Theor Chim Acta (1991) 78: 231-239

Theoretica
Chimica Acta

Utilization of 'pseudo-lattice symmetry' in cluster calculations

John D. Head and Ian P. Dillon
Department of Chemistry, University of Hawaii, Honolulu, HI 96822, USA

Received May 30, 1990; received in revised form and accepted August 6, 1990

Abstract. Cluster calculations which model chemisorption on a surface are often composed of substrate atoms arranged in a periodic manner. This pseudo-lattice symmetry of a cluster is used to reduce the number of 2-electron integrals computed in a SCF calculation by evaluating only unique integrals identified by lattice displacement vectors. The method, without using any explicit symmetry, is shown to be competitive with calculations which utilize point group symmetry. It is also demonstrated that the pseudo-lattice method markedly reduces the number of 2-electron integrals in multi-layer clusters which have little or no symmetry.

Key words: Cluster calculations – Chemisorption – Pseudo-lattice symmetry

1. Introduction

Cluster calculations are proving to be a popular method for theoretically modelling chemisorption on surfaces [1]. The clusters typically consist of a few atoms arranged as a fragment of the bulk material or surface, chosen so that an adsorbate has at least the same near-neighbor environment as that on the extended substrate. An attraction of the model is that the calculations can be performed using the well established methods of quantum chemistry allowing the prediction of adsorbate geometries, binding energies and vibrational frequencies. In contrast to the solid state approaches, such as 2-dimensional periodic slabs [2], clusters also readily allow the computation of properties for the low symmetry adsorption geometries. However, the number of atoms included in a cluster is often restricted owing to the computational requirements of the quantum chemical method. In an *ab initio* SCF calculation, the two major limitations are the disk storage requirements for the 2-electron integrals and the time needed to form and diagonalize the Fock matrix. Several methods for simplifying the 2-electron calculation already exist. Many *ab initio* computer packages automatically take advantage of the simplifications associated with point group symmetry [3]. As a consequence most cluster calculations are performed with the adsorbate and substrate having high symmetry. Integral approximations are another strategy making use of either thresholds to neglect integrals [4] or effective core potentials to replace the effects of the core electrons [5]. More recently direct SCF methods have been developed where the disk storage problem is eliminated by recomputing many of the 2-electron integrals in each SCF cycle [6]. For a large molecule or cluster the overhead of the extra 2-electron calculations per SCF cycle can be balanced against the time saving for the reduced input/output activity.

In this paper we present a new strategy solely for use in cluster calculations which reduces the integral disk storage requirements while maintaining reasonable computer time utilization in the SCF part of the calculation. Our new method takes advantage of the 'pseudo-lattice' translational symmetry of the cluster. In an infinite ordered solid with translational symmetry the nuclear coordinates are conveniently specified using the infinite set of lattice vectors

$$\mathbf{g}(n_1, n_2, n_3) = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_2\mathbf{a}_3 \tag{1}$$

where $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$ are the primitive translation vectors of the lattice. In our procedure we make use of a finite sized set of lattice vectors to describe the nuclear coordinates of the cluster. Obviously the cluster does not have translational symmetry in the group theoretical sense, but we utilize the 'pseudo-lattice' translational vectors connecting different atoms in the cluster to identify and restrict integral computation and storage to those 2-electron integrals that are unique. The unique 2-electron integrals can be compactly written as

$$(\chi_\mu^{\mathbf{0}}\chi_\nu^{\mathbf{v}_1}|\chi_\lambda^{\mathbf{v}_3}\chi_\sigma^{\mathbf{v}_2+\mathbf{v}_3}) \tag{2}$$

where $\chi_\mu^{\mathbf{g}}$ is the μ-th basis function in the $\mathbf{g}$-unit cell. 0 is an origin point and it is the lattice vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ which enable the unique integrals to be differentiated. That is, clusters which model chemisorption often have a regular arrangement of atoms, and because of matching displacement vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ between atoms, many of the 2-electron integral values between the cluster orbitals are repeated. In our 2-electron integral evaluation we only process the finite set of lattice vectors generated by the 'pseudo-lattice' translation vectors. To keep our algorithms simple we have not considered using local rotation or reflection operators, these would further reduce the number of integrals needing to be computed. Hence our integral storage requirements are not directly related to the point group symmetry of the cluster. Once all the pseudo-lattice atoms have been treated, further integrals between basis functions on additional atoms, such as adsorbates or substrate atoms distorted from their periodic positions, and the undistorted lattice can be computed to augment and complete a cluster's integral list.

