

MANUAL

PIPFit 2022

**A program to fit potential energy surfaces
using permutationally invariant polynomials**

Ke R. Yang, Zoltan Varga, Kelsey A. Parker, Yinan Shu,
and Donald G. Truhlar

*Department of Chemistry, Chemical Theory Center, and Supercomputing Institute
University of Minnesota, Minneapolis, Minnesota 55455*

Date of most recent program change: February 13, 2022

Date of most recent manual update: February 13, 2022

Abstract. The *PIPFit 2022* program can be used to develop analytic representations of potential energy surfaces for three-body and four-body systems. A weighted least-squares fit is performed with permutationally invariant polynomials (PIPs) whose variables are Morse-like bond functions, Gaussians, mixed exponential–Gaussians (MEGs), or hyperbolic secant variables.

Three kinds of fit can be performed with the program:

- PIPs fit to the whole potential, as originally proposed by Braams, Bowman, and Xie,
- connected PIPs fit to the whole potential after removing the unconnected terms,
- connected PIPs fit to the many-body part of the potential after removing the unconnected terms and the two-body terms.

The program can also perform a two-stage fit in which one first fits lower-level energetic data with a large number of geometries and then fits higher-level corrections with a smaller set of geometries.

Table of Contents

MANUAL	1
1. Licensing	3
2. Citation.....	3
3. Introduction.....	4
4. Theoretical background.....	6
5. Installation and general usage	16
6. Options	19
7. Program structure.....	29
8. Diatomic potentials.....	33
9. Test inputs	44
10. References.....	48
11. Revision history	49

1. Licensing

PIPFit 2022 is licensed under the [Apache License, Version 2.0](#).

The manual of *PIPFit 2022* is licensed under [CC-BY-4.0](#).

Publications of results obtained with the *PIPFit 2022* software should cite the program and/or an 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 *PIPFit 2022* implies acceptance of the terms of the licenses.

2. Citation

Publications including work performed with *PIPFit 2022* should cite the software, for example, as follows:

Ke R. Yang, Zoltan Varga, Kelsey A. Parker, Yinan Shu, and Donald G. Truhlar
PIPFit 2022, University of Minnesota, Minneapolis, 2022.

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

See also Section 10 for additional references that may be cited.

3. Introduction

This program can be used for fitting potential energy surfaces by permutationally invariant polynomials and a weighted least-squares fit. This method uses a basis of permutationally invariant polynomials (PIPs) of Morse, Gaussian, mixed exponential-Gaussian, or hyperbolic secant variables. Four kinds of fit can be performed:

imethod	Ref(s).	Description
0	[1,2]	Original Braams–Bowman-Xie approach, i.e., a direct fit to the full surface without removing unconnected terms
1	[3]	Remove the unconnected terms. Fit a global surface.
2	[4]	Remove 2-body terms (which are then provided by the user as external potentials). Also remove unconnected terms. Fit the many-body term.
3	[5]	A special case: remove unconnected 3-body terms from an AB ₃ system and fit a global surface.

- [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 Compute.* **6**, 26 (2010).
 [3] Y. Paukku, K. R. Yang, Z. Varga, and D. G. Truhlar *J. Chem. Phys.* **139**, 044309 (2013).
 [4] J. D. Bender, P. Valentini, I. Nompelis, Y. Paukku, Z. Varga, D. G. Truhlar, T. Schwartzenruber, G. V. Candler, *J. Chem. Phys.* **143**, 054304 (2015).
 [5] Y. Shu, J. Kryven, A. G. S. de Oliveira-Filho, L. Zhao, G.-L. Song, S. L. Li, R. Meana-Pañeda, B. Fu, J. M. Bowman, and D. G. Truhlar, *J. Chem. Phys.* **151**, 104311 (2019).

The current version of the code can treat four-body systems of the form A₄, AB₃, A₂BC, or A₂B₂, and it has been applied to N₄, O₄, OH₃, and N₂O₂. The program can also be used for a three-body system by setting the fourth atom very far from the other three atoms so that there is no interaction with them, and it has been applied this way to N₂O, O₃, NO₂, and N₃.

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, in the range from 5 to 10 for A_2B_2 systems, in the range from 5 to 14 for AB_3 systems, and in the range from 5 to 8 for A_2BC systems.

Two formats are allowed for the input data files; the first format does not include indices of geometry data points in the input file, and the second format does include indices of geometry data points.

4. Theoretical background

4.1. Fitting basis functions

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

$$\begin{aligned} r_1 &= r_{12}, & r_4 &= r_{23}, \\ r_2 &= r_{13}, & r_5 &= r_{24}, \\ r_3 &= r_{14}, & r_6 &= r_{34}, \end{aligned}$$

where r_{ij} is the distance between atoms i and j . The variables X_i are used to describe the potential energy surfaces; these variables can be

- Morse-like bond functions: $X_i = \exp(- (r_i - r_{a,i})/a_i)$
- Gaussians: $X_i = \exp(- (r_i - r_{a,i})^2/a_i)$
- Mixed exponential Gaussian (MEG) variables:

$$X_i = \exp\left\{-\left[(r_i - r_{a,i})/a_i\right] - \left[(r_i - r_{b,i})^2/b_i\right]\right\}$$

- Hyperbolic secant (Sech) variables: $X_i = 1/\cosh((r_i - r_{a,i})/a_i)$

The program uses explicitly permutationally invariant basis functions, $S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}]$, as proposed by Braams, Bowman, and Xie.^{1,2}

When `imethod` is 0, the potential energy surface is approximated as follows:

$$V(r_1, \dots, r_6) = V_0 + \sum_{n_1+n_2+n_3+n_4+n_5+n_6=0}^k D_{n_1 n_2 n_3 n_4 n_5 n_6} S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}] \quad (1)$$

where V_0 is a constant equal to the energy of infinitely separated atoms, n_i is power to which a variable is raised, and $D_{n_1 n_2 n_3 n_4 n_5 n_6}$ is a linear coefficient determined by least-squares fitting.

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 ($r_6 = r_{34}$). We label such terms as unconnected terms; they introduce spurious

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).

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 When `imethod=1`, the unconnected terms are removed, and we have

$$V(r_1, \dots, r_6) = V_0 + \sum_{\substack{\text{connected,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}}^k D_{n_1 n_2 n_3 n_4 n_5 n_6} S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}] \quad (2)$$

where V_0 is a constant equal to the energy of infinitely separated atoms, and the $D_{n_1 n_2 n_3 n_4 n_5 n_6}$ are again linear coefficients determined by least-squares fitting. Note that these are not the same coefficients as when one uses eq 1.

Other terms in eqs 1 and 2, such as $X_i^{n_i}$, describe two-body interactions. We can use pre-optimized two-body potential energy functions $V_{\text{PA}}(r_i)$ for better asymptotic behavior. When using pre-optimized two-body potentials, those terms can be detached as known terms and we get the following form:

$$V(r_1, \dots, r_6) = V_0 + \sum_{i=1}^6 V_{\text{PA}}(r_i) + \sum_{\substack{\text{connected, MB,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}}^k D_{n_1 n_2 n_3 n_4 n_5 n_6} S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}] \quad (3)$$

where V_0 is a constant equal to the energy of four atoms, V_{PA} is a pairwise additive term, and the $D_{n_1 n_2 n_3 n_4 n_5 n_6}$ are again linear coefficients determined by least-squares fitting. Note that these are not the same coefficients as when one uses eq 1 or eq 2. The restriction “MB” under the sum sign means that in this case we include only terms where more than one of the n_i is nonzero.

