanings of the variables $P_{\text{el}}^i$, $P_{\text{pol}}^i$, $P_{i+d}^i$, and $P_{\text{exp}}^i$ are the same as described in Sec. 4.2.
8.4 Programs and scripts in the autoPES suite

Unless otherwise noted, input arguments in this section which have the same names as options in Sec. 7 are exactly those values, passed to the subprogram.

- **add_geo** – FORTRAN 95 program which converts an XYZ dimer geometry file into the $R$ and Euler angle coordinates used by autoPES. Requires input arguments NAME, specifying the name of the system, GEO_FILE, specifying the .xyz file (in the standard format). The parameter MAX_DIST specifies a maximum closest-contact atom distance in angstroms that the program will consider a valid geometry (MAX_DIST equal to zero indicates no maximum distance). The parameter ERR_THRESH is given in bohrs and specifies the RMSD cutoff above which two geometries are not considered identical.

\$PATH/add_geo NAME GEO_FILE MAX_DIST ERR_THRESH

- **autoPES** – FORTRAN 95 program which is the front-end program for the autoPES package. It reads in the input file NAME.ctrl and calls the master.sh script. When called, it requires the NAME and optional MODE variables (see Sec. 4.4).

\$PATH/autoPES NAME MODE

- **combine_ener** – FORTRAN 95 program which combines results of SAPT(DFT) calculations and $\delta E_{\text{HF}}$ calculations into a single file. Requires as inputs NAME, the names of the two data files, and a logical value INC_DELTA indicating whether the delta-HF correction is included. If INC_DELTA is set to ‘F’, the program simply acts as a pass-through.

\$PATH/combine_ener NAME MAIN_ENER_FILE DELTAHF_ENER_FILE INC_DELTA

- **dimer_plot** – FORTRAN 95 program which evaluates many PES geometries at a time for use with the Dimerplot graphical utility. Batches of Cartesian coordinates are inputted and corresponding energy components outputted in unformatted binary from standard input and output.

\$PATH/dimer_plot NAME FF_FILE INTRA_FILE MODE MODEPARM < coordinates.dat

- **eval_pes** – FORTRAN 95 program which evaluates a generated PES at a given geometry. Requires the name of the intermolecular fit file and an XYZ dimer geometry file as input arguments. Optionally, a third argument can be provided which gives the name of the intramolecular fit file. The XYZ file must be in the standard ‘.xyz’ format, and have the atoms in the order that they appear in the NAME.input file used to generate the parameter file, with monomer A first and monomer B second.
$PATH/eval_pes FF_FILE INTRA_FILE < geo.xyz

- **extract_ener** – FORTRAN 95 program which extracts energies from *ab initio* output files and creates a NAME.ener file. Required inputs are NAME, OUTPUT_FILE, MODE (either ‘dft’, ‘deltahf’, ‘supmol’, ‘mono_dft’, ‘mono_cc’, or ‘asym’ depending on the type of OUTPUT_FILE), a logical value HYBRID (only used if MODE is ‘dft’), INTERFACE_PROGRAM, and CBS_TZ_BASIS.

$PATH/extract_ener NAME OUTPUT_FILE MODE HYBRID INTERFACE_PROGRAM CBS_TZ_BASIS

- **find_minima.sh** – (DEPRECATED IN FAVOR OF minimia_genetic) Bash script which finds local minima of the given PES. It creates input files, runs multiple instances of pesw_ff (described below) in parallel, collects outputs, then calls sort_minima to sort and filter the minima. Required arguments are NAME, BIN_DIR (points to the autoPES/bin directory), FF_FILE and FF_TYPE (which specify the name and type of PES parameter file), and NPAR (which specifies the number of parallel instances of pesw_ff to run). Requires the files NAME.input_long and FF_FILE and produces output file minima.dat.

$PATH/find_minima.sh NAME BIN_DIR FF_FILE FF_TYPE NPAR

- **fit_asymp** – FORTRAN 95 program which fits the distributed asymptotic expansion of the PES using an implementation of Powell’s algorithm[18]. It requires NAME.input_long and NAME.ener input files. It generates fit_asymp.dat, which is used as input for the main fitting program. Required arguments are NAME, COM_RULE (1 = use geometric combination rules, 0 = site-pair specific parameters), MAX_CN (maximum power in the induction plus dispersion expansion), INC_PEN (logical value indicating whether or not to use the dlDF penalty function), ELS_PEN_STR, POLY_ORDER_ELS, POLY_ORDER_CN, and INNDISP_SCALE_FACT (asymptotic induction and dispersion energies are scaled by this value).