The 'pseudo-lattice' method involves two steps: (1) searching for, computing and storing the unique 2-electron integrals; and (2) unpacking the 2-electron integrals and forming the Fock matrix. Both of these steps are described in the next section. Our procedure has several features in common with the methods used by Pisani et al. to treat the integrals needed in Hartree–Fock *ab initio* calculations on 2- and 3-dimensional crystalline solids [7]. However, our procedure is complicated by having incomplete 'stars' of lattice vectors when searching for the unique 2-electron integrals due to the finite cluster sizes. In Sect. 3 a comparison of the integral storage savings possible by using the 'pseudo-lattice' method with those obtained using point group symmetry for different sized H and Be clusters is given. Concluding remarks are given in the final section of the paper.

2. Method

The first step requires the definition of primitive translation vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$ which designate unit cell positions used to build the cluster. The unit cell may contain one or more nuclei with one or more basis functions. However, since the method processes only the displacement vectors between unit cells some of the cells may be left empty, such as in the case of a vacant site. Thus in the treatment of a fragment of a hexagonal close packed (*hcp*) lattice, rather than using a unit cell containing two atoms, it is easier to use primitive vectors from two interpenetrating simple hexagonal lattices. These simple hexagonal lattice unit cells now contain at most one atom, with several of the unit cells remaining empty. The coordinates of the atoms forming the cluster are determined from Eq. (1) where the appropriate integer values n_1, n_2 and n_3 are selected.

The unique positive displacement vectors between unit cells are found next. A displacement vector is the difference between two lattice sites

$$\begin{aligned}\boldsymbol{v}_{12} &= \boldsymbol{g}_2 - \boldsymbol{g}_1 \\ &= m_1\boldsymbol{a}_1 + m_2\boldsymbol{a}_2 + m_3\boldsymbol{a}_3 \end{aligned} \tag{3}$$

and is taken to be positive when $m_3 > 0$, or if $m_3 = 0$ then when $m_2 > 0$, or if $m_2 = m_3 = 0$ then when $m_1 \geqslant 0$. To every positive displacement vector, apart from the zero vector which has $m_1 = m_2 = m_3 = 0$, there is a corresponding negative vector pointing in the opposite direction. The definition of a positive displacement enables an ordering for the vectors, $\boldsymbol{v}_1 > \boldsymbol{v}_2$ when $\boldsymbol{v}_1 - \boldsymbol{v}_2$ is positive. A cluster's set of unique positive vectors are determined by looping pair wise over all the lattice points forming the cluster and by keeping only the distinct positive displacement vectors. For example, in a 2-dimensional cluster composed of a $n \times n$ square array of unit cells there will be $n^2 + (n-1)^2$ unique positive displacement vectors. The n^2 comes from counting all the possible displacements from a unit cell in the bottom left corner of the array to any other unit cell in the array and includes the zero displacement with itself. The $(n-1)^2$ term comes from displacements in the $-y$ direction starting from the unit cell in the bottom right corner and connecting to the other unit cells in the array except for those unit cells on the bottom row and right edge of the cluster. To facilitate finding the unique 2-electron integrals it is necessary to rearrange the set of unique positive displacement vectors into ascending order with the first element being the zero vector.