Equation 4 is the recommended method to fit potential energy surfaces with permutationally invariant polynomials (PIPs). In eq 4 the last term with the PIPs is generally called the many-body term.

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

The diatomic potential energies can be described many ways. In Section 4.5 we give a detailed list of the ones currently available in the program.

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 1, 2, and 3 as:

$$\begin{aligned}
 V(r_1, \dots, r_6) = & V_0 + \\
 & + \sum_{\substack{k_a \\ n_1+n_2+n_3+n_4+n_5+n_6=1}} D_{n_1 n_2 n_3 n_4 n_5 n_6}^a S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}] + \\
 & + \sum_{\substack{k_b \\ n_1+n_2+n_3+n_4+n_5+n_6=1}} D_{n_1 n_2 n_3 n_4 n_5 n_6}^b S[Y_1^{n_1} Y_2^{n_2} Y_3^{n_3} Y_4^{n_4} Y_5^{n_5} Y_6^{n_6}]
 \end{aligned} \tag{1A}$$

$$\begin{aligned}
 V(r_1, \dots, r_6) = & V_0 + \\
 & + \sum_{\substack{k_a \\ \text{connected,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}} D_{n_1 n_2 n_3 n_4 n_5 n_6}^a S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}] + \\
 & + \sum_{\substack{k_b \\ \text{connected,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}} D_{n_1 n_2 n_3 n_4 n_5 n_6}^b S[Y_1^{n_1} Y_2^{n_2} Y_3^{n_3} Y_4^{n_4} Y_5^{n_5} Y_6^{n_6}]
 \end{aligned} \tag{2A}$$

and

$$\begin{aligned}
 V(r_1, \dots, r_6) = & V_0 + \sum_{i=1}^6 V_{\text{PA}}(r_i) + \\
 & + \sum_{\substack{k_a \\ \text{connected, MB,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}} D_{n_1 n_2 n_3 n_4 n_5 n_6}^a S[X_1^{n_1} X_2^{n_2} X_3^{n_3} X_4^{n_4} X_5^{n_5} X_6^{n_6}] + \\
 & + \sum_{\substack{k_b \\ \text{connected, MB,} \\ n_1+n_2+n_3+n_4+n_5+n_6=1}} D_{n_1 n_2 n_3 n_4 n_5 n_6}^b S[Y_1^{n_1} Y_2^{n_2} Y_3^{n_3} Y_4^{n_4} Y_5^{n_5} Y_6^{n_6}]
 \end{aligned} \tag{3A}$$

In these equations, a and b label two sets of permutationally invariant polynomial basis functions with the orders of k_a and k_b , respectively. The X_i and Y_i variables need not be the same kind of variables in the two sets; either of them can be Morse, Gaussian, MEG, or Sech variables, and this provides improved flexibility to 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 with a large number of geometry data points and a higher-level (HL) correction with a subset of these geometry data 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 geometry data 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 incomplete inclusion of dynamic electron correlation – the available accuracy is limited. But such calculations can be taken as an LL calculation that – due to its lower cost – can be carried out for a relatively large number of geometry data points. Then, for higher accuracy one can use HL calculations, for instance, subsequent CASPT2 calculations at a smaller number of geometry data 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_{Total}). The surface of LL energies is usually as rugged as the surface of V_{Total} ; therefore, it requires a large number of geometry data points to get a reasonable surface. But the surface of HL corrections will often be relatively smoother, and its energy range will be relatively narrow as compared to that of the LL surface. Hence, the HL surface can be described well with a significantly lower number of input geometry 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, provided that both levels are qualitatively correct for the global shape of the surface.

4.3. Weighted least-squares fit

With m basis functions to fit n geometry 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 geometry 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 ($j = 1, \dots, m$). The determination of linear coefficients is equivalent to minimizing the functional,

$$F = \sum_{i=1}^n \left(\sum_{j=1}^m b_{ij} c_j - V_i \right)^2 \quad (4)$$

Weights, W_{ii} , can be assigned to different geometry data point i ($i = 1, \dots, n$), yielding a weighted functional:

$$F = \sum_{i=1}^n W_{ii} \left(\sum_{j=1}^m b_{ij} c_j - V_i \right)^2 \quad (5)$$

Equation 5 is the functional used in *PIPFit 2022*. The minimization of the functional F in eq 5 leads to the normal equation,

$$\mathbf{B}^T \mathbf{W} \mathbf{B} \mathbf{c} = \mathbf{B}^T \mathbf{W} \mathbf{V} \quad (6)$$

where \top denotes a matrix transpose, and

$$\mathbf{B} = \begin{bmatrix} b_1(\mathbf{x}_1) & b_2(\mathbf{x}_1) & \cdots & b_m(\mathbf{x}_1) \\ b_1(\mathbf{x}_2) & b_2(\mathbf{x}_2) & \cdots & b_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(\mathbf{x}_n) & b_2(\mathbf{x}_n) & \cdots & b_m(\mathbf{x}_n) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \quad (7)$$

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix}, \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} W_{11} & 0 & \cdots & 0 \\ 0 & W_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{nn} \end{bmatrix}$$

4.4. Weighting functions

Various weighting functions are available for weighted least-squares fits in *PIPFit 2022*, as selected by the input variable `iwop`.

iwop	weighting function	eq
1	$W_{ii} = W_{ii}^{\text{en}} = \begin{cases} 1 & \text{for } (V_i - E_{\text{sh}}) \leq E_{\text{cr}} \\ (E_{\text{cr}} / (V_i - E_{\text{sh}}))^{E_{\text{pow}}} & \text{for } (V_i - E_{\text{sh}}) > E_{\text{cr}} \end{cases}$	(8)
3	$W_{ii} = W_{ii}^{\text{en}} = \begin{cases} 1 & \text{for } (V_i - E_{\text{sh}}) \leq E_{\text{cr}} \\ \left(\frac{V_i - E_{\text{sh}}}{E_{\text{cr}}} \right)^{E_{\text{pow}}} & \text{for } (V_i - E_{\text{sh}}) > E_{\text{cr}} \end{cases}$	(9)
2 or 4	$W_{ii} = W_{ii}^{\text{en}} W_{ii}^{\text{geom}}$ $W_{ii}^{\text{geom}} = \frac{d_i}{\max(d_i)}$ $d_i = \left(\sum_{k=1}^n d_{i,k} \right) / n$ $d_i = \left(\sum_{k=1}^n d_{i,k} \right) / n$	(10) (11) (12) (13)

In eqs 8 and 9,

- W_{ii}^{en} is the weighting factor of geometry data point i with energy V_i
- E_{cr} is a user-selectable energy parameter to reduce the weights of high-energy geometry data points
- E_{sh} is a selectable energy shift that is useful when fitting triatomic potentials in the four-body frame,
- E_{pow} is a selectable power (the default value is 2).

Equation 8 is designed for fitting adiabatic or diabatic potential energy surfaces, where higher energies (which may be classically inaccessible during dynamics) get less

weight, whereas eq 9 is designed for fitting diabatic couplings if it is desired for higher values to get more weight.

In eqs 10-13, $r'_{i,j}$ is one of the internuclear distances (j) for a given geometry data point i , and $r'_{k,j}$ is an internuclear distance between atoms with the same atomic numbers for any geometry other data point k . 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 in $r_{i,j}$ 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.

