JCTC Journal of Chemical Theory and Computation

Permutationally Invariant Polynomial Basis for Molecular Energy Surface Fitting via Monomial Symmetrization

Zhen Xie*,† and Joel M. Bowman*,‡

Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, Illinois 60439 and Department of Chemistry and Cherry L. Emerson Center for Scientific Computation, 1515 Dickey Drive, Emory University, Atlanta, Georgia 30322

Received September 16, 2009

Abstract: We describe a procedure to develop a fitting basis for molecular potential energy surfaces (PESs) that is invariant with respect to permutation of like atoms. The method is based on a straightforward symmetrization of a primitive monomial basis and illustrated for several classes of molecules. A numerically efficient method to evaluate the resulting expression for the PES is also described. The fitting basis is used to obtain a new PES for H_3O^+ based on roughly 62 000 ab initio energies.

1. Introduction

The representation of potential energy surfaces (PESs) that are fits to electronic energies is a long-standing goal of computational chemistry.[1−3] Progress to extend this goal beyond three and four-atom systems has been made recently by using a fitting basis that is invariant with respect to all permutations of like atoms.[4] (Also consult this paper for a review of other recent approaches to fitting PESs.) The approach makes use of powerful algorithms from computational invariant polynomial theory. The key feature of this method is to compute the *primary* and *secondary* invariants for a particular molecule permutation group. Once the primary and secondary invariants are computed, every invariant polynomial basis function can be uniquely factorized as the product of secondary invariants with a polynomial of the primary invariants (typically this polynomial is just a product of some primary invariants when constructing the invariant basis functions). The computational efficiency of this representation comes from this factorization. Since every invariant polynomial can be written as the product of two invariant polynomials, hence in the real evaluation of the potential energy function, only N multiplications are needed to evaluate all the N basis functions, once the necessary multiplication and additions are done to evaluate the primary and secondary invariants.

This method has been applied to a variety of molecules and molecular systems such as CH_5^+,[5−7] H_5^+,[8,9] C_2H_3,[10] H + CH_4,[11−13] F + CH_4,[14,15] malonaldehyde ($CHOCH_2CHO$),[16] OH + NO_2,[17] H_3O^+,[18] HO_2 + NO,[19] and $H_5O_2^+$ [20] and water dimer $(H_2O)_2$[21,22] and trimer.[23] The large set of primary and secondary polynomial for as many as 10 atom molecules was obtained with the commercial code magma.[24] This large library of fitting bases is available at the iOpenShell Web site.[25]

A much more straightforward approach to develop a permutationally invariant basis termed monomial symmetrization was briefly and only schematically described by Huang et al.[20] The method was described in more detail by Xie in his Ph.D. thesis[26] and was recently reviewed and illustrated for several molecules.[4]

The monomial symmetrization approach (MSA) appears prima facie to be much less efficient than the invariant polynomial approach, and so its presentation was mainly done as a pedagogical tool. In this paper, we describe a method to make this straightforward fitting procedure quite efficient and, thus, potentially competitive with the computational invariant polynomial approach. Another key point of the paper is to show that the implementation of the approach is straightforward and does not require access to the magma code.

* Corresponding author. E-mail: zhenxie@mcs.anl.gov.
† Argonne National Laboratory.
‡ Emory University.

10.1021/ct9004917 © 2010 American Chemical Society
Published on Web 11/23/2009

The paper is organized as follows: A brief review of the monomial symmetrization approach is presented in Section 2, followed by the algorithm to speed up the function evaluation process. The method is exemplified by an application for the PES of H_3O^+ in Section 3. A brief discussion including some possible new directions using this approach is given in Section 4. A summary and conclusions are given in Section 5.

2. Theoretical Background and Computational Details

2.1. Coordinate Representation of the Potential Energy Surface. The molecular PES, which we denote as V, should respect the key invariant properties of the physics. The well-known properties are overall translational and rotational invariances. A third property, one that is obvious but less well-known, is invariance with respect to all permutations of like atoms. This invariance has been noted in the classic book by Murrell and co-workers[1] and is at the heart of the approach recently reviewed by Braams and Bowman.[4] As noted in that review and the primary sources cited there, the ideal choice of coordinates for V should enable these invariance properties to be "built in". The full set of internuclear distances (of which there are $n(n - 1)/2$ for a system of n atoms) almost satisfies this property. That is, these distances are invariant with respect to overall translation and rotation of the molecular system. They are not invariant with respect to permutation of like atoms; however, they are closed under these permutations, and this property has been exploited to make basis functions for the representation of V obey that invariance.

Before we show how this can be done using the MSA,[4,20,26] we make two important remarks. First, the full set of internuclear distances for molecules with more than four atoms is known as a redundant set of coordinates, since it is well-known that only $3n - 6$ internal degrees of freedom are needed to specify an n atom molecular configuration. Thus, for n greater than four, there are more internuclear distances than internal degrees of freedom, and in that sense, the choice of all internuclear distances enlarges the space of variables for $n > 4$. Second, and independent of the value of n, internuclear distances are not a good choice of variables for a monomial or polynomial basis to represent V. This is because these distances become arbitrarily large when fragments form, and thus, V would diverge erroneously in these regions. Therefore, we and others use simple transformed variables that go to a constant in these regions. We have used several choices for these variables, some of which will be given below. A specific choice is made when we consider an application to the H_3O^+ PES. However, to investigate how these variables permute under permutations of like atoms, we need only consider the set of internuclear distances, and so we do so here.