$PATH/fit_asymp NAME INTRA_FILE COM_RULE_VDW MAX_CN INC_PEN ELS_PEN_STR POLY_ORDER_ELS POLY_ORDER_CN INDDISP_SCALE_FACT

- **fit_inter** – FORTRAN 95 program which performs the fitting of the intermolecular potential energy surface using an implementation of Powell’s algorithm [18]. It requires as standard input a FORTRAN namelist file fit.in specifying fit options. It reads in NAME.input_long, NAME.ener, fit_asymp.dat, and fit_intra.dat (in the flexible case). It generates fit_inter.dat, which contains all optimized fit parameters. The available options in the fit.in file are described in the comments of the fit_inter.f90 file.

43
$PATH/fit_inter NAME < fit.in

- **fit_intra** – FORTRAN 95 program which performs the fitting of the intramolecular potential energy surface. It reads in NAME.input_long and NAME.ener. It generates fit_intra.dat, which contains all optimized intramolecular fit parameters for both monomers. Required arguments are NAME, POLY_ORDER_INTRA_BN, POLY_ORDER_INTRA_BMN, INTRA_CONSTRAINT_SCALE (unused), FIT_WEIGHT_SCALE, and GEO_IDOF_THRESH.

$PATH/fit_intra NAME POLY_ORDER_INTRA_BN POLY_ORDER_INTRA_BMN 0.0 FIT_WEIGHT_SCALE 1.0

- **gen_report** – FORTRAN 95 program which generates a human-readable report file containing important information about the fit. Requires as input NAME, FF_FILE and FF_TYPE specifying an intermolecular parameter file and type, INTRA_FILE specifying an intramolecular parameter file, optional MINIMA_FILE containing local minima of the PES, the four combination rules used, and four values giving information about the repulsive wall: MIN_WALL_HEIGHT, MEAN_WALL_HEIGHT, MIN_WALL_DIST, and MAX_WALL_DIST.

$PATH/gen_report NAME FF_FILE FF_TYPE INTRA_FILE MINIMA_FILE
  COM_RULE_EXP COM_RULE_DMP COM_RULE_LIN COM_RULE_VDW
  MIN_WALL_HEIGHT MEAN_WALL_HEIGHT MIN_WALL_DIST MAX_WALL_DIST

- **gridgen_ff** – FORTRAN 95 program which generates the main set of grid points for the *ab initio* calculations. Requires the following variables: NAME, NGRID (number of points to generate), FF_FILE (file containing the force field parameters), FF_TYPE (1 for initial guiding potential or 2 for generated potential), GRID_WEIGHT_SCALE, WALL_HEIGHT, GRID_MODE, FRACR0, IDOF_THRESH. The final two arguments are only used when IDOFs are defined. If the file NAME.ener exists, it will be used to define pre-existing grid points. Creates the output file NAME.geo_main.

$PATH/gridgen_ff NAME NGRID FF_FILE FF_TYPE INTRA_FILE GRID_WEIGHT_SCALE
  GRID_WEIGHT_SCALE_INTRA WALL_HEIGHT GRID_MODE FRACR0 IDOF_THRESH

- **gridgen_ivmg** – FORTRAN 95 program which generates the main set of grid points for the *ab initio* calculations when the IVMG option is enabled. Requires the following variables: NAME, FF_FILE (file containing the intermolecular fit parameters), INTRA_FILE (file containing intramolecular fit parameters), NGRID (number of points to generate), LARGE_GRID_SIZE (size of the discretized domain), INC_DELTA (whether to include delta
terms in IVMG (T) or set them to zero (F)), GRID_WEIGHT_SCALE, COM_RULE_EXP, COM_RULE_LIN, FRACR0, IDOF_THRESH. The final two arguments are only used when IDOFs are defined. If the file NAME.ener exists, it will be used to define pre-existing grid points. Creates the output file NAME.geo

$PATH/gridgen_ivmg NAME FF_FILE INTRA_FILE NGRID LARGE_GRID_SIZE INC_DELTA
GRID_WEIGHT_SCALE COM_RULE_EXP COM_RULE_LIN FRACR0 IDOF_THRESH

• get_nfreeparm – FORTRAN 95 program which determines the number of free parameters in the final stage of fitting of the complete intermolecular PES.

