Manual

PIPFit:
A program to perform linear least-squares fitting of potential energy surfaces using permutationally invariant polynomials

Ke R. Yang, Zoltan Varga, and Donald G. Truhlar
Department of Chemistry, Chemical Theory Center, and Supercomputing Institute
University of Minnesota, Minneapolis, Minnesota 55455

version 2015
Date of most recent program change July 28, 2015
Date of most recent manual update: July 21, 2021

Abstract. This program can be used to fit potential energy surfaces of four-body systems. A weighted least-squares fit is performed with permutationally invariant polynomials (PIPs) of Morse, Gaussian, mixed exponential Gaussian, or hyperbolic secant variables. Three kinds of fit can be performed with the program: PIPs as original proposed by Braams, Bowman, and Xie, modified PIPs by removing the unconnected terms, or modified PIPs by removing the unconnected and the two-body terms. The program can also perform a two-stage fitting scheme in which one fits lower-level energy data at a large number of points and higher-level corrections at a small set.
Table of Contents

Manual.................................................................................................................................1
1. Licensing........................................................................................................................3
2. Citation..........................................................................................................................3
3. Introduction....................................................................................................................4
4. Theoretical background..............................................................................................5
5. Installation and general usage ....................................................................................11
6. Options.........................................................................................................................14
7. Program structure........................................................................................................23
8. Test inputs....................................................................................................................27
9. Revision history............................................................................................................31
1. Licensing

\textit{PIPFit} - version 2015 is licensed under the\footnote{Apache License, Version 2.0.} \hyperlink{https://www.apache.org/licenses/LICENSE-2.0}{Apache License, Version 2.0}. The manual of \textit{PIPFit} - version 2015 is licensed under\footnote{Creative Commons Attribution 4.0 license.} \hyperlink{https://creativecommons.org/licenses/by/4.0}{CC-BY-4.0}. Publications of results obtained with the \textit{PIPFit} - version 2015 software should cite the program and/or the article describing the program.

No guarantee is made that this software is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within. No consulting or maintenance services are guaranteed or implied.

The use of the \textit{PIPFit} - version 2015 implies acceptance of the terms of the licenses.

2. Citation

Publications including work performed with \textit{PIPFit} should cite the software, for example, as follows:

Ke R. Yang, Zoltan Varga, and Donald G. Truhlar \textit{PIPFit} – version 2015, University of Minnesota, Minneapolis, 2015. \url{http://comp.chem.umn.edu/pipfit}
3. Introduction

This program can be used to fitting potential energy surfaces by using a weighted least-squares fit and the method of


This method uses a basis of permutationally invariant polynomials (PIPs) of Morse, Gaussian, mixed exponential-Gaussian, or hyperbolic secant variables. Three kinds of fit can be performed:

- IMethd = 0: Original Braams–Bowman–Xie approach, i.e., a direct fit to the full surface without removing unconnected terms;
- IMethd = 1: Remove the unconnected terms;
- IMethd = 2: Remove unconnected terms and fit difference from pairwise interactions.

The current version of the code can treat four-body systems of the form \( A_4 \) or \( A_2B_2 \) and has been tested for \( N_4, O_4 \), and \( N_2O_2 \). The user can choose to read input in either internal coordinates or Cartesian coordinates. The maximum order of the polynomials is an input variable and can be selected in the range from 5 to 12 for \( A_4 \) systems and in the range from 5 to 10 for \( A_2B_2 \) systems. Two formats are allowed for the input data files; the first format does not include indices of data points in the input file, and the second format does include indices of data points.
4. Theoretical background

4.1. Fitting basis functions

The inter-nuclear distances are used in the construction of potential energy surfaces (PESs). In the four-atom case, the six internuclear distances are defined as follows:

\[ r_1 = r_{12}, \quad r_4 = r_{23}, \]
\[ r_2 = r_{13}, \quad r_5 = r_{24}, \]
\[ r_3 = r_{14}, \quad r_6 = r_{34}, \]

where \( r_{ij} \) is the distance between atoms \( i \) and \( j \). The variables \( X_i \) are used to describe the PESs; these variables can be

- Morse variables \( X_i = \exp\left(\frac{-\left(r_i - r_a\right)}{a_i}\right) \),
- Gaussian variables \( X_i = \exp\left(\frac{-\left(r_i - r_a\right)^2}{a_i}\right) \), or
- Mixed exponential Gaussian (MEG) variables
  \( X_i = \exp\left[\frac{-\left(r_i - r_a\right)}{a_i} - \frac{\left(r_i - r_b\right)^2}{b_i}\right] \), or
- Hyperbolic secant (Sech) variables \( X_i = \frac{1}{\cosh\left(\frac{r_i - r_a}{a_i}\right)} \)

The PES is expanded in a Taylor series of \( X_i(r_j) \) as

\[
V(\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6) = \sum_{n_1+n_2+n_3+n_4+n_5+n_6=0}^{k} C_{n_1n_2n_3n_4n_5n_6} X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6},
\]

where \( n_i \) is the order of polynomial of \( X_i \), and the coefficients \( C_{n_1n_2n_3n_4n_5n_6} \) are the linear parameters that need to be determined through least-squares fits. In eq. 1 the \( n_1, n_2, n_3, n_4, n_5, \) and \( n_6 \) variables are starting from 0 values and their sum is equal to \( k \); therefore \( k \) is the maximum degree of the multinomials.