We begin by labeling the n atoms in a molecule as 1, ..., n, and the $n(n - 1)/2$ internuclear distances are given in lexical order as

$$(r_{1,2}, ..., r_{1,n}, r_{2,3}, ..., r_{2,n}, ..., r_{n-1,n})$$

At this point, it is worth noting that one choice of associated Morse variables that we have used extensively is $y_{i,j} = \exp(-r_{i,j}/\lambda)$. Also, we introduce a shorthand notation for these distances (or Morse variables), x_l, $l = 1, n(n - 1)/2$, where the x_l are ordered according to the lexical ordering of $r_{i,j}(y_{i,j})$.

To proceed it is useful to consider a specific example of a tetraatomic molecule, A_4, and the six internuclear distances are

$$(r_{1,2}, r_{1,3}, r_{1,4}, r_{2,3}, r_{2,4}, r_{3,4})$$

and associated variables

$$(x_1, x_2, x_3, x_4, x_5, x_6)$$

The usual expression for V in terms of these variables is

$$V(x_1, x_2, x_3, x_4, x_5, x_6) = \sum_{a+b+c+d+e+f=0}^{k} C_{a,b,c,d,e,f}\, x_1^a x_2^b x_3^c x_4^d x_5^e x_6^f \tag{2.1}$$

where a, b, c, d, e, and f are all non-negative integers, and k is a positive integer that sets the maximum as the sum of all the exponents. (The coefficients $C_{a,b,c,d,e,f}$ would typically be determined by a standard linear least-squares fit to a data set of ab initio electronic energies.) Clearly this expression is not invariant with respect to permutations of like atoms. However, one can easily modify the expression so that it is. This procedure involves symmetrizing the monomials, as discussed in detail elsewhere,[4,26] which yields a sum of monomials with a single coefficient, which we denote as $D_{a,b,c,d,e,f}$.[4]

Continuing with the tetraatomic example, consider an example permutation of the four atoms where the original atom order (1, 2, 3, 4) is permuted to (4, 2, 1, 3). The internuclear distances change from

$$(r_{1,2}, r_{1,3}, r_{1,4}, r_{2,3}, r_{2,4}, r_{3,4})$$

to

$$(r_{4,2}, r_{4,1}, r_{4,3}, r_{2,1}, r_{2,3}, r_{1,3}) \equiv (r_{2,4}, r_{1,4}, r_{3,4}, r_{1,2}, r_{2,3}, r_{1,3})$$

or in the "x" notation:

$$(x_1, x_2, x_3, x_4, x_5, x_6)$$

maps to

$$(x_5, x_3, x_6, x_1, x_4, x_2)$$

Thus the monomial $x_1^a x_2^b x_3^c x_4^d x_5^e x_6^f$ maps to $x_5^a x_3^b x_6^c x_1^d x_4^e x_2^f$ ($\equiv x_1^d x_2^f x_3^b x_4^e x_5^a x_6^c$).

In order to complete the symmetrization, all permutations must be considered. We indicate the final result by

$$V = \sum_{a+b+c+d+e+f=0}^{k} D_{a,b,c,d,e,f}\, \mathcal{S}[x_1^a x_2^b x_3^c x_4^d x_5^e x_6^f] \tag{2.2}$$

where $\mathcal{S}$ is the operator that symmetrizes monomials. Examples of sums of symmetrized monomials were given for A_2B_2 and A_3B molecules elsewhere.[4] This was done by explicitly enumerating all the permutations and by showing how the internuclear distances permute. For a general molecule, the total number of permutations is just the direct

Scheme 1. Algorithm 1

```
Algorithm 1: Monomial symmetrization approach
Data: $k$ ;                                  // maximum total degree of monomials
Data: $\mathscr{P}$ ;                                            // permutation group
Result: $B$ ;                                     // invariant basis functions set
$B \leftarrow \emptyset$;
$m \leftarrow [0 \cdots 0]$;
while $deg(m) \leq k$ do
    $\text{orb}(m) \leftarrow \{p_i \cdot m | p_i \in \mathscr{P}\}$;              // $p_i$: a permutation
    $B \leftarrow B \cup \text{orb}(m)$;
    $m \leftarrow m + 1$ ;                                 // get the next monomial
end
```

product of permutations of sets of like atoms. For example, for molecule A_nB_m, the order of the direct product symmetric group is $n!m!$, and the full set of permutations of the sets of like atoms and how those map onto permutations of internuclear distances must be determined. We note that if the resulting monomials are reordered in lexical form 1, 2, 3, etc., then the effect of the mappings is equivalent to permuting the powers of the original "seed" monomial. We use this convention henceforth.

The generation of the permutation of internuclear distances (or equivalently the powers) has been automated with software we have written for any molecule,[27] and Algorithm 1 summarizes the process.

2.2. Algorithm for Efficient Basis Evaluation. Although the potential energy function as shown in eq 2.2 has the permutationally invariant property and we have Algorithm 1 to generate the basis function set, the terms (the symmetrized sum of monomials) are costly to evaluate, and so this representation is not nearly as efficient as the one based on primary and secondary invariants.[4] Clearly there exist strategies that can be used to speed up the evaluation of the symmetrized sum of monomials, which, henceforth, we denote as invariant polynomials. If the invariant polynomials could be factored into lower order invariant polynomials, considerable speedup would result. To our knowledge, however, this factorization cannot be done easily (primary and secondary invariants essentially do this, but the computation of primary and secondary invariants is difficult and complicated, as is the factorization).