$PATH/get_nfreeparm NAME COM_RULE_EXP COM_RULE_LIN POLY_ORDER
POLY_ORDER_MIXED_INTER POLY_ORDER_MIXED_INTRA

• get_nfreeparm_intra – FORTRAN 95 program which determines the number of free parameters in the final stage of fitting of the intramolecular PESs. Outputs the maximum of the number of parameters for monomers A and B.

$PATH/get_nfreeparm_intra NAME GEO_IDOF_THRESH
POLY_ORDER_INTRA_BN POLY_ORDER_INTRA_BMN

• gridgen_min – FORTRAN 95 program which generates grid points around the local minima. As input it reads a previously generated text file with the coordinates of the minima. As output, a file NAME.geo_min is produced, which contains grid points for the subsequent ab initio calculations. Requires input arguments NAME, MINIMA_FILE (a file containing minima coordinates), and NPOINT (the number of grid points to add).

$PATH/gridgen_min NAME MINIMA_FILE NPOINT

• gridgen_mono – FORTRAN 95 program which generates grid points for the intramolecular grids. As output, a file NAME.geo is produced, which contains grid points for the subsequent monomer ab initio calculations. Requires input arguments NAME, NPOINT_INTRA (the number of grid points to generate), and GEO_IDOF_THRESH.

$PATH/gridgen_mono NAME NPOINT_INTRA GEO_IDOF_THRESH

• hole_scan – FORTRAN 95 program which scans for holes in the repulsive wall of the generated PES. Requires input variables NAME, FF_FILE (containing fit parameters), FF_TYPE (usually 2), INTRA_FILE (containing intramolecular fit parameters), WALL_HEIGHT, MIN_ENERGY (value of PES global minimum), and WALL_MIN_DIST.
$PATH/hole_scan NAME FF_FILE FF_TYPE INTRA_FILE WALL_HEIGHT MIN_ENERGY WALL_MIN_DIST

• **make_asym_grid** – FORTRAN 95 program, which creates a NAME.ener for the asymptotic calculations. It requires input arguments NAME, DISTR, and NPOINT (the number of grid points to generate).

$PATH/make_asym_grid NAME DISTR NPOINT

• **make_asym_mono_grid** – FORTRAN 95 program, which creates monomer asymptotic grids in the flexible-monomer case. Each line of the output file NAME.geo contains coordinates for only one or the other monomer. It requires input arguments NAME, INTRA_FILE, WALL_HEIGHT, and NGRID_MULT (the number of grid points is equal to this times the maximum number of IDOFs in either monomer and it has default value 5).

$PATH/make_asym_grid NAME INTRA_FILE WALL_HEIGHT 5

• **make_dalt_inp** – FORTRAN 95 program, which creates DALTON inputs (*.mol and *.dal) to be used in the asymptotic SAPT calculations. Additionally this program creates input NAMEP.data and NAME.dispinp files for the asymptotic SAPT program. Requires input arguments NAME, BASDIR (location of the basis set repository), BASIS, AUXBASIS, HYBRID, MODE (set to ‘asym’, ‘distr’, or ‘deltahf’), MB_CUTOFF, and MAX_MEM (memory for SAPT program in words).

$PATH/make_dalt_inp NAME BASDIR BASIS HYBRID MODE MB_CUTOFF MAXMEM

• **make_dlpoly_inp** – FORTRAN 95 program, which generates the FIELD and TABLE input files for DL_POLY. It requires the following variables: NAME, FF_FILE (PES parameter file), R_CUTOFF (DL_POLY potential cutoff radius).

$PATH/make_dlpoly_inp NAME FF_FILE R_CUTOFF

• **make_input2** – FORTRAN 95 program which takes user-supplied input files NAME.input and NAME.ctrl and generates NAME.input_long, which is used by most other programs. Requires as input NAME. Optional inputs are IP_FILE (a file containing monomer ionization potentials), DM_FILE (a file containing monomer dipole moments), PZ_FILE (a file containing monomer isotropic polarizabilities). If the optional input files are not provided, values of zero are used.