Due to the permutation symmetry of the same kind of atoms, some of the coefficients in eq. 1 are identical. One can therefore construct an explicitly permutationally invariant basis, \( S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}] \), as proposed by Braams, Bowman, and Xie.\(^1\)\(^2\) Thus eq. 1 is written as

Some terms in eq. 1, such as \( X_1^{n_1}X_6^{n_6} \), are products of the functions of unconnected distances, e.g., a function of the 1–2 distance \((r_1 = r_{12})\) times a function of the 3–4 distance \((r_6 = r_{34})\). Those terms, which we call unconnected terms, introduce spurious interactions between fragments 1–2 and 3–4 since they survive even when two fragments are far from each other. These unconnected terms should be removed in the construction of a permutationally invariant polynomials basis.\(^3\) By removing the unconnected terms, eq. 2 is modified to

\[
V(r_1, r_2, r_3, r_4, r_5, r_6) = k \sum_{n_1+n_2+n_3+n_4+n_5+n_6=0} D_{n_1n_2n_3n_4n_5n_6} S[X_1^{n_1}X_2^{n_2}X_3^{n_3}X_4^{n_4}X_5^{n_5}X_6^{n_6}].
\]

Other terms in eq. 1, such as \( X_i^{n_i} \), describe two-body interactions. We can further use pre-optimized two-body potential energy functions \( V_{\text{PA}}(r_i) \) for better asymptotic behavior. By using such pre-optimized two-body potentials, those terms can be detached as known terms and we get the following form from eq. 3:

\[
V(r_1, r_2, r_3, r_4, r_5, r_6) = V_0 + \sum_{i=1}^{6} V_{\text{PA}}(r_i) + \sum_{n_1+n_2+n_3+n_4+n_5+n_6=0} k \sum_{\text{connected}} D_{n_1n_2n_3n_4n_5n_6} S[X_1^{n_1}X_2^{n_2}X_3^{n_3}X_4^{n_4}X_5^{n_5}X_6^{n_6}].
\]

where \( V_0 \) is a constant equal to the energy of four atoms. Equation 4 is the recommended method to fit PESs with permutationally invariant polynomials (PIPs).

The diatomic potential energy functions of \( \text{N}_2 \), \( \text{O}_2 \), and \( \text{NO} \) have been fitted the following generalized Morse potential:

\[
V_{\text{PA}}(r) = D_e \left[ 1 - \exp(-f(r)(r-r_e)) \right]^2 - D_e,
\]

where $D_e$ is the equilibrium bond dissociation energy and $f(r)$ depends on $r$. We
defined intermediate variable $y = \frac{r^4 - r_e^4}{r^4 + r_e^4}$ and expanded $f(r)$ in a Taylor series of $y$:

$$f(r) = \sum_{i=0}^{6} a_i y^i.$$  \hspace{1cm} (6)

The linear coefficients $a_i$ are optimized to yield accurate two-body interactions.

In some cases, two sets of basis functions are used (option $\text{nfun}=2$, see later) for the fitting process. For such treatments, one can write eqs. 2, 3, and 4 as:

$$V(r_1, r_2, r_3, r_4, r_5, r_6) = D_{000000}$$
$$+ \sum_{n_1+n_2+n_3+n_4+n_5+n_6=1}^{k_d} D_{a}^{n_1 n_2 n_3 n_4 n_5 n_6} S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}]$$  \hspace{1cm} (2')
$$+ \sum_{n_1+n_2+n_3+n_4+n_5+n_6=1}^{k_b} D_{b}^{n_1 n_2 n_3 n_4 n_5 n_6} S[Y_1^{n_1} Y_2^{n_2} Y_3^{n_3} Y_4^{n_4} Y_5^{n_5} Y_6^{n_6}]$$  \hspace{1cm} (3')

and

$$V(r_1, r_2, r_3, r_4, r_5, r_6) = V' + \sum_{j=1}^{6} V_{pA}(r_j) +$$
$$+ \sum_{n_1+n_2+n_3+n_4+n_5+n_6=0}^{k_d} \frac{D_{a}^{n_1 n_2 n_3 n_4 n_5 n_6}}{r_e} S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}]$$  \hspace{1cm} (4')
$$+ \sum_{n_1+n_2+n_3+n_4+n_5+n_6=0}^{k_b} \frac{D_{b}^{n_1 n_2 n_3 n_4 n_5 n_6}}{r_e} S[Y_1^{n_1} Y_2^{n_2} Y_3^{n_3} Y_4^{n_4} Y_5^{n_5} Y_6^{n_6}]$$

In eqs. 2'-4', $a$ and $b$ denote two different sets of permutationally invariant polynomial basis functions with the orders of $k_d$ and $k_b$, respectively. Both $X_i$ and $Y_i$ variables
independently can be Morse, Gauss, MEG, or Sech variable, and it provides an improved flexibility for the fit.

4.2. Separate fits of lower-level energy results and high-level corrections

The program can also be used for a procedure in which one fits a lower-level (LL) energy at a large number of points and a higher-level (HL) correction at a subset of these points. The motivation for this is that the high-level calculations are more expensive, but since the difference between high level and low-level energies is smoother than the energy surface itself, the high-level calculations might not be needed at as many points as are required to map out the shape of the surface in a single-level fit.

For instance, one can do CASSCF calculations, where - due to the lack of dynamic electron correlation - the available accuracy is limited. But such calculations can be considered to be a LL calculation that – due to its lower cost – can be carried out for a relatively large number of points. Then, for higher accuracy one can use HL calculations, for instance, subsequent CASPT2 calculations at a smaller number of points. And the energy difference between the two calculations \( E_{\text{CASPT2}} - E_{\text{CASSCF}} \) is the HL correction. This gives two data sets, one with the LL data (geometries and energies) and the other one with the HL data (geometries and energy corrections).

Then the potential energy surfaces of LL energies and HL corrections can be fitted separately and their sum gives final fit \( V_{\text{Total}} \). The surface of LL energies is usually as rugged as the surface of \( V_{\text{Total}} \); therefore it requires a large number of data points to get a reasonable surface. But the surface of HL corrections is relatively smooth, and its energy range is also relatively narrow as compared to the LL surface. Hence, the HL surface can be described well with a significantly lower number of input data points. In our tests, 1/10 or even 1/15 of the LL data set provided acceptable accuracy.