Here we describe and demonstrate a less ambitious factorization scheme, which is a binary factorization plus remainder method, where the remainder is one or more invariant polynomials that may have already been computed. To describe this approach, we introduce some new notation and nomenclature. As noted already, an invariant polynomial (of some total degree) is the sum of all the monomials generated by acting all the possible permutations on a seed monomial. We denote this set of monomials as the orbit of the seed monomial m, and denote it as orb(m), which could be represented as $\text{orb}(m) = \{m^{p_i} | p_i \in \mathscr{P}\}$. Hence eq 2.2 can be simplified as

$$V(x_1, x_2, x_3, x_4, x_5, x_6) = \sum_{a+b+c+d+e+f=0}^{k} D[abcdef]\Big(\sum_{\mathscr{P}} \text{orb}([abcdef])\Big) \tag{2.3}$$

where $[abcdef]$ is a shorthand notation for a general monomial $x_1^a x_2^b x_3^c x_4^d x_5^e x_6^f$.

Clearly, if $\sum_{\mathscr{P}} \text{orb}([abcdef])$ could be evaluated effectively, then the whole potential energy function could be computed effectively. One way to speed this process is to evaluate or build these orbits recursively, that is, to express the "later" orbits as some simple expression from the "earlier" ones, where the terms "later" and "earlier" are yet to be defined as the ordering of monomials.

For monomials with different total degrees, it is easy to order them according their total degree. For monomials with the same total degree, we order them according to the number of nonzero powers. For convenience, we order a monomial with more nonzero powers before another one with fewer nonzero elements but with same total degree, for instance, [010101] < [030000]. If two monomials have the same total degree and the same number of nonzero elements, then we order these monomials according to their lexicographical order, for example, [010101] < [101010]. This ordering scheme is made for later computational convenience. Similar to this ordering of monomials, we can also order the polynomials or orbits, and here the polynomials are ordered according to the "largest" monomials (ordered in the last position according to the monomial ordering) in it. Note that for molecules with greater than four atoms, the maximum total degree is typically much less than the number of internuclear distances there are no polynomials with, where every x_i has a nonzero power.

To illustrate the approach consider the polynomial $x_1^2 + x_2^2$, which is invariant with respect to the permutation of x_1 and x_2, and note that it cannot be simply decomposed as a product of two lower order polynomials. (It is, in fact, a primary invariant polynomial.) However, it can be given as a low-order product of invariant polynomials with a remainder:

$$x_1^2 + x_2^2 = (x_1 + x_2) \times (x_1 + x_2) - x_1x_2 - x_1x_2$$

As a result, if the polynomial $x_1 + x_2$ and x_1x_2 are both in the invariant basis function sequence and both appear "earlier" than $x_1^2 + x_2^2$, then $x_1^2 + x_2^2$ could be easily evaluated with one multiplication and two subtractions. To evaluate $x_1^2 + x_2^2$ directly, we need two multiplications and one addition. Since multiplication is more expensive than addition or "subtraction", the decomposition should speed the evaluation. If such a decomposition can be found for a general invariant polynomial, the evaluation of the invariant basis function will be greatly sped up. Now the question comes to the existence and the uniqueness of this kind of decomposition. Unfortunately, neither of these is guaranteed. For a successful decomposition, which we will denote as a "usable decomposition", not only must the two polynomials that form the product be of a lower degree (which, of course, they are) but also the remainder polynomial must come "before" the polynomial becomes decomposed. This is not always the case. Another issue for the decomposition is the uniqueness. There exists various ways to factorize a high-degree monomial as a low-degree one; for example, a monomial [012101] can be factorized as

$$[012101] = [010001] \times [002100]$$

or

$$[012101] = [011101] \times [001000]$$

Scheme 2. Algorithm 2

Algorithm 2: Trial-and-error decomposition algorithm for all the permutationally invariant basis functions

```
Data: $B$ ;                                   // invariant basis functions set
Result: $D$ ;                                 // decomposition of $b_i \in B$
Result: $M$ ;                                 // monomial set
$B_P \leftarrow \emptyset$ ;                  // single term polynomial set
// populate the single term polynomial set
foreach $b_i \in B$ do
    if $|b_i| = 1$ then
        $B_P \leftarrow B_P \cup b_i$ ;
    end
end
// trying to decompose every basis function
$M \leftarrow \emptyset$;
$i_M \leftarrow 0$ ;                          // size index of $M$
foreach $b_i \in B$ do
    if $b_i \in B_P$ then
        // deal with single term basis function
        $M \leftarrow M \cup b_i$;
        $i_M \leftarrow i_M + 1$;
        $D \leftarrow D \cup [i, 0, i_M]$;        // 0 indicates failed decomposition
    else
        // deal with multiple terms basis function
        if $(\exists b_m, b_n, b_{l_1}, \cdots \leq b_i) \vee (b_i = b_m \cdot b_n - b_{l_1} - \cdots)$ then
            // the basis function has a usable decomposition
            $D \leftarrow D \cup [i, 1, m, n, l_1, \cdots]$;      // 1 indicates successful
            decomposition
        else
            // the basis function has no usable decomposition
            $M \leftarrow M \cup \mathrm{mono}(b_i)$;             // save the monomial set
            $D \leftarrow D \cup [i, 0, i_M + 1, \cdots, i_M + |b_i|]$; // record the sum of monomials
            $i_M \leftarrow i_M + |b_i|$;
        end
    end
end
```

or any other two monomials whose sum of exponents are equal to the original exponents. Different factorizations of the monomials result in different remainder polynomials, and different factorizations may or may not lead to usable decomposition of a high-degree polynomial.

