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 29, 2015

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

1. User agreement

PIPFit is a licensed program, and the use of this program implies acceptance of the terms of the license, which are repeated here for convenience:

A. No user or site will redistribute the source code or executable code to a third party in original or modified form without written permission of the principal investigator (Donald G. Truhlar). A license does not entitle the licensee to relicense the code or distribute it in original or modified form to parties not covered by the license. The licensee has no ownership rights in the *PIPFit* software or in any copyrights for the *PIPFit* software or documentation through this license. A user license covers the work of a single research group and the code may be shared and disseminated within a group without requiring permission. Site licenses are available with a different license (see the *PIPFit* home page).

B. Publications resulting from using this package will cite the corresponding program. The required references are given in the documentation.

C. No guarantee is made that this program is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within the program.

D. No consulting or maintenance services are guaranteed or implied

2. Citation

Publications including work performed with *PIPFit* should cite the software, for example, as follows:

Ke R. Yang, Zoltan Varga, and Donald G. Truhlar PIPFit – version 2015, University of Minnesota, Minneapolis, 2015. 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

- B. J. Braams and J. M. Bowman *Int. Rev. Phys. Chem.* **28**, 577 (2009).
- Z. Xie and J. M. Bowman *J. Chem. Theory Comput.* 6, 26 (2010).
- Y. Paukku, K. R. Yang, Z. Varga, and D. G. Truhlar *J. Chem. Phys.*, 139, 044309 (2013).
- J. D. Bender, P. Valentini, I. Nompelis, Y. Paukku, Z. Varga, D. G. Truhlar, T. Schwartzentruber, G. V. Candler, *J. Chem. Phys.,* submitted.

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:

$$r_1 = r_{12}, \qquad r_4 = r_{23},$$

$$r_2 = r_{13}, \qquad r_5 = r_{24},$$

$$r_3 = r_{14}, \qquad r_6 = r_{34},$$

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 ($X_i = \exp(-(r_i - r_a)/a_i)$),
- Gaussian variables ($X_i = \exp(-(r_i - r_a)^2/a_i)$), or
- Mixed exponential Gaussian (MEG) variables ($X_i = \exp[-(r_i - r_a)/a_i - (r_i - r_b)^2/b_i]$), or
- Hyperbolic secant (Sech) variables ($X_i = 1/\cosh((r_i - r_a)/a_i)$)

The PES is expanded in a Taylor series of $X_i(r_i)$ as

$$V(r_1, r_2, r_3, r_4, r_5, r_6) = \sum_{n_1+n_2+n_3+n_4+n_5+n_6=0}^{k} C_{n_1n_2n_3n_4n_5n_6} X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}, \qquad (1)$$

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, $S[X_1^{n1} X_2^{n2} X_3^{n3} X_4^{n4} X_5^{n5} X_6^{n6}]$, as proposed by Braams, Bowman, and Xie.[1,2] Thus eq. 1 is written as

[1] B. J. Braams and J. M. Bowman, Int. Rev. Phys. Chem. **28**, 577 (2009).

[2] Z. Xie and J. M. Bowman, J. Chem. Theory Comput. **6**, 26 (2010).

$$V(r_1,r_2,r_3,r_4,r_5,r_6)=\sum_{n_1+n_2+n_3+n_4+n_5+n_6=0}^{k} D_{n_1n_2n_3n_4n_5n_6} S[X_1^{n1}X_2^{n_2}X_3^{n_3}X_4^{n_4}X_5^{n_5}X_6^{n_6}]\cdot \quad (2)$$

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

$$V(r_1,r_2,r_3,r_4,r_5,r_6)=\sum_{\substack{\text{connected,}\\ n_1+n_2+n_3+n_4+n_5+n_6=0}}^{k} {}'D_{n_1n_2n_3n_4n_5n_6} S[X_1^{n1}X_2^{n_2}X_3^{n_3}X_4^{n_4}X_5^{n_5}X_6^{n_6}]\cdot \quad (3)$$