In the above example, we used CASSCF as the LL method and CASPT2 as the HL method, but in general any kind of combinations of methods can be considered. For instance, LL can be HF or MP2, while the HL corrections may come from CCSD(T).
4.3. Weighted least-squares fit

With $m$ basis functions to fit $n$ data points, the linear coefficients need to be determined by least-squares fitting if $n > m$, which is essentially always the case. Denote by $V_i$ the energy of data point $i$, by $x_i$ the geometry of data point $i$, by $b_{ij}$ the basis function $j$ at geometry $x_i$, and by $c_j$ the linear coefficient ($j = 1, \ldots, m$). The determination of linear coefficients is equivalent to minimizing the functional,

$$F = \sum_{i=1}^{n} \sum_{j=1}^{m} (b_{ij} c_j - V_i)^2.$$  \hspace{1cm} (10)

Weights, $W_{ii}$, can be assigned to different data point $i$ ($i = 1, \ldots, n$), yielding

$$F = \sum_{i=1}^{n} \sum_{j=1}^{m} W_{ii} (b_{ij} c_j - V_i)^2.$$  \hspace{1cm} (11)

The minimization of the functional $F$ in eq. 11 leads to the normal equation,

$$B^T W B c = B^T W V,$$  \hspace{1cm} (12)

where $^T$ denotes the matrix transpose operation, and

$$B = \begin{bmatrix}
  b_1(x_1) & b_2(x_1) & \cdots & b_m(x_1) \\
  b_1(x_2) & b_2(x_2) & \cdots & b_m(x_2) \\
  \vdots & \vdots & \ddots & \vdots \\
  b_1(x_n) & b_2(x_n) & \cdots & b_m(x_n)
\end{bmatrix}, \quad c = \begin{bmatrix}
  c_1 \\
  c_2 \\
  \vdots \\
  c_m
\end{bmatrix}, \quad V = \begin{bmatrix}
  V_1 \\
  V_2 \\
  \vdots \\
  V_n
\end{bmatrix},$$

$$W = \begin{bmatrix}
  W_{11} & 0 & \cdots & 0 \\
  0 & W_{22} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & W_{nn}
\end{bmatrix}.$$  

Various weighting functions can be used for weighted least-squares fits. In PIPFit program, we use the following weighting function to avoid putting too much emphasis on the high-energy data points:

$$W_{ii} = W_{ii}^{en} = \begin{cases}
  1 & \text{for } V_i < E_{cr} \\
  \left(\frac{E_{cr}}{V_i}\right)^{E_{pow}} & \text{for } V_i > E_{cr}
\end{cases}.$$  \hspace{1cm} (13)
where $W_{ii}^{en}$ is the weighting factor of data point $i$ with energy $V_i$, and $E_{cr}$ is a selectable energy threshold to reduce the weights of high-energy data points. And $E_{pow}$ is a selectable power (the default value is 2).

An extended version of the above $W_{ii}$ weighting function contains a geometry dependent weighting function:

$$W_{ii} = W_{ii}^{en} W_{ii}^{geom},$$  \hspace{1cm} (14)

where $W_{ii}^{geom}$ is calculated the following way. First define

$$d_{i,k} = \left( \frac{1}{6} \sum_{j=1}^{n} \frac{(r_{i,j} - r_{k,j})^2}{(r_{i,j} + r_{k,j})^2} \right)^{\frac{1}{2}}.$$

(15)

Here $r'_{i,j}$ is one of the internuclear distances ($j$) for a given data point $i$, and $r'_{k,j}$ is an internuclear distance between atoms with the same atomic numbers for any other data point $k$. Note that the bonds (among $j=1$-6) between the same type of atoms (e.g., in $N_4$ the six N-N bonds) of the original $r_{i,j}$ and $r_{k,j}$ data set are reorganized into ascending order to get $r'_{i,j}$ and $r'_{k,j}$. For instance, in the case of $N_4$, there are six N-N distances so $r'_{i,j}$ and $r'_{k,j}$ with $j=1$ refer to the shortest N-N distance, while $j=6$ refers to the longest N-N distance. And in case of $N_2O_2$ it means that in $r'_{i,j}$ and $r'_{k,j}$ with $j=1$ refer to the unique O-O distance, $j=2, 3, 4, \text{ and } 5$ are the NO distances in ascending order, and $j=6$ is the unique N-N distance. Then an average value, called $d_i$, is calculated:

$$d_i = \frac{1}{n} \sum_{k=1}^{n} d_{i,k},$$  \hspace{1cm} (16)

and the $d_i$ values are normalized based on the highest value in the $d_i$ set:

$$W_{ii}^{geom} = \frac{d_i}{\max(d_i)}.$$  \hspace{1cm} (17)
5. Installation and general usage

5.1. PIPFit program

The PIPFit software package is free-of-charge and can be downloaded at

http://comp.chem.umn.edu/pipfit.

The PIPFit is distributed as a tar.gz file, which can be untarred by executing

% tar -xvf PIPFitv2015.tar.gz

This should create the following directories:

```
doc/  For the Manual
src/  PIPFit program code
src_list/  PIPLList program code
test/  Examples
```

The Fortran source codes to perform PES fit are in the src/ directory. A make file named makefile is also in that directory. LAPACK\(^4\) (http://www.netlib.org/lapack/) is required to solve the linear least squares problem by the subroutine DGELSD, and a path to this library should be provided in the make file; see the following example:

```
COMPLIB = /soft/lapack/3.4-gcc4.5/lib/liblapack.so
/soft/lapack/3.4-gcc4.5/lib/libblas.so
```

To compile the source codes you need to load the Intel FORTRAN compiler. For example, on high-performance computers at Minnesota Supercomputing Institute, this requires the following command:

% module load intel

Then, type “make” to compile the source code and to generate the executable file pipfit.