If in eq 10 the W_{ii}^{en} is calculated by eq 8, then $i_{wop} = 2$ and if in eq 10 the W_{ii}^{en} is calculated by eq 9, then $i_{wop} = 4$.

4.5. Diatomic potentials

Currently three different functions are used for describing the diatomic potentials.

The generalized Morse (GM) potential is defined by

$$V_{\text{PA}}(r) = D_e[1 - \exp(-f(r)(r - r_e))]^2 - D_e, \quad (13)$$

where D_e is the equilibrium bond dissociation energy and $f(r)$ depends on r . We defined intermediate variable

$$y = \frac{r^4 - r_e^4}{r^4 + r_e^4} \quad (14)$$

and expanded $f(r)$ in a Taylor series of y :

$$f(r) = \sum_{i=0}^6 a_i y^i \quad (15)$$

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

The mixed exponential–Gaussian (MEG) is defined by

$$V_{\text{PA}}(r) = -D_e \sum_{i=1}^6 c_i X^i, \quad (16)$$

where D_e is the equilibrium bond dissociation energy and X is the MEG variable:

$$X = \exp[-(r - r_e)/a - (r - r_e)^2/b], \quad (17)$$

where r_e is the equilibrium bond length, and the non-linear parameters a and b as well as the linear parameters c_i are optimized against calculated energies.

The third possibility is a long-range-corrected potential defined by

$$V_{\text{PA}}(r) = V_{\text{SR1}}(r) + V_{\text{LR}}(r), \quad (18)$$

where SR denotes short range, LR denotes long range, and $x = 1$ for O_2 , $x = 2$ for NO , and $x = 3$ for N_2 . Grimme's D3 function with the Becke-Johnson damping (BJ) function is used as long-range term, $V_{\text{LR}} = V_{\text{D3(BJ)}}$.^{4,5} For a diatomic molecule it is given by

$$V_{\text{D3(BJ)}}(r_i) = \sum_{n=6,8} s_n \frac{C_n}{r_i^n + (a_1 r_0 + a_2)^n} \quad (19)$$

where r_i is the distance between the two atoms, r_0 is given by

⁴ S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, *J. Chem. Phys.* **132**, 154104 (2010).

⁵ S. Grimme, S. Ehrlich, and L. Goerigk, *J. Comput. Chem.* **32**, 1456, (2011).

$$r_0 = \sqrt{C_6/C_8}, \quad (20)$$

C_6 and C_8 , are the dispersion coefficients for a pair of atoms, the parameter s_6 is 1.0, the parameter s_8 is 2.0, and based on Ref. 6 we set $a_1 = 0.5299$ and $a_2 = 2.2$. The treatment of C_n is a simplified version of that used originally by Grimme because we do not need consider coordination number dependence in diatomic molecules.

PIPFit 2022 contains the parameters for applying eqs 16–18 to O₂, NO, and N₂. For O₂ molecule, we fixed C_6 at 176.37 kcal·Å⁶/mol, and C_8 is calculated inside the program from C_6 by the formula used in Refs. 4 and 5. For NO molecule, the C_6 value is 230.11 kcal·Å⁶/mol. For N₂ molecules, the C_6 value is 271.45 kcal·Å⁶/mol.

For O₂ molecule, the parameters of V_{SR1} were determined by fitting eq 18 to the accurate potential curve of O₂ by Bytautas *et al.*⁷ This potential energy curve is based on accurate calculations and includes corrections for core electron correlations and relativistic effects. For V_{SR1} we use the even-tempered Gaussian fitting function of Bytautas *et al.*⁷, given by

$$V_{\text{SR1}}(r_i) = \sum_{k=0}^7 a_k \exp(-\alpha\beta^k r_i^2), \quad (20)$$

where we obtain the coefficients a_k by linear regression and the exponent parameters α and β by nonlinear minimization.

For NO molecule, the parameters of V_{SR2} were determined by fitting eq 18 to the accurate potential curve of NO. For V_{SR2} an eight-order polynomial is used based on MEG variables

$$V_{\text{SR2}}(r_i) = -D_e \sum_{i=1}^8 c_i X^i, \quad (21)$$

where D_e is the equilibrium bond dissociation energy and X is the MEG variable of eq 17.

For N₂ molecule, the parameters of V_{SR3} were determined by fitting eq 18 to the accurate potential curves of N₂ (with and without SEC corrections taken into account).

⁶ P. Verma, B. Wang, L. E. Fernandez, and D. G. Truhlar, *J. Phys. Chem. A* **121**, 2855 (2017).

⁷ L. Bytautas, N. Matsunaga, and K. Ruedenberg, *J. Chem. Phys.* **132**, 074307 (2010).

For these potentials, the functional form of V_{SR3} corresponds to the functional form of the GM function (see eqs 13–15).

If separate fits of lower-level energies and high-level corrections are used, then it also affects the treatment of diatomic potential energies. As it was already mentioned in Section 4.2, one can do, for instance, CASSCF and subsequent CASPT2 calculations as LL and HL, respectively. Thus, the diatomic curves have to be fitted based on CASSCF data as LL, and the HL correction energy can be obtained as a difference between the LL diatomic curve and the accurate diatomic one. Currently, the CASSCF diatomic curves are described as functions of MEG variables. Note that these preliminary CASSCF curves were only used in the tests of LL + HL formalism so far; they were not used in any published work or surface subroutine.

5. Installation and general usage

5.1. *PIPFit 2022* program

The *PIPFit 2022* software package is free-of-charge. Download instructions are given at <https://comp.chem.umn.edu/pipfit/>

The *PIPFit 2022* program is distributed as a tar.gz file, which can be untarred by executing

```
% tar -xzvf PIPFitv2022.tar.gz
```

This should create the following directories

doc/	For the Manual
src/	<i>PIPFit 2022</i> program code
src_list/	<i>PIPList</i> program code
test/	Examples

The Fortran source codes to perform potential energy surface fit are in the src/ directory.

A make file named makefile is also in that directory. LAPACK⁸

(<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, one needs 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:

⁸ 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.abn or <name>.abn          if ecor =0 or 1
main.abn11 or <name>.abn11      if ecor =2 or 4
main.abnh1 or <name>.abnh1      if ecor =3 or 4

```

If `<name>.abn` (`.abn11`, `.abnh1`) is not found, `main.abn` (`.abn11`, `.abnh1`) will be read as a secondary option. We suggest using the `main.abn` (`.abn11`, `.abnh1`) file for the “whole” data set and the `<name>.abn` (`.abn11`, `.abnh1`) files if data set was modified for specific purposes.

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