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 $V_{\mathrm{PA}}(r_i)$ 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:

$$\begin{aligned}V(r_1,r_2,r_3,r_4,r_5,r_6)&=V_0+\sum_{i=1}^{6}V_{\mathrm{PA}}(r_i)+\\ &+\sum_{\substack{\text{connected,}\\ n_1+n_2+n_3+n_4+n_5+n_6=0}}^{k} {}'D_{n_1n_2n_3n_4n_5n_6} S[X_1^{n1}X_2^{n_2}X_3^{n_3}X_4^{n_4}X_5^{n_5}X_6^{n_6}]\end{aligned} \quad (4)$$

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 N_2, O_2, and NO have been fitted the following generalized Morse potential:

$$V_{\mathrm{PA}}(r) = D_e[1-\exp(-f(r)(r-r_e)]^2 - D_e, \quad (5)$$

[3] Y. Paukku, K. R. Yang, Z. Varga, and D. G. Truhlar, J. Chem. Phys. **139**, 044309 (2013).

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:

$$f(r) = \sum_{i=0}^{6} a_i y^i \quad . \tag{6}$$

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

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

$$\begin{aligned} V(r_1,r_2,r_3,r_4,r_5,r_6) &= D_{000000} \\ &+ \sum_{n_1+n_2+n_3+n_4+n_5+n_6=1}^{k_a} D^a_{n_1n_2n_3n_4n_5n_6} S[X_1^{n1} X_2^{n2} X_3^{n3} X_4^{n4} X_5^{n5} X_6^{n6}]' \\ &+ \sum_{n_1+n_2+n_3+n_4+n_5+n_6=1}^{k_b} D^b_{n_1n_2n_3n_4n_5n_6} S[Y_1^{n1} Y_2^{n2} Y_3^{n3} Y_4^{n4} Y_5^{n5} Y_6^{n6}] \end{aligned} \tag{2′}$$

$$\begin{aligned} V(r_1,r_2,r_3,r_4,r_5,r_6) &= D_{000000} \\ &+ \sum_{\substack{\text{connected,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}}^{k_a}{}' D^a_{n_1n_2n_3n_4n_5n_6} S[X_1^{n1} X_2^{n2} X_3^{n3} X_4^{n4} X_5^{n5} X_6^{n6}]' \\ &+ \sum_{\substack{\text{connected,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}}^{k_b}{}' D^b_{n_1n_2n_3n_4n_5n_6} S[Y_1^{n1} Y_2^{n2} Y_3^{n3} Y_4^{n4} Y_5^{n5} Y_6^{n6}] \end{aligned} \tag{3′}$$

and

$$\begin{aligned} V(r_1,r_2,r_3,r_4,r_5,r_6) &= V_0 + \sum_{i=1}^{6} V_{\mathrm{PA}}(r_i) + \\ &+ \sum_{\substack{\text{connected,} \\ n_1+n_2+n_3+n_4+n_5+n_6=0}}^{k_a} {}'D^a_{n_1n_2n_3n_4n_5n_6} S[X_1^{n1} X_2^{n2} X_3^{n3} X_4^{n4} X_5^{n5} X_6^{n6}] \cdot \\ &+ \sum_{\substack{\text{connected,} \\ n_1+n_2+n_3+n_4+n_5+n_6=0}}^{k_b} {}'D^b_{n_1n_2n_3n_4n_5n_6} S[Y_1^{n1} Y_2^{n2} Y_3^{n3} Y_4^{n4} Y_5^{n5} Y_6^{n6}] \end{aligned} \tag{4′}$$