The program requires two kinds of input files:

1. One of them contains the options of the run. This is file `<name>.inp`.
2. The other one contains the geometry and energy data. Based on the selected options, the program looks for one (or several) of the following files:

```
main.abn or `<name>.abn` if `ecor =0` or `1`
main.abnll or `<name>.abnll` if `ecor =2` or `4`
```

---

main.abnhl or <name>.abnhl if ecor =3 or 4

If <name>.abn (.abnll, .abnhl) is not found, main.abn (.abnll, .abnhl) will be read as a secondary option. We suggest using the main.abn (.abnll, .abnhl) file for the “whole” data set and the <name>.abn (.abnll, .abnhl) files if data set was modified for specific purposes.

When you have the two the above mentioned input and data files are in the same directory as the executable file pipfit, you should enter:

% ./pipfit

After the execution the program will inquire about the file name, you should type the name part of <name>.inp file

<name>

This will generate the following output files:

<name>.out General output file
<name>.err Contains the selected errors with data points listed in decreasing order
<name>.cff Contains the optimized linear coefficients
<name>.vf0 Contains the fitted and calculated energies for each geometry
<name>.vf1 (optional) Contains the fitted, the calculated, and the diatomic energies
<name>.vf2 (optional) Contains the fitted and calculated energies as well as each term of the polynomial

If ecor is not zero, then procedure of the fitting is slightly different and the names of the files are modified. Please consult section 6.4 for details.
5.2. PIPList program

The package contains an auxiliary program named: PIPList. This based on the code of PIPFit program, but instead of optimizing the coefficients, the program reads them from `<name>.cff (.cffll, .cffhl)` file and evaluates the fitted energies for the requested geometries. This code is located in `src_list/` directory and can be compiled the same way as the PIPFit code, except the request of LAPACK package.

The execution of PIPList program requires three input files (their formats correspond to PIPFit code):

- `main.abn (.abnll, .abnhl)` or `<name>.abn (.abnll, .abnhl)`
- `<name>.inp`
- `<name>.cff (.cffll, .cffhl)`

The coefficients are generated by the PIPFit program with a certain set of fitting parameters (e.g., number of function(s), order of polynomial(s), selected variable(s)), and for the proper application of PIPList program they should not be changed. Nevertheless, other input parameters (e.g., request for statistical analysis of coefficients, borders for energy bins, ranges for the listed errors) can be freely varied.

The PIPList program will generate the same kinds of output files as the PIPFit program.

*If `ecor` is not zero, then procedure of the evaluation is slightly different, and the names of the files are altered, for details see section 6.4.*
6. Options
6.1. Keyword list
In the <name>.inp file the keywords are grouped into four input sections: $molecule, $energy, $fitfunct, and $inout. Each section begins with its name, e.g. $molecule, and is closed by a $end statement. The $molecule section includes molecule-specific keywords. The $fitfunct section contains keywords related to the fitting functions. The $inout section includes keywords related to input and/or output options. And the $energy section includes keywords related to energy thresholds in the fit, as well as parameters for the applied weighing function.

$molecule:  Molecule-specific variables

ityp = 1  A₄ system  (IOrder = 5-12)  usable
       = 2  A₃B system  (IOrder = 5-10)  currently unavailable
       = 3  A₂B₂ system  (IOrder = 5-10)  usable
       = 4  A₂BC system  (IOrder = 5-10)  currently unavailable
       = 5  ABCD system  (IOrder = 5-9)  currently unavailable

iatyp = 56  N₄  Unique identifier for a given four-body system.
         = 60  N₂O₂
         = 64  O₄

$fitfunct:  Variables of fitting function

nfun = 1  One set of primary functions
       = 2  Two sets of primary functions

imgx (x= a,b,c)

       = 1  Morse variable: exp[−(r − ra)/a]  
       = 2  Gaussian variable: exp[−(r − rb)² / b]  
       = 3  Mixed exponential Gaussian (MEG) variable: exp[−(r − ra)/a − (r − rb)² / b]
       = 4  Sech variable: X_i = 1/cosh((r_i − ra)/a_i)  

**ecor** = 0  Original procedure, only one energy set is used.

= 1  The lower-level (LL) energies and higher-level (HL) corrections are fitted separately. But the data are strictly connected, since for each geometry point the HL correction is calculated as a difference between $V_{\text{Total}}$ and $V_{\text{LL}}$. In the (single) data file the total energy and the HL energies must be given. For details see section 6.4.

= 2  Only the LL energies are fitted.

= 3  Only the HL corrections are fitted.

= 4  The LL energies and HL corrections are fitted completely separately, but in the same procedure. This procedure requires two data files. One of them contains the $V_{\text{LL}}$ energies, and the other one contains the $V_{\text{HL}}$ energy corrections. Since the two kinds of input data are not connected in this procedure, the included data sets can contain different number of points. For details see section 6.4.

**iordera**  Degree of the first set of polynomial basis functions ($n_{\text{fun}} = 1$)

See the available IOrder numbers for different kind of systems at `ityp` command (under `$molecule` section).

**iorderb**  Degree of the second set of polynomial basis functions ($n_{\text{fun}} = 2$)

See the available IOrder numbers for different kind of systems at `ityp` command (under `$molecule` section).

**iorderc**  Degree of polynomial basis functions for HL corrections.

See the available IOrder numbers for different kind of systems at `ityp` command (under `$molecule` section).