$PATH/make_input2 NAME IP_FILE DM_FILE PZ_FILE
• **make_ip_inp** – FORTRAN 95 program, which creates ORCA inputs for the ionization potential and isotropic polarizability calculations. It requires the following variables: NAME, BASIS, POLARIZABLE (whether the induced dipole model is used)

$PATH/make_ip_inp NAME BASIS POLARIZABLE

• **make_molpro_inp** – FORTRAN 95 program, which creates the input files for the *ab initio* calculations with MOLPRO interface. Files created by this program are: NAMEA.data, NAMEB.data, NAMEP.data, NAME.data, NAME.aux. Required input arguments are NAME, BASDIR (basis directory of the autoPES package), BASIS, AUXBASIS, CBS,TZ,BASIS, MODE (set to ‘supmol’ or ‘mono_cc’), MB,CUTOFF, FROZEN,CORE, NCORES,SCF, and MAX_MEM (memory for MOLPRO in words).

$PATH/make_molpro_inp NAME BASDIR BASIS AUXBASIS CBS_TZ_BASIS MODE MB_CUTOFF FROZEN_CORE NCORES_SCF MAX_MEM

• **make_orca_inp** – FORTRAN 95 program, which creates the input files for the *ab initio* calculations with ORCA interface. Output file is NAME.input. Required input arguments are NAME, BASDIR (basis directory of the autoPES package), BASIS, AUXBASIS, CBS,TZ,BASIS, HBRID, MODE (set to ‘dft’, ‘deltahf’, ‘distr’, ‘distr_A’, ‘distr_B’, ‘distr_dim’, ‘supmol’, ‘mono_dft’, or ‘mono_cc’), MB,CUTOFF, NCORES,SCF, and MAX_MEM (memory for SAPT programs in words).

$PATH/make_orca_inp NAME BASDIR BASIS AUXBASIS CBS_TZ_BASIS HYBRID MODE MB_CUTOFF NCORES_SCF MAX_MEM

• **make_pesw_inp** – (DEPRECATED IN FAVOR OF minimia_genetic) FORTRAN 95 program, which prepares input files for the pesw_ff program. It requires input variables NAME, NPART (the number of parallel batches of points to create), NCOORD (the number of starting dimer configurations), and NEIG (the order of eigenvector to follow, usually 1).

$PATH/make_pesw_inp NAME NPART NCOORD NEIG

• **make_saptddft_inp** – FORTRAN 95 program, which creates the input files for the *ab initio* calculations with the SAPT(DFT) interface. Output file is NAME.input. Required input arguments are NAME, BASIS, AUXBASIS, HBRID, MODE (set to ‘dft’ or ‘deltahf’), MB,CUTOFF, NCORES,SCF, and MAX_MEM (memory for SAPT programs in words).

$PATH/make_saptddft_inp NAME BASIS AUXBASIS HYBRID MODE MB_CUTOFF NCORES_SCF MAX_MEM
• **minima_genetic** – FORTRAN 95 program which attempts to locate all minima on a PES using a genetic algorithm. Required command line parameters are NAME, FF_FILE (containing fit parameters), FF_TYPE (usually 2), INTRA_FILE (containing intramolecular fit parameters), STATIONARY (logical value, T to minimize gradient, F to minimize energy), and OPTIM_IDOF (logical, T to optimize all degrees of freedom or F to freeze IDOFs).

```plaintext
$PATH/minima_genetic NAME FF_FILE FF_TYPE INTRA_FILE STATIONARY OPTIM_IDOF
```

• **minimize_point** – FORTRAN 95 program, which performs local minimization of either energy or gradient of energy starting from a single grid point using Powell’s method. Starting configuration is read in from standard input in the usual grid point format. Chirality remains fixed at the input value. Required command line parameters are NAME, NITER (integer giving the number of iterations to perform), STATIONARY (logical value, T to minimize gradient, F to minimize energy), and OPTIM_IDOF (logical, T to optimize all degrees of freedom or F to freeze IDOFs).

```plaintext
$PATH/minimize_point NAME NITER STATIONARY OPTIM_IDOF
```

• **master.sh** – Bash script, which is the main script for the autoPES suite. It is called by the autoPES front-end program and subsequently calls all other program and scripts. It requires as inputs all options read in from `NAME.ctrl` (or defaults).