In eqs. 2′-4′, a and b denote two different sets of permutationally invariant polynomial basis functions with the orders of k_a 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_{CASPT2}-E_{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_{Total}). The surface of LL energies is usually as rugged as the surface of V_{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 $\mathbf{x}_i$ the geometry of data point i, by b_{ij} the basis function j at geometry $\mathbf{x}_i$, and by c_j the linear coefficient ($j = 1, \ldots, m$). The determination of linear coefficients is equivalent to minimizing the functional,

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

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

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

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

$$\mathbf{B}^{\mathrm{T}}\mathbf{WBc} = \mathbf{B}^{\mathrm{T}}\mathbf{WV} , \tag{12}$$

where $^{\mathrm{T}}$ denotes the matrix transpose operation, and

$$\mathbf{B} = \begin{bmatrix} b_1(\mathbf{x}_1) & b_2(\mathbf{x}_1) & \cdots & b_m(\mathbf{x}_1) \\ b_1(\mathbf{x}_2) & b_2(\mathbf{x}_2) & \cdots & b_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(\mathbf{x}_n) & b_2(\mathbf{x}_n) & \cdots & b_m(\mathbf{x}_n) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix}, \quad \text{and} \quad ,$$

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

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:

$$W_{ii} = W_{ii}^{en} = \begin{cases} 1 & \text{for } V_i \le E_{cr} \\ \left(E_{cr} / V_i\right)^{Epow} & \text{for } V_i > E_{cr} \end{cases} , \tag{13}$$

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 *Epow* 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:

$$W_{ii} = W_{ii}^{en} W_{ii}^{geom} , \tag{14}$$

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

$$d_{i,k} = \sqrt{\frac{1}{6}\sum_{j=1}^{6}\left(\frac{(r'_{i,j} - r'_{k,j})}{(r'_{i,j} + r'_{k,j})}\right)^2} \ . \tag{15}$$

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, 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:

$$d_i = \frac{\sum_{k=1}^{n} d_{i,k}}{n} , \tag{16}$$

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

$$W_{ii}^{geom} = \frac{d_i}{\max(d_i)} \cdot \tag{17}$$

5. Installation and general usage

5.1. PIPFit program

The PIPFit sofware 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/`	PIPList 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

[4] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J., J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *{LAPACK} Users' Guide*, 3rd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

`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 key words related to energy thresholds in the fit, as well as parameters for the applied weighing function.

$molecule:		Molecule-specific variables		
ityp	= 1	A_4 system	(IOrder = 5-12)	usable
	= 2	A_3B system	(IOrder = 5-10)	currently unavailable
	= 3	A_2B_2 system	(IOrder = 5-10)	usable
	= 4	A_2BC system	(IOrder = 5-10)	currently unavailable
	= 5	ABCD system	(IOrder = 5-9)	currently unavailable
iatyp	= 56 N_4	Unique identifier for a given four-body system.		
	= 60 N_2O_2			
	= 64 O_4			

$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: $\exp[-(r-r_a)/a]$
	= 2	Gaussian variable: $\exp[-(r-r_b)^2/b]$
	= 3	Mixed exponential Gaussian (MEG) variable: $\exp[-(r-r_a)/a-(r-r_b)^2/b]$
	= 4	Sech variable: $X_i = 1/\cosh((r_i - r_a)/a_i)$

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_{Total} and V_{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_{LL} energies, and the other one contains the V_{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 (`nfun` = 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 (`nfun` = 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 `nfun`=2. For further details see Section 6.3

rd1-rd6 The fourth set of six distances parameters in Å (`nfun` = 2 and MEG variable is applied), for further details see Section 6.3

re1-re6 The fifth set of six distances parameters in Å for HL corrections. For further details see Section 6.3

rf1-rf6 The sixth set of six distances parameters in Å for HL corrections (MEG variable is applied). For further details see Section 6.3

arng1-arng6 The first set of six range-separation parameters *a*(6), in Å. For further details see Section 6.3

brng1-brng6 The second set of six range-separation parameters *b*(6) (if MEG variable is applied), the unit is Å^2. For further details see Section 6.3