**ra1-ra6**  The first set of six distances parameters in Å. For further details see Section 4.3

**rb1-rb6**  The second set of six distances parameters in Å for MEG variables. For further details see Section 6.3

**rc1-rc6**  The third set of six distances parameters in Å for $n_{\text{fun}}=2$. For further details see Section 6.3

**rd1-rd6**  The fourth set of six distances parameters in Å ($n_{\text{fun}} = 2$ and MEG variable is applied), for further details see Section 6.3
The fifth set of six distances parameters in Å for HL corrections. For further details see Section 6.3

The sixth set of six distances parameters in Å for HL corrections (MEG variable is applied). For further details see Section 6.3

The first set of six range-separation parameters a(6), in Å. For further details see Section 6.3

The second set of six range-separation parameters b(6) (if MEG variable is applied), the unit is Å². For further details see Section 6.3

The third set of six range-separation parameters c(6), in Å for nfun=2. For further details see Section 6.3

The fourth set of six range-separation parameters d(6) (nfun = 2 and MEG variable is applied), the unit is Å². For further details see Section 6.3

The fifth set of six range-separation parameters e(6), in Å for the HL correction. For further details see Section 6.3

The sixth set of six range-separation parameters f(6) for CASPT2 correction, when MEG variable is applied, the unit is Å². For further details see Section 6.3

Generalized Morse potentials are used for diatomic molecules

Braams- Bowman-Xie approach

Remove the unconnected terms

Remove the 2-body and unconnected terms

No constraints (default)

Force \( C_0 = V_0 \) where \( V_0 \) is the energy of four infinitely separated atoms

User-provided parameter (the energy of four infinitely separated atoms)

Parameter used in the DGELSD subroutine (in LAPACK) to determine the effective rank of matrix \( A \) to minimize \( \| b - A^*x \| \) (default: rcond = 1.0d-15)
$energy$:

- **iwop** = 0: No weighting functions
- = 1: Weighting functions are used, see eq. 13
- = 2: Weighting function (eq. 13) is extended with a geometry dependent factor (see eqs- 14-16)

**Ecr**

Critical energy threshold used in the weighting function (eq. 13), in kcal/mol

**Epow**

The applied power in weighting function (eq. 13)

**Elimit**

High energy threshold to be removed in the fitting, in kcal/mol

**erange1-erange4**

Boundaries of energy to report root mean square deviations (RMSDs) and mean unsigned deviations (MUDs), in kcal/mol

**sael, smiel, smael**

Parameters to control error information printing in the <name>.err file, all in kcal/mol. (The data points with electronic structure energies in the range of smiel < V < smael will be printed in the <name>.err file if the absolute deviations are larger than sael)

$sinout$:

- **icoord** = 0: Internal (Jacob) coordinates
- = 1: Cartesian coordinates
- = 2: Internuclear distances

**ifmt**

- = 0: The electronic structure data set does not include data indices
- = 1: The electronic structure data includes data indices

**iout**

- = 0: Write electronic structure and fitted energies of geometries into <name>.vf0 file
- = 1: Write electronic structure, fitted, and two-body energies for geometries into <name>.vf1 file
- = 2: For test purposes this option writes the energy contributions term by term into <name>.vf2 file (if you use this option, you should use a very small geometry set due to the large file size)
isan = 0  Skip the statistical analysis of the coefficients
= 1  Do a statistical analysis of the coefficients
ired = 0  Do regular fit with all data points
= 1  Generate reduced data set in “main.abn.red” file by deleting every
nred-th data point.
nred  Integer number used to generate reduced data set for test purpose.

6.2. Format of the geometry data files

In the main.abn or <name>.abn files the electronic structure data are collected.
Internal coordinates (icoord = 0), Cartesian coordinates (icoord = 1), and
internuclear distances (icoord = 2) can be used in the data to specify molecular
geometries. And indices of data points may or may not be included (ifmt = 0 or 1). In
general, the first 20 lines of the file are allocated to comments and the program reads the
data set from line 21.

The applied internal coordinates are shown in Scheme 1, which based on the
centers of mass of the two diatomic molecules.

![Scheme 1. Internal coordinates of a four-body system.](image)

The data order of a geometry point is the following for the internal coordinates in a single
line:

ID number (if applied), $r_A$, $r_B$, $d$, $\theta_A$, $\theta_B$, $\phi$, $\Delta E_{Total}$, $\Delta E_{LL}$ (if $\text{ecor} = 1$)
where the distance parameters ($r_A$, $r_B$, and $d$) should be in Å, the angles ($\theta_A$, $\theta_B$, and $\phi$)
should be in degrees, and the relative energy ($\Delta E_{Total}$) should be in kcal/mol. In this later
case the reference, zero, energy is the energy of $N_2(r_e) + N_2(r_e)$ for $N_4$ system.

When the data are organized in Cartesian coordinates, then each geometry point
occupies five lines as a block:
ID number (if applied), $\Delta E_{\text{Total}}$, $\Delta E_{\text{LL}}$ (if $\text{ecor}=1$)

$X_1, Y_1, Z_1$
$X_2, Y_2, Z_2$
$X_3, Y_3, Z_3$
$X_4, Y_4, Z_4$

where the relative energy ($\Delta E_{\text{Total}}$) should be in kcal/mol and the 12 Cartesian coordinates should be in Å and the indices are the atom numbers.

For either data format (internal or Cartesian coordinates) the program calculates six internuclear distances ($r_i$, where $r_1 = r_{12}$, $r_2 = r_{13}$, $r_3 = r_{14}$, $r_4 = r_{23}$, $r_5 = r_{24}$, and $r_6 = r_{34}$). When an A$_4$ (homonuclear) system is calculated, the order of the atoms ($i = 1$-$6$) does not matter. But for heteronuclear systems, e.g., A$_2$B$_2$, these indices of individual atom pairs must to be consistent with $ra_1, rb_1, rc_1, rd_1, re_1, rf_1, arng_1, brng_1, crng_1, drng_1, erng_1,$ and $frng_1$ parameters in the input file (see Section 6.1) as well as with the corresponding indices of pairwise potentials. For example, for the N$_2$O$_2$ heteronuclear system, there are six internuclear distances, where the only O-O distance and the only N-N distance are unique, but the four N-O distances are not. By our convention, the first two atoms in the Cartesian coordinate lists are oxygens and the second two atoms are nitrogens. Therefore, the O-O bond length is $r_1$ ($i = 1$), the N-N bond length is $r_6$ ($i = 6$), and $i = 2$-$5$ denote the four N-O distances related by permutationally invariant polynomial(s). The diatomic potential are hard coded with this convention.