One practical approach to this factorization of polynomials is to list all the possible factorizations of the seed (it can be any monomial of the polynomial) of a high-degree polynomial. Denoting the high-degree polynomial as p_1, and one monomial in p_1 as m_1, we have $p_1 = \mathrm{orb}(m_1)$. Note that we do not distinguish the set of monomials as $\mathrm{orb}(m_1)$ and the polynomial, which is the sum of all the monomials in this set. Suppose p_2 is a polynomial arranged before p_1 in the invariant polynomial basis sequence. Then we can scan over all the monomials in p_2, to determine if there is one monomial (m_2) in p_2 that is a factor of m_1, then we can factorize m_1 as $m_1 = m_2 m_3$ and find the orbit of m_3 ($\mathrm{orb}(m_3)$) in the polynomial basis. Further suppose $p_3 = \mathrm{orb}(m_3)$. Then we can form the product $p_2 p_3$. The difference between $p_2 p_3$ and p_1 should be a sum of orbits of some other monomials (sum of some other invariant polynomials), and we try to minimize the number of remainder polynomials already computed. If it is not possible to find a usable decomposition, then we have to evaluate the polynomial by evaluating all monomials. All monomials within a polynomial that has no usable decomposition are pushed into a queue. The speed up for this process is determined partially by the size of the monomial queue. If most of the polynomials can be decomposed successfully, the size of the monomial queue is small, and the efficiency of the approach is high. For monomials in the queue, the evaluation process could and is sped up by factorizing the later ones as the product of earlier ones.

In summary, all the possible monomials with total degrees less than some threshold are enumerated and grouped into orbits by the action of permutations. The sum of such monomials in an orbit is an invariant polynomial. They are arranged according to the polynomial ordering scheme, as defined above. Then the polynomials are decomposed into the product of two polynomials, and possibly subtracting a small number of same-order polynomials, if this is possible. Otherwise, the polynomial is kept as the sum of monomials. A factorization step is performed for every monomial from the previous step and evaluated as the product of two previous ones. In the following section, an example will be given to show the process.

The algorithms just described above are summarized as Algorithm 2, 3, and 4.

2.3. Example: Basis Functions for H_3O^+. H_3O^+ is an important molecule in chemistry, and it has three identical hydrogen atoms which make it a good example to illustrate the MSA process. Of course the results shown apply to any A_3B molecule.

The three H atoms are labeled as 1, 2, and 3, and the O atom is labeled as 4. Table 1 shows all the permutation actions on the internuclear distances and also on a general monomial [*abcdef*]. For demonstration purposes, *a*, ..., *f* are all different integers, in reality, some of them may be equal.

Scheme 3. Algorithm 3

```
Algorithm 3: Decomposition of a basis function (polynomial) $b_i$
Data: $B, b_i \in B$;      // invariant basis functions set, and one basis
Result: $b_i = b_m \cdot b_n - b_l \cdots, (m, n, l, \cdots \le i)$ or fail
$D \leftarrow \emptyset$;                           // usable decomposition set
$m_1 \leftarrow \mathrm{seed}(b_i)$;                                // $m_1 \in \mathrm{mono}(b_i)$
foreach $b_j \in B \vee b_j < b_i$ do
    if $\exists m_2 \in mono(b_j) \vee m_1 = m_2 \cdot m_3$ then
        $b_m \leftarrow b_j$;
        $b_n \leftarrow \mathrm{orb}(m_3)$;
        $p \leftarrow b_m \times b_n - b_i$;
        if $(\exists b_{l_1}, \cdots \in B) \vee (b_{l_1}, \cdots < b_i) \vee (p = \sum_i b_{l_i})$ then
            $D \leftarrow D \cup [i, m, n, l_1, \cdots]$;
        end
    end
end
if $D = \emptyset$ then
    return fail;
else
    return $d \in D : |d| \le |d_i|, \forall d_i \in D$; // return the simplest decomposition
end
```

Scheme 4. Algorithm 4

```
Algorithm 4: Decomposition of monomials
Data: $M$ ;                                              // a monomial set
Result: $H$ ;                            // the decomposition of monomials
$H \leftarrow \emptyset$;
foreach $m \in M$ do
    if $(\exists m_i, m_j \in M) \vee (m_i, m_j < m) \vee (m = m_i \cdot m_j)$ then
        $H \leftarrow H \cup [i, j]$;
    else
        $H \leftarrow H \cup m$;
    end
end
```

Table 1. Permutation of the H_3O^+ Molecule and Its Action on a General Monomial [*abcdef*]

permutation	atom labels	internuclear distance vector	permuted monomial
p_1	1 2 3 4	$(x_1, x_2, x_3, x_4, x_5, x_6)$	[*abcdef*]
p_2	1 3 2 4	$(x_2, x_1, x_3, x_4, x_6, x_5)$	[*bacdfe*]
p_3	2 1 3 4	$(x_1, x_4, x_5, x_2, x_3, x_6)$	[*adebcf*]
p_4	2 3 1 4	$(x_4, x_1, x_5, x_2, x_6, x_3)$	[*bdface*]
p_5	3 1 2 4	$(x_2, x_4, x_6, x_1, x_3, x_5)$	[*daebfc*]
p_6	3 2 1 4	$(x_4, x_2, x_6, x_1, x_5, x_3)$	[*dbfaec*]

Consequently, the orbit of a monomial [*abcdef*] may have less than six monomials, as shown in Table 2.

Table 2 lists all the invariant polynomial basis for molecule H_3O^+ up to degree 3. All those polynomials that have usable decomposition under the polynomial ordering scheme are indicated as a product of two other basis functions less some other basis functions. As for those that do not have a usable decomposition, the decomposition is written as the sum of monomials.

The factorization of all the monomials is shown in Table 3.

As we can see from Table 2, many basis functions have usable decompositions, i.e., can be expressed as the product of two lower degree basis functions, less some other basis functions that are at the same degree but come before the current invariant basis. As a result, the evaluation of those basis functions costs one multiplication with some addition. For those monomials, almost all could be factorized as a product of two previous ones, thus, significantly reducing the evaluation time—especially when the total degree of monomial increases. The monomials that cannot be factorized are those single variable terms, and they are trivial to evaluate.