```
% ./pipfit
```

After the execution the program will inquire about the file name, the user 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 geometry 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, the diatomic
                 energies, and the sum of dispersion corrections
<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 2022* 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 2022* code, except the request of LAPACK package.

The execution of *PIPList* program requires three input files (their formats correspond to *PIPFit 2022* code):

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

The coefficients are generated by the *PIPFit 2022* 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 2022* 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 records:

\$molecule, \$energy, \$fitfunct, and \$inout. Each input record begins with its name, e.g., \$molecule, and is closed by a \$end statement. The \$molecule input record includes molecule-specific keywords. The \$fitfunct input record contains keywords related to the fitting functions. The \$inout input record includes keywords related to input and/or output options. Finally, the \$energy input record includes keywords 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	AB ₃ system	(IOrder = 5-14)	usable
= 3	A ₂ B ₂ system	(IOrder = 5-12)	usable
= 4	A ₂ BC system	(IOrder = 5-8)	usable
= 5	ABCD system	(IOrder = 5-9)	currently unavailable
iatyp = 56 (default)	N ₄ , a unique identifier for a given four-body system.		
= 60	N ₂ O 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-like bond function: $X_i = \exp[-(r_i - r_{a,i})/a_i]$		
= 2	Gaussian: $X_i = \exp[-(r_i - r_{b,i})^2/a_i]$		
= 3	Mixed exponential Gaussian (MEG):		
	$X_i = \exp[-(r_i - r_{a,i})/a_i - (r_i - r_{b,i})^2/b_i]$		
= 4	Sech: $X_i = 1/\cosh((r_i - r_{a,i})/a_i)$		

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 data 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 geometry data 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 <code>ityp</code> command (under <code>\$molecule</code> input record).
iorderb		Degree of the second set of polynomial basis functions ($n_{\text{fun}} = 2$) See the available IOrder numbers for different kinds of systems at <code>ityp</code> command (under <code>\$molecule</code> input record).
iorderc		Degree of polynomial basis functions for HL corrections. See the available IOrder numbers for different kind of systems at <code>ityp</code> command (under <code>\$molecule</code> input record).

Depending on the selected `imgx`, `nfun`, and `ityp` options, multiple distance parameters and range-separation parameters have to be set in the input record. In the bond order variables (`imgx`), the distance parameters are the **ra1-ra6**, **rb1-rb6**, etc. values and the range-separation parameters are the **arng1-arng6**, **brng1-brng6**, etc. values. The units of the distance parameters are in lengths and the units of the range-separation parameters are lengths or length squares depending on the selected type of the bond order variable. We

prefer using Å-based units in our surface fits. Currently six sets (a, b, c, d, e, f) of these parameters are implemented in the code and in each set there are six parameters (1, 2, 3, 4, 5, 6). For further details about these parameters, see Sections 6.2, 6.3, and 6.4.

ra1-ra6		The first set of six distance parameters
rb1-rb6		The second set of six distance parameters
rc1-rc6		The third set of six distance parameters
rd1-rd6		The fourth set of six distance parameters
re1-re6		The fifth set of six distance parameters
rf1-rf6		The sixth set of six distance parameters
arng1-arng6		The first set of six range-separation parameters
brng1-brng6		The second set of six range-separation parameters
crng1-crng6		The third set of six range-separation parameters
drng1-drng6		The fourth set of six range-separation parameters
erng1-erng6		The fifth set of six range-separation parameters
frng1-frng6		The sixth set of six range-separation parameters
pairw	= 1	Generalized Morse (GM) potentials are used for all diatomics, for more details see Sections 4.4. and 8.1.
	= 2	GM potential for N ₂ , MEG for NO and for O ₂ , for more details see Sections 4.4. and 8.1.
	= 3	SR+LR potentials for O ₂ , NO, and N ₂ . For more details see Sections 4.4. and 8.1.
imethod	= 0	Braams- Bowman-Xie approach
	= 1	Remove the unconnected terms
	= 2	Remove the 2-body, which treated as external potentials, and unconnected terms
	= 3	Special case to remove the unconnected 3-body terms, only works for AB ₃ systems
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 $\ \mathbf{b} - \mathbf{A}*\mathbf{x}\ $ (default: rcond = 1.0d-15)
\$energy:	Energy thresholds input record
iwop = 0	No weighting functions is used.
= 1	Equation 8 is used.
= 2	Equation 8 is extended with a geometry-dependent factor using eqs 10-13.
= 3	Equation 9 is used.
= 4	Equation 9 is extended with a geometry-dependent factor using eqs 10-13.
Ecr	Critical energy threshold used in the weighting function (eqs 8 and 9), in kcal/mol
Esh	An energy shift value used in the weighting function (eqs 8 and 9), in kcal/mol
Epow	The applied power in weighting function (eqs 8 and 9)
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
grange1-grange4	Boundaries for geometries, as $(\sum_{i=1}^6 r_i)/6$, to report root mean square deviations (RMSDs) and mean unsigned deviations (MUDs), in kcal/mol
sacl, smiel, smacl	Parameters to control error information printing in the <code><name>.err</code> file, all in kcal/mol. (The geometry data points with electronic structure energies in the range of smiel < V < smacl will be printed in the <code><name>.err</code> file if the absolute deviations are larger than sacl)
\$inout:	Input and output related keywords
icoord = 0	Internal (Jacobi) coordinates

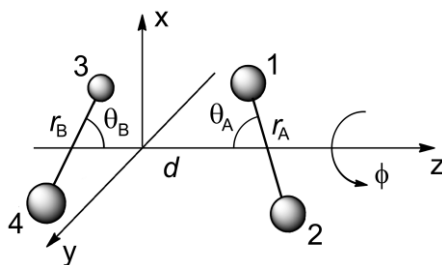
	= 1	Cartesian coordinates
	= 2	Internuclear distances
ifmt	= 0	The electronic structure data set does not include data indices.
	= 1	The electronic structure data includes data indices.
iout	= 0	Write electronic structure and fitted energies of geometries into <name>.vf0 file.
	= 1	Write electronic structure, 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 this option is used, then the user should use a <i>very</i> 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 geometry data points.
	= 1	Generate reduced data set in “main.abn.red” file by deleting every <i>nred</i> -th geometry data point.
nred		Integer number used to generate reduced geometry 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 geometry 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 two diatomic molecules.



Scheme 1. Internal coordinates of a four-body system.

The data order of a geometry data 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 $\text{N}_2(r_e) + \text{N}_2(r_e)$ for N_4 system.

When the data are organized in Cartesian coordinates, then each geometry data point occupies five lines as a block:

ID number (if applied), ΔE_{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 (ΔE_{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 be consistent with r_{a_i} , r_{b_i} , r_{c_i} , r_{d_i} , r_{e_i} , r_{f_i} , arng_i , brng_i , crng_i , drng_i , erng_i , and frng_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 potentials are hard coded with this convention.

In the work of OH_3 system, which uses the general AB_3 atom system, one can consider the three H atoms as A, B, and C. Then $r_1 = r_{\text{OA}}$, $r_2 = r_{\text{OB}}$, $r_3 = r_{\text{OC}}$, $r_4 = r_{\text{AB}}$, $r_5 = r_{\text{AC}}$, and $r_6 = r_{\text{BC}}$.

In the work of methylamine ($\text{CH}_3\text{-NH}_2$), which uses the general A_2BC atom system, the order is $r_1 = r_{\text{H}(1)\text{H}(2)}$, $r_2 = r_{\text{H}(1)\text{N}}$, $r_3 = r_{\text{H}(1)\text{C}}$, $r_4 = r_{\text{H}(2)\text{N}}$, $r_5 = r_{\text{H}(2)\text{C}}$, and $r_6 = r_{\text{NC}}$, where atom H(1) and H(2) are the H atoms of the amine group and C represents the methyl (CH_3 -)group.

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 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 r_{a_i} , r_{b_i} , ..., r_{f_i} parameters are the “equilibrium distances” in the equations of the selected variables. If the user sets up 1.098 Å for parameter r_{a_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 (a_{rng_i} , b_{rng_i} , f_{rng_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 , ΔE_{Total} , ΔE_{LL} (if `ecor=1`)

6.3. Parameters of \mathbf{X}_i variables

Depending on the kind of the four-body system (`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} , a_{rng_i} , b_{rng_i} , c_{rng_i} , d_{rng_i} , e_{rng_i} , f_{rng_i} parameters. For an A_4 (homonuclear) system, values of r_{a_1} (r_{b_1} , ...) and a_{rng_1}

($brng_1, \dots$) must be given, and the program repeats their values for $i = 2-5$. For an A_2B_2 (heteronuclear) system, ra_i (rb_i, \dots) and $arng_i$ ($brng_i, \dots$) ($i = 1, 2, \text{ and } 6$) entries must be given and the parameters of $i = 2$ will be repeated in the code for the cases of $i = 3-5$. For example, in N_2O_2 system (see Section 6.2.) $i = 1$ corresponds to the O-O bond, $i = 2-5$ are the four N-O bonds, and $i = 6$ is the N-N bond. For this N_2O_2 system an example input file can be seen in Section 8. For an AB_3 system, two parameter sets ($i = 1$ and 4) are required, $i = 1$ also correspond to $i = 2$ and 3 as well as $i = 4$ also correspond to $i = 5$ and 6 based on the order of the atom pairs. For an A_2BC system, four parameter sets ($i = 1, 2, 3, \text{ and } 6$) are required, $i = 2$ also correspond to $i = 4$ as well as $i = 3$ also correspond to $i = 5$.

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 procedures were implemented:

- 1) If $ecor=1$, then the file `<name>.abn` or `main.abn` is read. In this case the file must contain the ΔE_{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` belongs to the HL correction part of the fit. Depending on the selected `imgx` option several `rxi` and `xrngi` ($x = a, b, c, d, e, f$) parameters are required. For example, if $ecor = 1, nfun=2, imga = 3, imgb = 3,$ and $imgc = 1,$ then the sets `ra`, `arng`, `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 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` belong to the 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 ΔE_{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` are required as additional parameters).
- 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 `rxi` and `xrngi` (`x=a, b, ...`) 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 `erng` 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>.vfl1 and <name>.vfh1. These files contain the ID number, the input energy, the calculated energy, and the diatomic contributions for each geometry data 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 by CASSCF calculations and scaled external corrected CASPT2 calculations. They are located in $ev2gm2.f$ file. And the $V_{PA,HL}$ correction data (for $ecor = 1, 3,$ and 4) is calculated as energy difference of $V_{PA,HL}$ and $V_{PA,LL}$.

7. Program structure

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