The parameters of the many-body terms are provided by the user. When a new four-body system is set up, the user has to decide that which parameter set belongs to which internuclear distance. For convenience, the user can use comments after the character # in the input file <name>.inp to indicate information about the convention being used. The $ra_1, rb_1, ..., rf_1$ parameters are the “equilibrium distances” in the equations of the selected variables. If the user sets up 1.098 Å for parameter $ra_1$ (as compared to the correct the O-O bond length of 1.208 Å), the results will be wrong. The
above description of equilibrium distance parameters is also applied to the non-linear parameters \((\text{arng}_i, \text{brng}_i, \text{frng}_i)\).

The order of the data is as follows (in a single line):

\[
\text{ID number (if applied), } r_1, r_2, r_3, r_4, r_5, r_6, \Delta E_{\text{Total}}, \Delta E_{\text{LL}} \text{ (if } e_{\text{cor}}=1)\]

6.3. Parameters of \(X_i\) variables

Depending on the kind of the four-body system \((\text{ityp})\), the program requires different numbers of \(r_{a_i}, r_{b_i}, r_{c_i}, r_{d_i}, r_{e_i}, r_{f_i}, \text{arng}_i, \text{brng}_i, \text{crng}_i, \text{drng}_i, \text{erng}_i, \text{frng}_i\) parameters. For an \(A_4\) (homonuclear) system, values of \(r_{a_1} (r_{b_1}, \text{etc}...)\) and \(\text{arng}_1 (\text{brng}_1, \text{etc}...)\) must be given, and the program repeats their values for \(i = 2-5\). For an \(A_2B_2\) (heteronuclear) system, \(r_{a_1} (r_{b_1}, \text{etc}...)\) and \(\text{arng}_i (\text{brng}_i, \text{etc}...) (i = 1, 2, \text{and} 6)\) entries must to be given and the parameters of \(i = 2\) will be repeated for the cases of \(i = 3-5\). For example, in \(\text{N}_2\text{O}_2\) system (see Section 6.2.) \(i = 1\) corresponds to O-O bond, \(i = 2-5\) are the four N-O bonds, and \(i = 6\) covers the N-N bond, for this \(\text{N}_2\text{O}_2\) system an example input file can be seen in section 8.

6.4. Separate fitting of \(V_{\text{LL}}\) and \(V_{\text{HL}}\) potentials

For separate fitting of a set of lower-level energies on a dense grid and a set of energy corrections on a sparser grid, four types of procedure were implemented:

1) If \(e_{\text{cor}}=1\), then the file \(<\text{name}.\text{abn}\text{ or main.\text{abn}\text{ is read}}. In this case the file must contain the \(\Delta E_{\text{Total}}\) energies as well as the \(\Delta E_{\text{LL}}\) energies (see section 6.2.). In the file \(<\text{name}.\text{inp}\text{ the option nfun can be 1 or 2 and it is applied for the fit of the LL energies. For the HL corrections only one function is used. Therefore, iordera and iorderb (if nfun=2) belong to the LL part of the fit, while iorderc belong to the HL correction part of the fit. Depending on the selected \(imgx\) option several \(rx_i\) and \(x_{\text{rng}}_i (x=a,b,c,d,e,f)\) parameters are required. For example if \(e_{\text{cor}}=1, nfun=2 \text{ imga}=3, \text{imgb}=3, \text{and imgc}=1, then the sets ra, arrng, rb, and brng belong to the first MEG function (iordera) and the sets rc, crng, rd, and drng belong to the second MEG function (iorderb). Both of the functions are used in the fit of the LL
energies. And finally, the sets re and errng belong to the third Morse function (iorderc) for the fit of HL correction.

Another example when ecor=1, imga=1, and imgc=3, then the sets “a” belongs to the first Morse function (iordera). It fits the LL energies. And the sets re, errng, fd, and frng belongs to the a MEG function (iorderc), which fits the HL correction.

After the fitting procedure the program will generate two coefficient files, <name>.cffll and <name>.cffhl. The error calculations correspond to $\Delta E_{\text{Total}}$, $\Delta E_{\text{LL}}$ as well as $\Delta E_{\text{HL}}$ energies.

2) If ecor=2, then the file(s) <name>.abnll or main.abnll is read. The file .abnll must contain the $\Delta E_{\text{LL}}$ energies (see section 6.2.). In this case, only the $\Delta E_{\text{LL}}$ energies are fitted.

3) If ecor=3, then the file(s) <name>.abnhl or main.abnhl is read. The file .abnhl must contain the $\Delta E_{\text{HL}}$ energies (see section 6.2.). In this case, only the $\Delta E_{\text{HL}}$ corrections are fitted, and the following parameters are required: iordrc, imgc, re, errng (if imgc=3, then rf and frng as addition). 

4) If ecor=4, then the file(s) <name>.abnll or main.abnll and <name>.abnhl or main.abnhl are read. In this case the file .abnll must contain the $\Delta E_{\text{LL}}$ energies (see section 6.2.). And the file .abnhl must contain the $\Delta E_{\text{HL}}$ correlation energies (see section 6.2.). In the file <name>.inp the option nfun can be 2, and this option is applied for the fit of LL energies. Hence, depending on the selected imgx options several rx and xrng (x=a,b, etc...) parameters are required.