2.4. Applications to Other Molecules. Relevant information about the monomial and the symmetrized monomial basis for a variety of molecules and total degree is given in Table 4. Columns three and four give the size of the monomial and symmetrized monomial (polynomial) basis for each example. The reduction in size for the polynomial basis is large. However, it is not as large as the order of the corresponding symmetric group; this reduction factor is approached only as the total degree approaches infinity. For example, for A_4, the maximum possible reduction is 24, and as seen for degree 8, the actual reduction is 15.4. Next, consider the number of polynomials and the percentage of the of all polynomials that have a usable decomposition shown in column five and six. The percent ranges from almost 100% to a low of 51.95%. The number of remainder monomials to be evaluated and the number that can be factored are given next. Finally, an "efficiency" column is included, which contains two values. The ratio of total monomials to total polynomials is given in parentheses, and the first number is the ratio of monomials to the sum of usable decomposition plus the monomials left.

Perusal of this table leads to the conclusion that the current factorization method works better for molecules with two or three identical atoms than for molecules with more identical atoms. On the other hand, the size of polynomial basis decreases by a larger factor for such molecules (since the order of the symmetric group increases).

The lack of usable decomposition of an arbitrary (invariant) polynomial in the present algorithm comes from the polynomial ordering scheme. It is not difficult to show that any high-degree invariant polynomial constructed by using monomial symmetrization approach can be represented as the product of two low-degree polynomials and as a remainder invariant polynomial of the same degree. If the polynomial ordering scheme can ensure that polynomials need to be subtracted are always come before the target polynomial, then the inefficiency factor could be removed, and we can claim that the monomial symmetrization approach is almost as fast as the computation invariant theory based approach. However, this ordering scheme is difficult to construct. A simple example may show the difficulty. Suppose that $x_1 + x_2$, $x_1^2 + x_2^2$ and x_1x_2 are invariant polynomials, with respect to the permutation of x_1 and x_2. Both $x_1^2 + x_2^2$ and x_1x_2 are of degree 2, and $x_1 + x_2$ is of degree 1, which always comes before the other two polynomials. $x_1^2 + x_2^2$ could be decomposed as

$$x_1^2 + x_2^2 = (x_1 + x_2) \times (x_1 + x_2) - x_1x_2 - x_1x_2 \tag{2.4}$$

and x_1x_2 could be decomposed at

$$x_1x_2 = (x_1 + x_2) \times (x_1 + x_2) - (x_1^2 + x_2^2) - x_1x_2 \tag{2.5}$$

It is easy to rearrange eq 2.5 as

$$2x_1x_2 = (x_1 + x_2) \times (x_1 + x_2) - (x_1^2 + x_2^2) \tag{2.6}$$

As can be seen, no matter what the polynomial ordering scheme is (whether x_1x_2 comes before $x_1^2 + x_2^2$ or after it),

Table 2. Invariant Polynomial Basis for Molecule H_3O^+ up to Degree 3

ID	degree	invariant polynomial basis	decomposition
B_0	0	[000000]	M_0
B_1	1	[000001] + [000010] + [001000]	$M_1 + M_2 + M_3$
B_2	1	[000100] + [010000] + [100000]	$M_4 + M_5 + M_6$
B_3	2	[000011] + [001001] + [001010]	$M_7 + M_8 + M_9$
B_4	2	[001100] + [010010] + [100001]	$M_{10} + M_{11} + M_{12}$
B_5	2	[000101] + [000110] + [010001] + [011000] + [100010] + [101000]	$B_1 \times B_2 - B_4$
B_6	2	[010100] + [100100] + [110000]	$M_{13} + M_{14} + M_{15}$
B_7	2	[000002] + [000020] + [002000]	$B_1 \times B_1 - B_3 - B_3$
B_8	2	[000200] + [020000] + [200000]	$B_2 \times B_2 - B_6 - B_6$
B_9	3	[001011]	M_{16}
B_{10}	3	[001101] + [001110] + [010011] + [011010] + [100011] + [101001]	$M_{17} + M_{18} + M_{19} + M_{20} + M_{21} + M_{22}$
B_{11}	3	[000111] + [011001] + [101010]	$B_2 \times B_3 - B_{10}$
B_{12}	3	[010110] + [011100] + [100101] + [101100] + [110001] + [110010]	$M_{23} + M_{24} + M_{25} + M_{26} + M_{27} + M_{28}$
B_{13}	3	[110100]	M_{29}
B_{14}	3	[010101] + [100110] + [111000]	$B_1 \times B_6 - B_{12}$
B_{15}	3	[000012] + [000021] + [001002] + [001020] + [002001] + [002010]	$B_1 \times B_3 - B_9 - B_9 - B_9$
B_{16}	3	[002100] + [010020] + [100002]	$B_1 \times B_4 - B_{10}$
B_{17}	3	[000102] + [000120] + [010002] + [012000] + [100020] + [102000]	$B_2 \times B_7 - B_{16}$
B_{18}	3	[001200] + [020010] + [200001]	$B_2 \times B_4 - B_{12}$
B_{19}	3	[000201] + [000210] + [020001] + [021000] + [200010] + [201000]	$B_1 \times B_8 - B_{18}$
B_{20}	3	[010200] + [020100] + [100200] + [120000] + [200100] + [210000]	$B_2 \times B_6 - B_{13} - B_{13} - B_{13}$
B_{21}	3	[000003] + [000030] + [003000]	$B_1 \times B_7 - B_{15}$
B_{22}	3	[000300] + [030000] + [300000]	$B_2 \times B_8 - B_{20}$

Table 3. Factorization of the Monomials from Those Basis Functions that do not have Usable Decomposition for H_3O^+ Molecule