crng1-crng6 The third set of six range-separation parameters *c*(6), in Å for `nfun`=2 . For further details see Section 6.3

drng1-drng6 The fourth set of six range-separation parameters *d*(6) (`nfun` = 2 and MEG variable is applied), the unit is Å^2. For further details see Section 6.3

erng1-erng6 The fifth set of six range-separation parameters *e*(6), in Å for the HL correction. For further details see Section 6.3

frng1-frng6 The sixth set of six range-separation parameters *f*(6) for CASPT2 correction, when MEG variable is applied, the unit is Å^2. For further details see Section 6.3

pairw = 1 Generalized Morse potentials are used for diatomic molecules

imethod = 0 Braams- Bowman-Xie approach

= 1 Remove the unconnected terms

= 2 Remove the 2-body and unconnected terms

icop = 0 No constraints (default)

= 1 Force $C_0 = V_0$ where V_0 is the energy of four infinitely separated atoms

V_0 User-provided parameter (the energy of four infinitely separated atoms)

rcond 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: Energy thresholds section

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**)

$inout: Input and output section

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

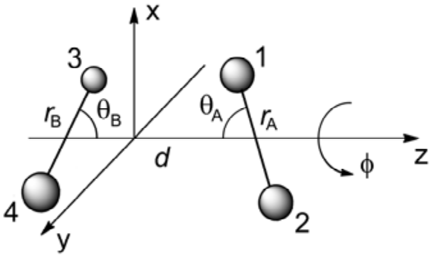
= 2 For test purposes this option writes the energy contributions term by term into `<name>.vf2` file (f you use this option, you should use a *very* small geometry set due to the large file size)

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.

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, θ_A, θ_B, ϕ, ΔE_{Total}, ΔE_{LL} (if `ecor`=1)

where the distance parameters (r_A, r_B, and d) should be in Å, the angles (θ_A, θ_B, and ϕ) should be in degrees, and the relative energy (ΔE_{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), $\Delta E_{\mathrm{Total}}$, ΔE_{LL} (if `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 ($\Delta E_{\mathrm{Total}}$) 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_2B_2, these indices of individual atom pairs must to be consistent with `ra`$_\mathrm{i}$, `rb`$_\mathrm{i}$, `rc`$_\mathrm{i}$, `rd`$_\mathrm{i}$, `re`$_\mathrm{i}$, `rf`$_\mathrm{i}$, `arng`$_\mathrm{i}$, `brng`$_\mathrm{i}$, `crng`$_\mathrm{i}$, `drng`$_\mathrm{i}$, `erng`$_\mathrm{i}$, and `frng`$_\mathrm{i}$ parameters in the input file (see Section 6.1) as well as with the corresponding indices of pairwise potentials. For example, for the N_2O_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`$_\mathrm{i}$, `rb`$_\mathrm{i}$, …, `rf`$_\mathrm{i}$ 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 (`arng`$_i$, `brng`$_i$, `frng`$_i$).

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

ID number (if applied), r_1, r_2, r_3, r_4, r_5, r_6, $\Delta E_{\mathrm{Total}}$, ΔE_{LL} (if `ecor`=1)

6.3. Parameters of X_i variables

Depending on the kind of the four-body system (`ityp`), the program requires different numbers of `ra`$_i$, `rb`$_i$, `rc`$_i$, `rd`$_i$, `re`$_i$, `rf`$_i$, `arng`$_i$, `brng`$_i$, `crng`$_i$, `drng`$_i$, `erng`$_i$, `frng`$_i$ parameters. For an A_4 (homonuclear) system, values of `ra`$_1$ (`rb`$_1$, etc…) and `arng`$_1$ (`brng`$_1$, etc…) must be given, and the program repeats their values for *i* = 2-5. For an A_2B_2 (heteronuclear) system, `ra`$_i$ (`rb`$_1$, etc…) and `arng`$_i$ (`brng`$_1$, etc…) (*i* = 1,2, 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 N_2O_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 N_2O_2 system an example input file can be seen in section 8.