For example if ecor=3, nfun=2, imga=1, imga=3 and imgc=2, then the sets ra and arrng belong to the first Morse function (iordera) and the sets rc, crng, rd, and drng belong to the second MEG function (iorderb). These functions fit the LL energies, and the third (Gauss) function will fit the HL correction. The sets re and errng belong to this third function (iorderc).
After the fitting procedure the program will generate two coefficient files 
<name>.cffl1 and <name>.cffh1. The error calculations correspond to the 
two separate fittings. And instead of files <name>.vf0, <name>.vf1, or 
<name>.vf2 the program will generate two files, namely the files 
<name>.vfll and <name>.vfhl. These files contain the ID number, the 
input energy, the calculated energy, and the diatomic contributions for each point. 
And the file <name>.err is not provided for this option.

In $\text{ecor} = 1, 2, 3,$ and 4 cases the $\text{ityp}=2$ option can be used if the diatomic potentials 
from LL calculation are available besides the diatomic potentials from HL calculation. 
Currently the potential energy curves of N$_2$, O$_2$, and NO molecules are available in the 
program from CASSCF calculations and scaled external corrected CASPT2 calculations. 
They are located in $\text{ev2gm2.f}$ file. And the $V_{\text{PA,HL}}$ correction data (for $\text{ecor} = 1, 3,$ 
and 4) is calculated as energy difference of $V_{\text{PA,HL}}$ and $V_{\text{PA,LL}}$. 

7. Program structure

The src/ (and src_list/) directory contains the following Fortran files:

- bmx2b1.f
- bmx2b3.f
- bmxnc1.f
- bmxnc3.f
- call_dgelsd.f
- calrmsd.f
- change_case.f
- coordconvt.f
- detcoef.f
- ev2b.f
- ev2gm2.f
- evbas.f
- evmono1.f
- evmono3.f
- evms.f
- evpoly1.f
- evpoly3.f
- frmbm.f
- frmbmp.f
- frmbmx2b.f
- frmbmxnc.f
- matvect.f
- numb.f
- pesfit.f
- prmt.f
- readdat.f
- readinp.f
- sort.f
- tsort.f
Some important subroutines are discussed below.

**pesfit.f:**
The *pesfit* is the main program of PIPFit to perform a weighted least-squares fit with permutationally invariant polynomials. *pesfit* calls the *read* subroutine to read in all of the parameters for the fitting as well as the geometries and electronic structure energies. The *frmbm* subroutine is then called to form basis function matrix. Linear coefficients are optimized by calling *detcoeff* subroutine. The fitted energies are evaluated by calling *matvect* subroutine, and the error analysis is carried out by *calrmsd* and *sort* subroutines.

**readinp.f:**
The *readinp* subroutine reads in all of the parameters for the least-squares fit from `<name>.inp`.

**readdat.f:**
The *readdat* subroutine reads in all of the geometries and electronic structure energies from `main.abn` or `<name>.abn` (for ecor=2,3, or 4 .abnll or/and .abnhl files). The geometries can be provided with internal coordinates (icoord = 0), Cartesian coordinates (icoord = 1), or internuclear distances (icoord = 2). The six internal coordinates are converted to Cartesian coordinates by calling the *coordconvt* subroutine.

**frmbm.f:**
The *frmbm* subroutine forms basis function matrix $B$. The *frmbm* subroutine calls the *evbas* subroutine to evaluate permutationally invariant polynomial (PIP) basis functions at a given geometry. The unconnected terms are removed by calling the *frmbmxnc* subroutine if imethd = 1 is specified. The two-body terms and unconnected terms are removed by calling the *frmbmx2b* subroutine if imethd = 2 is specified.
**detcoeff.f:**
The `detcoeff` subroutine optimizes the linear coefficients by solving normal equation. The `frmbmp` subroutine is called by the `detcoeff` subroutine to obtain normal equation. And the normal equation is solved by calling the `dgelsd` subroutine from LAPACK library.

**matvect.f:**
The `matvect` subroutine does multiplication of a matrix and a vector. It is called by `pesfit` to calculate the fitted energies with basis function matrix and optimized coefficients.

**calrmsd.f:**
The `calrmsd` subroutine calculates and reports root-mean-square deviation and mean unsigned deviation between fitted and calculated energies.

**sort.f:**
The `sort` subroutine sorts the absolute deviations between fitted and calculated energies in decreasing order. The geometries, fitted energies, and electronic structure energies of data points with deviation larger than preselected threshold will be printed in the `.err` file.

**ev2b.f:**
The `ev2b` subroutine evaluates the two-body interactions if `imethd = 2` is chosen. The two-body interactions are calculated by calling the `ev2gm2` subroutine.

**ev2gm2.f**
The `ev2gm2` subroutine contains the diatomic potentials. Currently the pairwise potentials of N$_2$, O$_2$, and NO are available based on CASSCF calculations and scaled external corrected CASPT2 calculations.
evbas.f:
The *evbas* subroutine calculates basis functions for a given geometry by calling the *evpoly1* or *evpoly3* subroutines which calculates the permutationally invariant polynomials (PIP) from monomials, which are evaluated with the *evmono1* or *evmono3* subroutines based on the *evms* subroutine.

frmbmxnc.f:
The *frmbmxnc* subroutine removes the unconnected terms in the PIP basis functions by calling the *bmxnc1* and *bmxnc3* subroutines.

frmbmx2b.f:
The *frmbmx2b* subroutine removes the unconnected terms in the PIP basis functions by calling the *bmx2b1* and *bmx2b3* subroutines.
8. Test inputs

The PIPFit program requires an input file (<name>.inp) containing all the parameters for the fitting and a data file (main.abn or <name>.inp) containing the electronic structure energies and molecular geometries.

In the input file, lines starting with “#” are treated as comments and will not be read by PIPFit. Specific parameters are called by keywords to do the fit, please refer to Section 4 for the detailed description of keywords. An input file, which performs PES fit of N₂O₂ with 10th order permutationally invariant polynomials of mixed exponential Gaussian variables where two-body and unconnected terms are removed, is shown in the next page.