ID	Monomial	factorization	ID	Monomial	factorization
M_0	[000000]	1	M_{15}	[110000]	$M_5 \times M_6$
M_1	[000001]	x_6	M_{16}	[001011]	$M_1 \times M_9$
M_2	[000010]	x_5	M_{17}	[001101]	$M_1 \times M_{10}$
M_3	[001000]	x_3	M_{18}	[001110]	$M_2 \times M_{10}$
M_4	[000100]	x_4	M_{19}	[010011]	$M_1 \times M_{11}$
M_5	[010000]	x_2	M_{20}	[011010]	$M_3 \times M_{11}$
M_6	[100000]	x_1	M_{21}	[100011]	$M_2 \times M_{12}$
M_7	[000011]	$M_1 \times M_2$	M_{22}	[101001]	$M_3 \times M_{12}$
M_8	[001001]	$M_1 \times M_3$	M_{23}	[010110]	$M_2 \times M_{13}$
M_9	[001010]	$M_2 \times M_3$	M_{24}	[011100]	$M_3 \times M_{13}$
M_{10}	[001100]	$M_3 \times M_4$	M_{25}	[100101]	$M_1 \times M_{14}$
M_{11}	[010010]	$M_2 \times M_5$	M_{26}	[101100]	$M_3 \times M_{14}$
M_{12}	[100001]	$M_1 \times M_6$	M_{27}	[110001]	$M_1 \times M_{15}$
M_{13}	[010100]	$M_4 \times M_5$	M_{28}	[110010]	$M_2 \times M_{15}$
M_{14}	[100100]	$M_4 \times M_6$	M_{29}	[110100]	$M_4 \times M_{15}$

there is always one nonusable decomposition. Unfortunately, either x_1x_2 or $x_1^2 + x_2^2$ must be evaluated explicitly. (One or the other is in fact a primary invariant.)

In another case, suppose that we have invariants:

$$\begin{aligned} A &= x_1^4 + x_2^4 \\ B &= x_1^2x_2^2 \\ C &= x_1x_2^3 + x_1^3x_2 \end{aligned} \tag{2.7}$$

and they could be decomposed as

$$\begin{aligned} A &= (x_1^2 + x_2^2) \times (x_1^2 + x_2^2) - (x_1^2x_2^2 + x_1^2x_2^2) \\ B &= (x_1x_2^2 + x_1^2x_2) \times (x_1 + x_2) - (x_1x_2^3 + x_1^3x_2) - x_1^2x_2^2 \\ C &= (x_1 + x_2) \times (x_1^3 + x_2^3) - (x_1^4 + x_2^4) \end{aligned} \tag{2.8}$$

$x_1 + x_2$, $x_1^2 + x_2^2$, $x_1^3 + x_2^3$ and $x_1x_2^2 + x_1^2x_2$ are at a lower degree compared to A, B, and C, and so their values can be assumed to be known. However, the remainders do contain A, B, and C and we wish to "disentable" these expressions. To do so first rewrite these equations as

$$\begin{aligned} A &= \alpha - 2B \\ B &= \beta - C - B \\ C &= \gamma - A \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} \alpha &= (x_1^2 + x_2^2) \times (x_1^2 + x_2^2) \\ \beta &= (x_1x_2^2 + x_1^2x_2) \times (x_1 + x_2) \\ \gamma &= (x_1 + x_2) \times (x_1^3 + x_2^3) \end{aligned} \tag{2.10}$$

and then rewrite them in matrix form as

$$\begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{2.11}$$

This can be solved easily as

$$\begin{aligned} A &= \frac{1}{2}\alpha - \frac{1}{2}\beta + \frac{1}{2}\gamma \\ B &= \frac{1}{4}\alpha + \frac{1}{4}\beta - \frac{1}{4}\gamma \\ C &= -\frac{1}{2}\alpha + \frac{1}{2}\beta + \frac{1}{2}\gamma \end{aligned} \tag{2.12}$$

This indicates that it is possible to further decouple intertwined polynomials and to obtain their usable decomposition simultaneously by solving a linear system. Solving the linear system is an extra cost for the decomposition, but it is an one-time cost and it will not be inherited in the future polynomial evaluation process. As a result, we can further speed up the polynomial evaluation process. We plan to investigate an implementation of this algorithm in the future.

3. Application to a Potential Surface for H_3O^+

The methods described in the previous section to automate the symmetrization of the monomial basis and to evalu-