```
bmx2b1.f
bmx2b3.f
bmx2b4.f
bmx3b2.f
bmxnc1.f
bmxnc3.f
bmxnc4.f
call_dgelsd.f
calrmsd.f
change_case.f
coordconvt.f
detcoef.f
dispersion.f
ev2b.f
ev2gm2.f
evbas.f
evmono1.f
evmono2.f
evmono3.f
evmono4.f
evms.f
evpoly1.f
evpoly2.f
evpoly3.f
evpoly4.f
frmbm.f
frmbmp.f
frmbmx2b.f
frmbmx3b.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 2022* 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 `imethod = 1` is specified. The two-body terms and unconnected terms are removed by calling the `frmbmx2b` subroutine if `imethod = 2` is specified. The

unconnected three-body terms are removed by calling the `frmbmx3b` subroutine if `imethod = 3` 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 `dge1sd` subroutine from LAPACK library.

`dispersion.f`:

The `d3disp` subroutine calculates the D3 dispersion correction for each diatom. For this purpose, it calls `copyc6`, `limit`, `edisp`, `cn_calc`, `getc6`, and `setr0ab` subroutines located in the same file. These subroutines are originated Grimme's DFT-D3 (Ver. 3.2 Rev 0) program.

`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` and `calrmsd_geom` subroutines calculate and report root-mean-square deviation and mean unsigned deviation between fitted and calculated energies based on energy and geometry separations, respectively.

`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 geometry 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 `imethod = 2` is chosen. The two-body interactions are calculated by calling the `ev2gm2` subroutine. It also calls the `d3disp` subroutine if `pairw = 2`.

`ev2gm2.f`

The `ev2gm2` subroutine contains the diatomic potentials. Currently the pairwise potentials of N_2 , O_2 , and NO are available based on various methods.

`evbas.f`:

The `evbas` subroutine calculates basis functions for a given geometry by calling the `evpolyx` subroutines ($x = 1-4$) which calculate the permutationally invariant polynomials (PIP) from monomials, which are evaluated with the `evmonox` subroutines ($x = 1-4$) based on the `evms` subroutine.

`frmbmxnc.f`:

The `frmbmxnc` subroutine removes the unconnected terms in the PIP basis functions by calling the `bxnc1`, `bxnc3`, and `bxnc4` subroutines.

`frmbmx2b.f`:

The `frmbmx2b` subroutine removes the unconnected terms in the PIP basis functions by calling the subroutines in the files `bx2b1.f`, `bx2b3.f`, and `bx2b4.f`.

`frmbmx3b.f`:

The `frmbmx3b` subroutine removes the unconnected terms in the PIP basis functions by calling the subroutines in the file `bx3b2.f`.

8. Diatomic potentials

8.1. Available diatomic potentials

Table 1 summarizes the currently available diatomic potentials. The input parameters `pairw`, `ityp`, and `iatyp` (these keywords are explained in Section 6.1.) control which kind of diatomic function set is called for the calculations. Then the internal parameter `imol` is associated with the different bond pairs (1 and 2 mean NN, 3 means NO, and 4 mean OO). For instance, in the case of singlet N₄ potential energy surface the GM diatomic potential was used and to call this the user has to set up `pairw=1`, `ityp=1`, and `iatyp=56` as input parameters. Another example is the triplet N₂O₂ potential energy surface, where the input parameters are `pairw=2`, `ityp=3`, and `iatyp=60` and these settings call the NN potential by GM function and the NO and OO potentials by MEG functions. The third example is the singlet and quintet O₄ potential energy surfaces, where the input parameters are `pairw=3`, `ityp=1`, and `iatyp=64` and this combination calls the SR1+LR function for OO.

Table 1. List of the currently implemented diatomic potentials

<code>pairw</code>	<code>ityp</code>	<code>iatyp</code>	<code>imol</code>	function ^a	diatom
= 1	= 1	= 56	= 1	GM	NN
	= 3	= 60	= 2	GM	NN
	= 3	= 60	= 3	GM	NO
	= 3	= 60	= 4	GM	OO
	= 1	= 64	= 4	GM	OO
= 2	= 3	= 60	= 2	GM	NN
	= 3	= 60	= 3	MEG	NO
	= 3	= 60	= 4	MEG	OO
	= 1	= 64	= 4	MEG	OO
= 3	= 1	= 56	= 1	SR3+LR	NN
	= 3	= 60	= 2	SR3+LR	NN
	= 3	= 60	= 3	SR2+LR	NO
	= 1	= 60	= 4	SR1+LR	OO
	= 1	= 64	= 4	SR1+LR	OO

^a Abbreviation GM: Generalized Morse; MEG: Mixed exponential Gaussian; SR_x+LR: short-range and long-range ($x = 1,2,3$)

8.2. Adding new diatomic potentials

The *PIPFit 2022* program can be extended with new diatomic potentials. This procedure can be a relatively straightforward file editing if only an existing parameter set is replaced by a new one, or slightly more difficult procedures if new function forms are introduced for such atoms, which are not defined yet. In this description we try to cover all the necessary cases by detailed descriptions.