6.4. Separate fitting of V_{LL} and V_{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 `ecor=1`, then the file `<name>.abn` or `main.abn` is read. In this case the file must contain the $\Delta E_{\mathrm{Total}}$ energies as well as the ΔE_{LL} energies (see section 6.2.). In the file `<name>.inp` 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 `xrng`$_i$ (`x=a,b,c,d,e,f`) parameters are required. For example if `ecor`=1, `nfun`=2 `imga`=3, `imgb`=3, 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 `erng` 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, erng, 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 $\Delta E_{\mathrm{Total}}$, ΔE_{LL} as well as ΔE_{HL} energies.

2) If `ecor=2`, then the file(s) `<name>.abnll` or `main.abnll` is read. The file `.abnll` must contain the ΔE_{LL} energies (see section 6.2.). In this case, only the ΔE_{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 ΔE_{HL} energies (see section 6.2.). In this case, only the ΔE_{HL} corrections are fitted, and the following parameters are required: `iordrc, imgc, re, erng` (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 ΔE_{LL} energies (see section 6.2.). And the file `.abnhl` must contain the ΔE_{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 $\mathtt{rx}_{\mathtt{i}}$ and $\mathtt{xrng}_{\mathtt{i}}$ (`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>.cffll` and `<name>.cffhl`. 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 `ecor` =1, 2, 3, and 4 cases the `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 `ev2gm2.f` file. And the $V_{\mathrm{PA,HL}}$ correction data (for `ecor` =1, 3, and 4) is calculated as energy difference of $V_{\mathrm{PA,HL}}$ and $V_{\mathrm{PA,LL}}$.

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 N_2O_2 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 N_4 in internal coordinate and one for N_2O_2 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 N_2 in internal coordinate is shown blow:

```
N4 ab initio electronic structure data
  *** Full data set: All points INCLUDED ***
  16435 total data points, updated 12/05/2013, all energies <= 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): N2+N2 scans: T,T2,H,Z,I,X,A
  points  15366 through  16382 ( 1017 points): N3+ N scans:
    N3+2N linear: L2,L2+,L2-; 2N3+2N bent: L3,L3+,L3-; 4N3+4N bent: L4,L4-,L4+
  points  16383 through  16383 (    1 point ): extra linear N3+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): N2+N2 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= 0.52917721092d0 A; 1 hartree= 627.509474d0 kcal/mol; 1 kcal/mol= 0.159360144d-2 hartree
  1 hartree/bohr= 1185.82105d0 kcal mol^-1 A^-1; 1 kcal mol^-1 A^-1= 0.843297564d-3 hartree/bohr
The first data point is listed on line 21 of this file.
id, rA [A], rB [A], d [A], thetaA [degrees], thetaB [degrees], phi [degrees], E_ref [kcal/mol]
      1   1.0980000   1.2000000   1.0000000   0.0000000   90.0000000   0.0000000  1971.5982651
      2   1.0980000   1.3000000   1.0000000   0.0000000   90.0000000   0.0000000  1587.2724321
      3   1.0980000   1.4000000   1.0000000   0.0000000   90.0000000   0.0000000  1280.7679395
      4   1.0980000   1.5000000   1.0000000   0.0000000   90.0000000   0.0000000  1034.6931742
      5   1.0980000   1.6000000   1.0000000   0.0000000   90.0000000   0.0000000   840.3957214
      6   1.0980000   1.7000000   1.0000000   0.0000000   90.0000000   0.0000000   689.8470370
……
```

A sample data file for N_2O_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
***
     1    1756.2463
    0.000000    0.000000   -0.104000
    0.000000    0.000000    1.104000
    0.650000    0.000000   -0.500000
   -0.650000    0.000000   -0.500000
……..
```

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.