Table 4. Information on Invariant Polynomial Basis Function Decomposition

molecule	max deg	tot mono.	tot poly.	usable decomp.	usable decomp. %	mono left	mono fact	efficiency
A_3	6	84	23	19	82.61%	8	4	3.11 (3.65)
A_3	8	165	41	37	90.24%	8	4	3.67 (4.02)
A_2B	6	84	50	46	92.00%	5	1	1.65 (1.68)
A_2B	8	165	95	91	95.79%	5	1	1.72 (1.74)
A_4	6	924	72	60	83.33%	88	81	6.24 (12.83)
A_4	8	3003	195	182	93.33%	112	105	10.21 (15.40)
A_3B	6	924	196	185	94.39%	33	26	4.24 (4.71)
A_3B	8	3003	590	579	98.14%	33	26	4.91 (5.09)
A_2B_2	6	924	291	282	96.91%	18	11	3.08 (3.18)
A_2B_2	8	3003	882	873	98.98%	18	11	3.37 (3.40)
A_2BC	6	924	502	494	98.41%	11	4	1.83 (1.84)
A_2BC	8	3003	1589	1581	99.50%	11	4	1.89 (1.89)
A_5	6	8008	140	94	67.14%	2368	2357	3.25 (57.20)
A_5	8	43758	580	443	76.38%	10158	10147	4.13 (75.44)
A_4B	6	8008	495	437	88.28%	807	796	6.44 (16.18)
A_4B	8	43758	2327	2216	95.23%	1953	1942	10.50 (18.80)
A_3B_2	6	8008	889	838	94.26%	390	367	6.52 (9.01)
A_3B_2	8	43758	4343	4249	97.84%	876	853	8.54 (10.08)
A_3BC	6	8008	1603	1565	97.63%	161	150	4.64 (5.00)
A_3BC	8	43758	8163	8121	99.49%	185	174	5.27 (5.36)
A_2B_2C	6	8008	2304	2278	98.87%	72	61	3.41 (3.48)
A_2B_2C	8	43758	11910	11884	99.78%	72	61	3.66 (3.67)
A_2BCD	6	8008	4264	4249	99.65%	22	11	1.87 (1.88)
A_2BCD	8	43758	22734	22719	99.93%	22	11	1.92 (1.92)
A_5B_2	4	12650	218	148	67.89%	4997	4975	2.46 (58.03)
A_5B_2	6	296010	2651	2080	78.46%	78066	77924	3.69 (111.66)
A_6B_3	3	9139	77	40	51.95%	5485	5448	1.65 (118.69)
A_6B_3	4	91390	327	208	63.61%	47533	47496	1.91 (279.48)

ate the subsequent terms by means of the decomposition procedure are applied to the PES of H_3O^+. There exists PESs for H_3O^+,[18,28,29] and an unpublished PES for this cation was obtained using the invariant polynomial approach previously[26] and was used as the "benchmark" PES both in terms of precision and speed of evaluations. The precision and speed of evaluation of the monomial symmetrization approach will be tested against this benchmark. In these fits, the internuclear distances (x) are replaced by the variables $(1.0 - \exp(-0.2x))^2$. This is a slightly different version of Morse variables[4] with the same goal to achieve a physically correct description, i.e., let the potential energy function approach to a constant in the limit of asymptotically large values of x.

Low-level direct dynamics calculations were run at several total energies to generate 61 604 H_3O^+ configurations. At these configurations, electronic energies were then obtained using the CCSD(T) method with aug-cc-pVTZ basis, as implemented in the code MOLPRO.[30] The energy range for these configurations is 29 674 cm^{-1} (84.84 kcal/mol). The maximum total degree for the polynomial basis is set to be 6. This is the same total degree used for the invariant polynomial fit. Since both approaches have the same number of basis functions, the number of coefficients is also the same. Note that the coefficient values are different because the basis functions from the two approaches differ. The coefficients for both fits were obtained with standard least-squares codes. The final root-mean-square error for both fitting approaches is 29 cm^{-1}, with the maximum absolute residual value deviation at 0.16 cm^{-1}. As expected, almost no difference is observed regarding the accuracy of these two approaches.

After obtaining the coefficients, we evaluated the energies for the molecule configurations in the original data set used for fitting three times. We obtained 184 812 potential function calls in total. The computational invariant theory approach took 8.77 s, and the modified monomial symmetrization approach took 9.50 s. It is about 8% less efficient for the monomial symmetrization approach, which is almost negligible. Comparing the details of these two approaches, one can see that slightly more time is consumed in evaluating and summing the monomials from those polynomial basis functions that cannot be effectively decomposed. For the H_3O^+ molecule, the number of monomials does not change when the maximum total degree of polynomial basis functions get higher and higher. As a result, the timing difference would be virtually the same.

4. Discussion

Perhaps the most striking practical effect of symmetrizing the monomial expression for the potential, eq 2.1, is the very large reduction in the number of terms and, hence, unknown linear coefficients to be determined. This was illustrated in Table 4 where the number of monomials in the unsymmetrized expression for V is compared to the number of symmetrized polynomials. The ratio of these two numbers approaches the order of the symmetric group in the limit of infinitely many terms in both expressions; however, as seen even for modest total orders, the reduction can be very big. Practically, this has allowed us to consider molecules with up to nine atoms and to deal with a relatively small linear algebra least-squares minimization. This very practical advantage is realized by using the full permutation group of a molecule.

This practical advantage in exploiting the full permutational symmetry also holds for molecular complexes. For example, consider the water dimer and trimer, where the order of the symmetric group is $4!2! = 48$ and $6!3! = 4\,320$, respectively. PESs for both the water dimer and trimer, using the full permutational symmetry, have been reported.[21,22,31] However, physically one knows that, for temperatures or total energies of interest, most of these permutations are infeasible, i.e., they must surmount high potential barriers to be realized. Thus, it would seem that using the full symmetric group, while greatly reducing the number of terms in V, is also physically irrelevant. Ideally one would like a representation that both greatly reduces the number of terms in V and only describes the feasible permutations. (In the case of water clusters, this would be the permutations of the H atoms associated with each monomer unit.)

If one is going to continue to use all the internuclear distances (probably with a standard many-body representation of the potential), then it seems clear that one will have to restrict the range of powers of some of the internuclear distances. These presumably would be the monomer internuclear distances, which undergo small amplitude motion. In fact, a rather primitive version of this type of restriction was used in a monomial representation of the global potential of H_2CO for the CO internuclear distance.[32] Further work along these lines is planned.

Also, as the observant reader has noted, the use of the Morse-type variables introduces a nonlinear range parameter. Generally we have used a single-range parameter, typically 2−3 Bohr, for all variables. In principle, this parameter could be optimized and also made specific for a given set of internuclear distances, for example, all OH, HH, and OO distances. Doing so would turn the straightforward linear-least-squares optimization into a combination of linear and nonlinear optimizations. This may be worth considering; however, it involves a substantial increase in numerical complexity.