8.2.1. Replacing the parameters of an existing diatomic potential

In this case the user should open the file `ev2gm2.f` and search for the parameters of the desired atom pair in subroutine `ev2gm2`. For instance, the parameters look like the following text:

```
C Parameter for N2 dissociation without SEC
  re=1.098d0
  de=228.7d0
  c(0) = 2.70963254293d0
  c(1) = 1.32620177271d-1
  c(2) = 2.96757048793d-1
  c(3) = 1.97112432229d-1
  c(4) = -5.02002309588d-1
  c(5) = 3.80734244606d-1
  c(6) = 1.21001628750d0
```

As it can be seen in the example, the application of comments is highly recommended in case of modifications. Currently the program contains multiple diatomic potential function forms (see them in Section 4.5). For instance, for the GM function the required parameters are the equilibrium bond length (r_e in Angstrom) and the equilibrium dissociation energy (d_e in kcal/mol) of the diatomic molecule as well as the fitted coefficients, $c(0) \dots c(6)$.

In that case if the user wants to use one of the LL + HL (CASSCF and correlation energies are separately fitted) options, then the subroutine `ev2gm2cas` in the file `ev2gm2.f` is used with the diatomic potential energy function, where the fitted data were obtained by CASSCF calculations. The current function form for CASSCF energies is based on a series of MEG variables (see eqs 16 and 17 in Section 4.5).

```
C Parameter for N2 dissociation without SEC (for N4 data)
  re=1.1038d0
  de=204.07d0
```

```

c(0) = 0.0d0
c(1) = 0.278807015301273d0
c(2) = -5.00836910898787d0
c(3) = 5.80223078491159d0
c(4) = -2.682668959071380
c(5) = 0.573795631347316d0
c(6) = 0.036615059234527d0
c(7) = 0.568570812839021d0 ! delta b
c(8) = 2.1408123632275d0 ! delta c

```

In this case the required parameters are the equilibrium bond length (r_e in Å) and the equilibrium dissociation energy (d_e in kcal/mol) of the diatomic molecule, the coefficients $c(7)$ and $c(8)$ are the two non-linear parameters of MEG variable (see eq 16), and the linear coefficients are $c(1) \dots c(6)$. The structures of subroutines `ev2gm2` and `ev2gm2cas` are very similar.

The codes of *PIPFit 2022* and *PIPList* only differ in the file `pesfit.f`, thus any other modified files can be copied from the directory `src/` to the directory `scr_list/`.

8.2.2. Introducing diatomic potentials for new atom pairs

This operation will contain the modification of several files. The order of the described steps is somewhat flexible until all of the necessary modifications are done. First, the user needs to define a new four-body group (`iatyp`, which is a common variable). The number of this variable is based on the masses of the atoms, for example, `iatyp = 56` for N_4 system (4×14). In the file `readinp.f` this new `iatyp` must be allowed. In the next example from the file `readinp.f` three cases are allowed, they are 56, 60, and 64 for N_4 , N_2O_2 , and O_4 systems, respectively.

```

if (iatyp .eq 56) then
  write(30,'(a)') ' N4 system is considered.'
else if (iatyp .eq 60) then
  write(30,'(a)') ' N2O2 system is considered.'
else if (iatyp .eq 64) then
  write(30,'(a)') ' O4 system is considered.'
else
  write(30,'(a)') 'Only 56, 60, or 64 is allowed for IATyp!'
endif

```

If the user wants to add, for instance, the H_4 system, then `iatyp` will be 4 and that has to be listed (in boldface):

```

if (iatyp .eq 56) then
  write(30,'(a)') ' N4 system is considered.'
else if (iatyp .eq 60) then
  write(30,'(a)') ' N2O2 system is considered.'
else if (iatyp .eq 64) then
  write(30,'(a)') ' O4 system is considered.'
else if (iatyp .eq 4) then
  write(30,'(a)') ' H4 system is considered.'
else
  write(30,'(a)') 'Only 4, 56, 60, or 64 is allowed for IATyp!'
endif

```

Then the next file to be modified is the `ev2b.f`. First let's consider that we have an A_4 type system (`ITyp = 1`). In this case one has to extend the subroutine `readingp`. Using the previous example, H_4 , the new lines will be right before the command `else` in the `if` statement of `iatyp`:

```

else if (iatyp .eq 4) then
  v=0.0d0
  do i=1,6
    r=dist(i)
C Call H2 PEC, De = 109.5 kcal/mol
    call ev2gm2(r,v2,dv2dr,pairw,5,0)
    v=v+v2
! list diatomic energies into file
    if (i .eq 1) then
      V2ABx=v2
    else if (i .eq 2) then
      V2ACx=v2
    else if (i .eq 3) then
      V2ADx=v2
    else if (i .eq 4) then
      V2BCx=v2
    else if (i .eq 5) then
      V2BDx=v2
    else if (i .eq 6) then
      V2CDx=v2
    endif
  enddo
C v = 0.0 for infinite separated 4H, so 2De needs to be added back.
  v=v+219.0d0

```

As discussed above, `iatyp` is equal to 4, this is the first boldface text in the example. The next two boldface entries are in a comment, which says that the potential energy curve of H_2 molecule will be called in the next line, where the equilibrium dissociation energy is 109.5 kcal/mol; in the next line, the subroutine `ev2gm2` is called. The parameters of this subroutine, in the parameter list of

`ev2gm2 (r, v2, dv2dr, pairw, imol, igrad)`, are the bond length, energy of the potential, gradient of the potential, pairwise switch, molecule switch, and gradient switch, respectively. As can be seen from the structure of subroutine `ev2b`, the variable `r` takes the values of internuclear distances, `dist(i)`, in a `do` cycle of `i = 1,6`; and it calculates the energies, `v2`, which modify the value of `v`. In the call of the subroutine, the variable `imol` has noteworthy importance. If `imol = 1` or `2`, then diatomic potentials of N_2 are called (see the differences in Table 1). The value `imol = 3` is assigned to the diatomic potential of NO , and `imol = 4` is assigned to the diatomic potential of O_2 . The next entry for the new diatomic potentials can be `imol = 5`; see the boldface number in the example above. Since the `igrad` variable equals 0, the gradient calculation is not carried out, and the value of `dv2dr` is not used during the run of the program. For the usage of `pairw`, see Section 8.2.3. Then the diatomic energies are saved separately; this part does not need to be modified. Finally, two times the dissociation energy is added to `v`; see the last two lines.

The subroutine `ev2gm2` in the file `ev2gm2.f` also has to be modified, because that subroutine is called in subroutine `ev2b`. The parameters and equations of the different diatomic potentials are placed into `if` statement structures according to the different pairwise and molecule switches. In this example, an already implemented equation form is used, which is selected by the `pairw` option, therefore only the `if` statement of `imol` for that selected pairwise option has to be changed. A new `else if` statement with `(imol.eq.5)` has to be added, and below that the new parameters of the diatomic potential have to be listed. For the structure of the parameters see in Section 8.2.1.