• **pesw_ff** – (DEPRECIATED IN FAVOR OF minima_genetic) FORTRAN 77 program which searches for the local minima of the PES. This program requires three input files `cluster.dat`, `control.dat` and `coords.dat`, which are created by the `make_pesw_inp` program. Reads the input file `NAME.input_long` and produces results through standard output. Requires the input arguments NAME, FF_FILE, and FF_TYPE.

```plaintext
$PATH/pesw_ff NAME FF_FILE FF_TYPE
```

• **print_table.sh** – Bash script which prints the iteration summary table. It must be run from the main PES generation directory.

```plaintext
$PATH/print_table.sh
```

• **run_abinitio_jobs.sh** – Bash script which is called by `slave_sapt.sh` and handles the interface with the system’s job queue. Required input arguments are MODE (set to either ‘dft’, ‘deltahf’, ‘supmol’, ‘mono_dft’, or ‘mono_cc’), and WAIT (whether to return immediately, or wait for jobs to finish), PART_COMMAND (the actual work command to run),
FINISH_COMMAND (command to run on job completion to extract data from job output), INPUT_FILES (list of required input files), OUTPUT (list of required output files), GOODSTR (string that needs to be in the job output to confirm successful completion), BADSTR (string which must not be in the job output to confirm successful job completion), and BADSTR2 (alternate BADSTR).

$PATH/run_abiniito_jobs.sh MODE WAIT PART_COMMAND FINISH_COMMAND INPUT_FILES OUTPUT GOODSTR BADSTR BADSTR2

• slave_asym.sh – Bash script which perform the asymptotic calculations and asymptotic fit in the rigid-monomer case. It is called by the master.sh script and requires the logical input WAIT, indicating whether the script should return immediately after submitting jobs (allowing the master.sh script to move to other parts of PES generation) or wait until job completion.

$PATH/slave_asym.sh WAIT

• slave_asym_idof.sh – Bash script which perform the asymptotic calculations and asymptotic fit in the flexible-monomer case. It is called by the master.sh script and requires the logical input WAIT, indicating whether the script should return immediately after submitting jobs (allowing the master.sh script to move to other parts of PES generation) or wait until job completion.

$PATH/slave_asym_idof.sh WAIT

• slave_fit.sh – Bash script which is called by master.sh and performs complete PES fitting and hole scanning. It requires input INC_TEST_SET (whether to use a test data set).

$PATH/slave_fit.sh INC_TEST_SET

• slave_fit_intra.sh – Bash script which is called by master.sh and performs intramolecular PES fitting.

$PATH/slave_fit_intra.sh

• slave_gridgen.sh – Bash script which performs the grid generation. It is called by the master.sh script and requires input FF_TYPE

$PATH/slave_gridgen.sh FF_TYPE
- **slave_ip.sh** – Bash script which performs ionization potential and isotropic polarizability calculations using ORCA.

  `$PATH/slave_ip.sh`

- **slave_sapt.sh** – Bash script which performs *ab initio* calculations. This script calls `run_abinitio_jobs` to handle the interface with the system’s job queue. Required input arguments are `MODE` (set to either ‘dft’, ‘deltahf’, ‘supmol’, ‘mono_dft’, or ‘mono_cc’), and `WAIT` (whether to return immediately, or wait for jobs to finish).

  `$PATH/slave_sapt.sh MODE WAIT`

- **submit.sh** – Bash script which takes input arguments which provide information about some computationally intensive work to be performed, and performs that work. This must be modified to work with a given computer system as part of the autoPES installation process. Generally this will interface with the job queuing mechanism on a given computer system. See the example file `submit.sh.EXAMPLE` for details.

  `$PATH/submit.sh RUN_MODE SYSTEM JOBNAME INFILES OUTFILES WORK NCORES_PER NCORES_TOT`

- **sort_minima** – (DEPRECATED IN FAVOR OF `minima_genetic`) FORTRAN 95 program which filters and sorts minima of the PES found by `pesw_ff`. As an input, it reads a previously generated text file with the coordinates of the minima. The standard output is a sorted list of local minima. Requires input arguments `NAME`, `MINIMA_FILE` (a file containing minima coordinates), `FF_FILE` (containing fit parameters), and `FF_TYPE` (1 = initial guiding potential, 2 = generated potential).

  `$PATH/sort_minima NAME MINIMA_FILE FF_FILE FF_TYPE`

- **toxyz** – FORTRAN 95 program which converts the *R* and Euler angle coordinates used by autoPES to Cartesian coordinates. `toxyz` requires as input the name of the system, and must be run from a directory containing the `NAME.input_long` file. Standard input is a space-separated list of coordinates as they appear on a single line of a `NAME.ener` file.

  `$PATH/toxyz NAME < coors.dat`
References