Finally we note that the library of primary and secondary invariant polynomials referred to in the paper does not include analytical expressions for partial derivatives. This may be added in the future; however, it will involve considerable symbolic computation. Clearly, using the monomial symmetrization described here makes this task much easier for the user. We plan to do this in the near future.

5. Conclusions

We have presented a straightforward monomial symmetrization scheme for the representing a multidimensional potential energy surface. We have proposed one scheme for efficient evaluation of the results terms in the representation and illustrated the approach by fitting roughly 62 000 electronic energies for H_3O^+. We also discussed several areas for future research. The codes described herein can be obtained by contacting either author.

Acknowledgment. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, United States Department of Energy, under contract no. DE-AC02-06CH11357 and the Office of Basic Energy Sciences, contract no. DE-FG02-97ER14782.

Note Added after ASAP Publication. This article was released ASAP on November 23, 2009. The URL in reference 27 was updated, and the correct version was posted on November 30, 2009.

References

(1) Murrell, J. N. *Molecular Potential Energy Functions*; J. Wiley: Chichester, U.K., 1984.

(2) Schatz, G. C. *Rev. Mod. Phys.* **1989**, *61*, 669–688.

(3) Hollebeek, T.; Ho, T. S.; Rabitz, H. *Annu. Rev. Phys. Chem.* **1999**, *50*, 537–570.

(4) Braams, B. J.; Bowman, J. M. *Int. Rev. Phys. Chem.* **2009**, *28*, 577−606.

(5) Jin, Z.; Braams, B.; Bowman, J. *J. Phys. Chem. A* **2006**, *110*, 1569–1574.

(6) Christoffel, K.; Jin, Z.; Braams, B.; Bowman, J. *J. Phys. Chem. A* **2007**, *111*, 6658–6664.

(7) Mann, J.; Xie, Z.; Savee, J.; Braams, B.; Bowman, J.; Continetti, R. *J. Am. Chem. Soc.* **2008**, *130*, 3730.

(8) Xie, Z.; Braams, B.; Bowman, J. *J. Chem. Phys.* **2005**, *122*, 224307.

(9) Acioli, P.; Xie, Z.; Braams, B.; Bowman, J. *J. Chem. Phys.* **2008**, *128*, 104318.

(10) Park, W.; Park, J.; Park, S.; Braams, B.; Chen, C.; Bowman, J. *J. Chem. Phys.* **2006**, *125*, 081101.

(11) Zhang, X.; Braams, B.; Bowman, J. *J. Chem. Phys.* **2006**, *124*, 021104.

(12) Xie, Z.; Bowman, J.; Zhang, X. *J. Chem. Phys.* **2006**, *125*, 133120.

(13) Xie, Z.; Bowman, J. *Chem. Phys. Lett.* **2006**, *429*, 355–359.

(14) Czako, G.; Braams, B.; Bowman, J. *J. Phys. Chem. A* **2008**, *112*, 7466–7472.

(15) Czako, G.; Shepler, B.; Braams, B.; Bowman, J. *J. Chem. Phys.* **2009**, *130*, 084301.

(16) Wang, Y.; Braams, B.; Bowman, J.; Carter, S.; Tew, D. *J. Chem. Phys.* **2008**, *128*, 224314.

(17) Chen, C.; Shepler, B.; Braams, B.; Bowman, J. *J. Chem. Phys.* **2007**, *127*, 104310.

(18) Mann, J.; Xie, Z.; Savee, J.; Bowman, J.; Continetti, R. *J. Chem. Phys.* **2009**, *130*, 041102.

(19) Chen, C.; Shepler, B.; Braams, B.; Bowman, J. *Phys. Chem. Chem. Phys.* **2009**, *11*, 4722–4727.

(20) Huang, X.; Braams, B.; Bowman, J. *J. Chem. Phys.* **2005**, *122*, 044308.

(21) Huang, X.; Braams, B.; Bowman, J. *J. Phys. Chem. A* **2006**, *110*, 445–451.

(22) Shank, A.; Wang, Y.; Kaledin, A.; Braams, B.; Bowman, J. *J. Chem. Phys.* **2009**, *130*, 144314.

(23) Wang, Y.; Shepler, B. C.; Braams, B. J.; Bowman, J. M. *J. Chem. Phys.* **2009**, *131*, 054511.

(24) Bosma, W.; Cannon, J.; Playoust, C. *J. Symbolic Comput.* **1997**, *24*, 235–265.

(25) *iOpenShell*; iOpenShell: University of Southern California, Los Angeles, CA; http://iopenshell.usc.edu/.

(26) Xie, Z. Ph.D. thesis, Emory University, Atlanta, GA, 2008.

(27) MSA Code; Argonne National Laboratory, Mathematics and Computer Science Division: Argonne, IL; http://www.mcs.anl.gov/msa.

(28) Huang, X.; Carter, S.; Bowman, J. *J. Chem. Phys.* **2003**, *118*, 5431–5441.

(29) Rajamaki, T.; Miani, A.; Halonen, L. *J. Chem. Phys.* **2003**, *118*, 10929–10938.

(30) Werner, H.-J.; Knowles, P. J. Lindh, R. Schütz, M. MOLPRO, version 2002.6; University College Cardiff Consultants Limited: Wales, U.K.; 2003.

(31) Wang, Y.; Carter, S.; Braams, B. J.; Bowman, J. M. *J. Chem. Phys.* **2008**, *128*, 071101.

(32) Zhang, X. B.; Zou, S. L.; Harding, L. B.; Bowman, J. M. *J. Phys. Chem. A* **2004**, *108*, 8980–8986.

CT9004917