For an A_2B_2 system, the procedure of adding anew diatomic potential is similar to the procedure described above. Consider an H_2O_2 system. In this case `iatyp` will be 34 (2×1 and 2×16). The user has to edit the file `readinp.f` as described above. In the file `ev2b.f`, the required modification is slightly different than has so far been described since – via the subroutine `ev2gm2` – different diatomic molecules (`imol`) will be called. Here is an example `else if` segment for the `if` statement of `iatyp`, where we assume that `dist(1)` and `dist(6)` correspond to O–O and H–H distances,

respectively. Hence, `dist(2)`, `dist(3)`, `dist(4)`, and `dist(5)` correspond to four possible HO distances.

```

        else if (iatyp .eq 34) then
            v=0.0d0
C r1 = rOO, call O2 PEC, De = 120.243 kcal/mol
            r=dist(1)
            call ev2gm2(r,v2,dv2dr,pairw,4,0)
            v=v+v2
! diatomic OO energy
            v2ABx=v2
C r2, r3, r4, r5 are rHO, call HO PEC, De = ??? kcal/mol
            do i=2,5
                r=dist(i)
                call ev2gm2(r,v2,dv2dr,pairw,6,0)
                v=v+v2
! each diatomic HO energy
                if (i .eq 2) then
                    V2ACx=v2
                else if (i .eq 3) then
                    V2ADx=v2
                else if (i .eq 4) then
                    V2BCx=v2
                else if (i .eq 5) then
                    V2BDx=v2
                endif
            enddo
C r6 = rHH, call H2 PEC, De = 109.5 kcal/mol
            r=dist(6)
            call ev2gm2(r,v2,dv2dr,pairw,5,0)
            v=v+v2
! diatomic HH energy
            v2CDx=v2
C v = 0.0 for infinitely separated 2O + 2H, so De(H2)+De(O2) needs to
C be added back.
            v=v+ 229.74d0

```

In this new thread, `iatyp` is equal to 34, and first the diatomic energy of the OO bond is calculated by calling subroutine `ev2gm2`. Since the OO potential is already there in the file `ev2gm2.f` and its number is 4, we have `imol` equal to 4 in this case. In the next step the diatomic energies of the four OH distances are calculated. Let's use number 6 as an identification number for the OH bonds (`imol = 6`). Finally, the HH potential is calculated, and we can use `imol = 5` for this as we did in the example above.

The subroutine `ev2gm2` in the file `ev2gm2.f` also has to be modified, because that subroutine is called in the subroutine `ev2b`. The parameters and equations of the different diatomic potentials are placed into `if` statement structures according to the

different pairwise and molecule switches. In this example, an already implemented equation form is used, which is selected by the `pairw` option; therefore only the `if` statement of `imol` for that selected pairwise option has to be changed. Two new `else if` statements with `(imol.eq.5)` for HH distance and `(imol.eq.6)` for OH distances have to be added and below them the new parameters of the diatomic potentials have to be listed. For the structure of the parameters see in Section 8.2.1.

If the user wants to use one of the LL + HL (CASSCF and correlation energies are separately fitted) options, then subroutine `ev2gm2cas` in the file `ev2gm2.f` is used with the diatomic potential energy function, where the fitted data were obtained by CASSCF calculations as LL. The structure of subroutines `ev2gm2` and `ev2gm2cas` are very similar, so the necessary modifications can be done based on the above descriptions. In a similar way in the file `ev2b.f`, besides the subroutine `ev2b` the subroutine `ev2bcas` needs to be modified too. Here again, the structure of the subroutines is very similar. Please note that subroutine `ev2bcas` has to call the subroutine `ev2gm2cas`. The file `ev2b.f` contains a subroutine `ev2bcor`, where no further modification is needed.

The codes of *PIPFit 2022* and *PIPList* only differ in the file `pesfit.f`, thus any other modified files can be copied from the directory `src/` to the directory `scr_list/`.

8.2.3. Introduction of new functional forms

A variable, `pairw`, to switch among different diatomic functional forms is already introduced into the code. In the current code the `pairw = 1, 2, and 3` options are used. When the user wants to introduce a new function form, then one has to extend the `if` statement of `pairw` in the subroutine `ev2b` in the file `ev2b.f` as well as increase the acceptable numbers for `pairw` in the file `readinp.f`. Here is an example for the necessary modifications in the file `readinp.f` (boldface texts):

```

if (pairw .gt. 4) then
  write(*,'(a)') ' Wrong pairw, it must be 1, 2, 3, or 4'
  stop
endif

```

The user has to make more extensive modifications in the subroutine `ev2gm2` in the file `ev2gm2.f`. The `if` statement of `pairw` needs to be extended with the new option `pairw = 4`.

```

else if (pairw .eq 3) then
...
else if (pairw .eq 4) then
...
endif

```

In each `pairw` segment there is an `if` statement for each implemented diatomic molecule (`imol`). These blocks contain the parameters for the potential energy curves of the diatomic molecules. For the generalized Morse function such a data block looks like this (including the `if` patterns):

```

if (imol .eq 1) then
C Parameter for N2 dissociation without SEC
re=1.098d0
de=228.7d0
c(0) = 2.70963254293d0
c(1) = 1.32620177271d-1
c(2) = 2.96757048793d-1
c(3) = 1.97112432229d-1
c(4) = -5.02002309588d-1
c(5) = 3.80734244606d-1
c(6) = 1.21001628750d0
else if (imol .eq 2) then
...
endif

```

For explanation see Section 8.2.1 and for more detailed explanation of `imol` please consult with Section 8.2.2.

Below the `if` statement for `imol` the equation(s) of the function can be given. The energy of the diatomic potential is defined as v_2 . The value of r (distance) will be taken from subroutine `ev2b` when the subroutine `ev2gm2` is called and it will give back the value of v_2 . If new variables are introduced for the equations, they must be defined. We recommend the usage of the current area of data `c` for coefficients. The current range of this area is running from `c(0)` to `c(10)`, but an extended range can be defined by modifying the following line:

```
double precision c(0:10)
```


The fitting program itself does not use the analytical gradients ($\text{dv}_{2\text{dr}}$) of the diatomic potentials, but we recommend including them into the code. It is useful, for instance, if a potential energy surface subroutine of a system is released or used in dynamics simulations. The equation(s) of calculation of gradient is within the `if` statement of `igrad`. This way the gradient is calculated only in that case if `igrad = 1`.

In that case if the user wants to use one of the LL + HL (CASSCF and correlation energies are separately fitted) options, then in the file `ev2gm2.f` there is a subroutine `ev2gm2cas` with the diatomic potential energy function, where the fitted data were obtained by CASSCF calculations. The structure of subroutines `ev2gm2` and `ev2gm2cas` are similar, but there are small differences. For instance, the `if` statement for `pairw` is not implemented yet; and a series of mixed exponential-Gaussian (MEG) variables describes the fitted diatomic potential (see eqs 16 and 17 in Section 4.5) based on geometry data point by CASSCF calculations.

The codes of *PIPFit 2022* and *PIPList* only differ in the file `pesfit.f`, thus any other modified files can be copied from the directory `src/` to the directory `scr_list/`.

8.2.4. Diatomic potentials with short-range and long-range formalism

As it was already discussed in Section 4.5, one of the diatomic descriptions contains Grimme's D3 dispersion corrections with damping function. It gives the long-range (LR) term of the diatom, but due to the nature of the applied damping function the energy goes to a finite value at short range (SR_x). Thus, the SR_x term can be considered as a difference between the "real" diatomic function and the LR term. This means that the D3 dispersion corrections (subroutines in files `dispersion.f` and `ev2b.f`) and the SR_x diatomic terms (subroutines in files `ev2gm2.f` and `ev2b.f`) will give together the diatomic potentials. Thus, it is recommended to consult with Sections 8.2.1.-8.2.3. before further modifications. Based on those descriptions the new SR_x term(s) can be introduced, and this section focuses only on the changes for dispersion correction.