The first 20 lines of the data file (main.abn or <name>.inp) are reserved for comments. Both internal coordinate (icoord = 0), Cartesian coordinate (icoord = 1), or internuclear distances (icoord = 2) can be used in the data to specify molecular geometries. Two sample data files, one for N₄ in internal coordinate and one for N₂O₂ in Cartesian coordinate, are given in this section.

More examples are given in the test/ directory.
Sample input file to fit $N_2O_2$ PES is shown below:

```
#input an example for comment line (#) it is not read by PIPFit
$molecule
ityp=3 iatyp=60
ra1=1.208 ra2=1.1508 ra6=1.098
rb1=1.208 rb2=1.1508 rb6=1.098
arng1=0.87 arng2=0.610 arng6=0.920
brng1=1.40 brng2=1.200 brng6=1.550
$end

$fitfunct
nfun=1
imga=3
iordera=10
iorderb=5
imethod=2
pairw=1
icop=0
rcond=1.0d-50
$end

$inout
ifmt=1
icoord=1
iout=1
isan=1
$end

$energy
iwop=1
elimit=2000.0
ecr=100.0
epow=2.0
erange1=100.0 erange2=200.0 erange3=350.0 erange4=1000.0
smiel=-50.0 smael=500.0 sael=10.0
$end

More examples can be find in the test/ directory.
Sample data file for $\text{N}_2$ in internal coordinate is shown below:

N4 ab initio electronic structure data  
*** Full data set: All points INCLUDED ***
16435 total data points, updated 12/05/2013, all energies $\leq$ 2000 kcal/mol  
CASPT2/maug-cc-pVTZ (12e/12o,g4,shift=0.30)

Data set information: [All point index ranges are inclusive]
points 1 through 15365 (15365 points): $\text{N}_2$+$\text{N}_2$ scans: T,T2,H,Z,I,X,A
points 15366 through 16382 (1017 points): $\text{N}_3$+$\text{N}$ scans:
  $\text{N}_3$+2$\text{N}$ linear: L2,L2+,L2-; 2$\text{N}_3$+2$\text{N}$ bent: L3,L3+,L3-; 4$\text{N}_3$+4$\text{N}$ bent: L4,L4-,L4+
points 16383 through 16383 (1 point): extra linear $\text{N}_3$+$\text{N}$ point
points 16384 through 16386 (3 points): stationary points
points 16387 through 16391 (5 points): random points added for testing
points 16392 through 16421 (30 points): points from 4 smooth paths for testing
points 16422 through 16435 (14 points): $\text{N}_2$+$\text{N}_2$ scans: extra large-separation points

Use the following conversion factors when manipulating this data:
[From the CRC Handbook of Chemistry and Physics, 94th edition, 2013-2014, Section 1, "CODATA recommended values of the fundamental physical constants: 2010"]
1 bohr = 0.529177210920 d A; 1 hartree = 627.509474 d kcal/mol; 1 kcal/mol = 0.159360144 d hartree
1 hartree/bohr = 1185.82105 d kcal mol$^{-1}$ A$^{-1}$; 1 kcal mol$^{-1}$ A$^{-1}$ = 0.843297564 d hartree/bohr

The first data point is listed on line 21 of this file.

<table>
<thead>
<tr>
<th>id</th>
<th>id, rA [Å], rB [Å], d [Å], thetaA [degrees], thetaB [degrees], phi [degrees], E_ref [kcal/mol]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0980000 1.2000000 1.0000000 0.0000000 90.0000000 0.0000000 1971.5982651</td>
</tr>
<tr>
<td>2</td>
<td>1.0980000 1.3000000 1.0000000 0.0000000 90.0000000 0.0000000 1587.2724321</td>
</tr>
<tr>
<td>3</td>
<td>1.0980000 1.4000000 1.0000000 0.0000000 90.0000000 0.0000000 1280.7679395</td>
</tr>
<tr>
<td>4</td>
<td>1.0980000 1.5000000 1.0000000 0.0000000 90.0000000 0.0000000 1034.6931742</td>
</tr>
<tr>
<td>5</td>
<td>1.0980000 1.6000000 1.0000000 0.0000000 90.0000000 0.0000000 840.3957214</td>
</tr>
<tr>
<td>6</td>
<td>1.0980000 1.7000000 1.0000000 0.0000000 90.0000000 0.0000000 689.8470370</td>
</tr>
</tbody>
</table>

……
A sample data file for N$_2$O$_2$ in Cartesian coordinates is shown below:

Points of N2O2 surface
last modified 07/01/2014
***
Dissociation energies of triplet N2O2 system (N2, O2, N2O, NO2, NO)
MS- CASPT2(g4,shift=0.30,2s in core)/DW-SA-CASCF(maug-ccpVTZ,RAS algorithm,14e/12o,3 or 2 states,dynw=9.0707)
Each MS- CASPT2 energy was modified by DSEC correction.
Ref. energy is N2(re) + O2(re).
***
Format of the dataset:
1. # of the point, DE (in kcal/mol)
2. O1= x1, y1, z1  (in Ang)
3. O2= x2, y2, z2  (in Ang)
4. N1= x3, y3, z3  (in Ang)
5. N2= x4, y4, z4  (in Ang)
***
Conversion factors:
1 hartree = 627.509474 kcal/mol
1 bohr = 0.52917721092 angstrom
1 hartree/bohr = 1185.82105 kcal mol^-1 angstrom^-1
***
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1756.2463</td>
<td>-0.104000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>-0.104000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>1.104000</td>
</tr>
<tr>
<td>0.650000</td>
<td>0.000000</td>
<td>-0.500000</td>
</tr>
<tr>
<td>-0.650000</td>
<td>0.000000</td>
<td>-0.500000</td>
</tr>
</tbody>
</table>
........
9. Revision history

version 2015 (July 28, 2015)
Authors: Ke R. Yang, Zoltan Varga, and Donald G. Truhlar.

Version 2015 is the first distributed version of PIPFit.