The parameter `iatyp` defines which four-body system is calculated in subroutine `d3disp` in file `dispersion.f`. If there is a new four-body system, then the `if`

statement needs to be extended as it was done in Section 8.2.2. The atom numbers are parameters `iz`, for instance, in case of O_4 :

```

else if (iatyp .eq 64) then
  iz(1)=8
  iz(2)=8
  iz(3)=8
  iz(4)=8

  c6(1)=12.8d0
  c6(2)=12.8d0
  c6(3)=12.8d0
  c6(4)=12.8d0
  c6(5)=12.8d0
  c6(6)=12.8d0

!   call copyc6(5,94,c6ab,mcx)
   call edisp(94,5,4,dist,iz,c6ab,mcx,
.       rs6,rs8,alp6,alp8,e6,e8,c6)

```

The user has to provide a new `iz` block for his/her new system. The next required parameters are the C_6 values for each atom pair. In case of O_4 system, we manually set up these values for the oxygen pairs. These values originated from Grimme's (coordination number dependent) C_6 reference values for pairs of 94 elements (H-Pu) given in subroutine `copyc6`. Since the C_6 values are fixed, the subroutine `copyc6` is currently turned off. This way of treatment was chosen because the system specific C_6 values obtained from subroutine `copyc6` are geometry dependent via the coordination numbers. Thus, it leads to a difference for numerical and analytical gradients and Hessians.

To get the C_6 reference values the user has to turn on the subroutine `copyc6` by deleting the exclamation sign (!) and request the program to print out the C_6 parameters `c6ab(iat,jat,iadr,jadr,1)` in subroutine `copyc6`.

The subroutine `ev2b` (in file `ev2b.f`) the dispersion energy calculation (i.e., the subroutine `d3disp`) is called if `pairw = 3`. For any other `pairw` options the dispersion energy is zero.

For LL + HL fitting formalism, the dispersion energy correction (if `pairw = 3`) is taken into account in the HL energy correction.

9. Test inputs

The *PIPFit 2022* program requires an input file (`<name>.inp`) containing all the parameters for the fitting and a data file (`main.abn` or `<name>.abn`) 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 potential energy surface 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>.abn`) are reserved for comments; these lines are also shown in the examples below. 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 N₂O₂ potential energy surface 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=2
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 smaerl=500.0 saerl=10.0
$end
```

Sample data file for N₄ in internal coordinates:

Recall that the first 20 lines of the data file are comments.

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 N₂O₂ in Cartesian coordinates:

Recall that the first 20 lines of the data file are comments.

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
2 1349.2560
0.000000 0.000000 -0.104000
0.000000 0.000000 1.104000
0.700000 0.000000 -0.500000
-0.700000 0.000000 -0.500000

```

.....

10. References

PIPFit 2015 and *PIPFit 2022* have been used for several published potential energy surfaces. Here is a list of such applications:

imethod = 2: Permutationally invariant polynomials fit to the many-body term of the surface, where the unconnected terms are removed, and the 2-body terms are external potentials:

1. Y. Paukku, K. R. Yang, Z. Varga, and D. G. Truhlar, "Global Ab Initio Ground-State Potential Energy Surface of N_4 ", *J. Chem. Phys.*, **139**, 044309 (2013). Erratum: **140**, 019903 (2014).
2. J. Bender, P. Valentini, I. Nompelis, Y. Paukku, Z. Varga, D. G. Truhlar, T. Schwartzenuber, and G. Candler, "An Improved Potential Energy Surface and Multi-Temperature Quasiclassical Trajectory Calculations of $N_2 + N_2$ Dissociation Reactions", *J. Chem. Phys.*, **143**, 054304 (2015).
3. W. Lin, Z. Varga, G. Song, Y. Paukku, and D. G. Truhlar, "Global Triplet Potential Energy Surfaces for the $N_2(X^1\Sigma) + O(^3P) \rightarrow NO(X^2\Pi) + N(^4S)$ Reaction", *J. Chem. Phys.*, **144**, 024309 (2016).
4. Z. Varga, R. Meana-Pañeda, G. Song, Y. Paukku, and D. G. Truhlar, "Potential Energy Surface of Triplet N_2O_2 ", *J. Chem. Phys.*, **144**, 024310/1-14 (2016).
5. Y. Paukku, K. Yang, Z. Varga, G. Song, J. Bender, and D. G. Truhlar, "Potential energy surfaces of quintet and singlet O_4 ", *J. Chem. Phys.*, **147**, 034301 (2017).
6. Z. Varga, Y. Paukku, and D. G. Truhlar, "Potential energy surfaces for $O + O_2$ collisions", *J. Chem. Phys.*, **147**, 154312 (2017).
7. Y. Paukku, Z. Varga, and D. G. Truhlar, "Potential Energy Surface of Triplet O_4 ", *J. Chem. Phys.*, **148**, 124314 (2018).
8. J. Li, Z. Varga, D. G. Truhlar, and H. Guo, "Many-Body Permutationally-Invariant-Polynomial Neural-Network Potential Energy Surface for N_4 ", *J. Chem. Theory Comput.*, **16**, 4822-4832 (2020).
9. Z. Varga, Y. Liu, J. Li, Y. Paukku, H. Guo, and D. G. Truhlar, "Potential Energy Surfaces for High-Energy $N + O_2$ Collisions", *J. Chem. Phys.*, **154**, 084304 (2021).
10. Z. Varga and D. G. Truhlar, "Potential Energy Surface for High-Energy $N + N_2$ Collisions", *Phys. Chem. Chem. Phys.*, **23**, 26273-26284 (2021).

imethod = 3: Permutationally invariant polynomials fit to the global coupling surface of a state pair with unconnected terms removed:

1. Y. Shu, J. Kryven, A. G. S. de Oliveira-Filho, L. Zhao, G.-L. Song, S. L. Li, R. Meana-Pañeda, B. Fu, J. M. Bowman, and D. G. Truhlar, "Direct Diabatization and Analytic Representation of Coupled Potential Energy Surfaces and Couplings for the Reactive Quenching of the Excited $^2\Sigma^+$ State of OH by Molecular Hydrogen", *J. Chem. Phys.*, **151**, 104311 (2019).

11. Revision history

PIPFit 2015 (July 28, 2015)

Authors: Ke R. Yang, Zoltan Varga, and Donald G. Truhlar.

Version 2015 is the first distributed version of *PIPFit*.

PIPFit 2022 (Jan. 17, 2022)

Authors: Ke R. Yang, Zoltan Varga, Kelsey A. Parker, Yinan Shu, and Donald G. Truhlar.

Improvements and additions:

- New diatomic potentials were introduced for N₂, NO, and O₂.
- Grimme's D3 dispersion correction with BJ damping was added.
- Geometry sorting was added, where the boundaries are controlled with parameters `grange`.
- Parameter E_{sh} was introduced for the energy weighting scheme.
- Subroutines for A₂BC-type systems were added. This includes the monomials and polynomials up to 8th order and schemes for removal of unconnected terms and the diatomic terms from the polynomials.
- Two new weighting functions were added.
- Subroutines of AB₃-type systems were added. This includes the monomials and polynomials up to 14th order and schemes for removal of unconnected 3-body terms.

Error corrections:

- An error was fixed where the energy of one of the diatomic contributions was not printed properly into `.vfl` file for homonuclear molecules.
- An extra data counter was added to subroutine `readdat` to avoid crashing or reading wrong data if parameter `Elimit` is lower than the highest energy in the data set.
- Typos in the output file were fixed.
- A bug in the calculation of `dvec` was fixed.