The unique 2-electron integrals $(\chi_\mu^0 \chi_\nu^{\boldsymbol{v}_1} | \chi_\lambda^{\boldsymbol{v}_3} \chi_\sigma^{\boldsymbol{v}_2 + \boldsymbol{v}_3})$ distinguished by the displacement vector set $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ can now be computed. The basic idea is to search through all the possible $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ combinations and examine whether the 2-electron integral actually occurs for the cluster; if it does then the integral is computed. The Fortran code with the required loop structure has the following form:

```
do $i_2 = 1$, $number\_displacement\_vectors$
$\mathbf{v}_2 = ordered\_vector(i_2)$
  do $i_1 = 1, i_2$
  $\mathbf{v}_1 = ordered\_vector(i_1)$
    do $i_3 = 1$, $number\_displacement\_vectors$
    $\mathbf{v}_3 = ordered\_vector(i_3)$

      $\mathbf{v}_4 = \mathbf{v}_3 + \mathbf{v}_2 - \mathbf{v}_1$
      if $\mathbf{v}_4 \neq \mathbf{0}$ then
        compute unique integral $I(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$
        else if $\mathbf{v}_1 = \mathbf{v}_2$ then
        compute unique integral $I(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$
      end if

      $\mathbf{v}'_4 = \mathbf{v}_3 + \mathbf{v}_1 - \mathbf{v}_2$
      if $\mathbf{v}'_4 < \mathbf{0}$ and $\mathbf{v}_3 \leqslant \mathbf{v}'_4$ then
        compute unique integral $I(\mathbf{v}_2, \mathbf{v}_1, \mathbf{v}_3)$
      end if

    end $i_3$ loop
  end $i_1$ loop
end $i_2$ loop                                                  (5)
```

The array *ordered_vector* contains all the unique positive displacement vectors stored in ascending order. The two outer loops are arranged so that $\mathbf{v}_2 \geqslant \mathbf{v}_1 \geqslant 0$, with $\mathbf{v}_2$ taking all the allowed positive unique displacement vector values. The logic of the loop structure is best explained using Fig. 1, where the 2-electron integrals are represented by a shape. Displacement vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ connect the basis functions associated with electron 1 and 2 respectively; $\mathbf{v}_3$, gives the displacement between the feet of $\mathbf{v}_1$ and $\mathbf{v}_2$. Figure 1a,b are what we call 'direct' shapes and the vectors $\mathbf{v}_4$ and $\mathbf{v}'_4$ have been included to connect the heads of $\mathbf{v}_1$ and $\mathbf{v}_2$. It is the repeat of the same shape on the cluster which allows a reduction in the integral storage. To each direct shape there is an 'inverted' shape, and these are illustrated in Fig. 1c,d. The inverted shape integral has the same value, to within a sign, as the direct shape integral. The loop structure in Eq. (5) initially picks out the direct shape shown in Fig. 1a with $\mathbf{v}_2 \geqslant \mathbf{v}_1$, and always results in $\mathbf{v}_4 \geqslant \mathbf{0}$. If $\mathbf{v}_1$ and $\mathbf{v}_2$ are interchanged, as in Fig. 1b, and $\mathbf{v}'_4 \geqslant \mathbf{0}$ then Fig. 1d shows that the corresponding inverted shape is equivalent to one of the direct shapes given by Fig. 1a. Sometimes the interchange of $\mathbf{v}_1$ and $\mathbf{v}_2$ generates the shape shown in Fig. 1e with $\mathbf{v}'_4 < \mathbf{0}$. Drawing the inverted shape with $\mathbf{v}'_4$ positive, Fig. 1f, still results in $\mathbf{v}_1$ and $\mathbf{v}_2$ being interchanged. Hence, in addition to the unique 2-electron integral represented by the direct shape in Fig. 1a, another unique 2-electron integral is generated when $\mathbf{v}'_4$ is negative. The purpose of the two if blocks in Eq. (5) should now be apparent. The first if block drives forming the direct shapes in Fig. 1a. If $\mathbf{v}_4 = \mathbf{0}$ the integral is not computed because it matches one of the inverted shapes of Fig. 1c. The exception to this is the integral with $\mathbf{v}_1 = \mathbf{v}_2$ and $\mathbf{v}_3 = \mathbf{v}_4 = \mathbf{0}$ which occurs once for each $\mathbf{v}_2$. The second if block checks for negative $\mathbf{v}'_4$ when $\mathbf{v}_1$ and $\mathbf{v}_2$ are interchanged. The additional test $\mathbf{v}_3 \leqslant \mathbf{v}'_4$ again avoids computing more than one unique integral because, as Fig. 1f illustrates, the inversion of the direct shape 1e results in a negative $\mathbf{v}_3$.

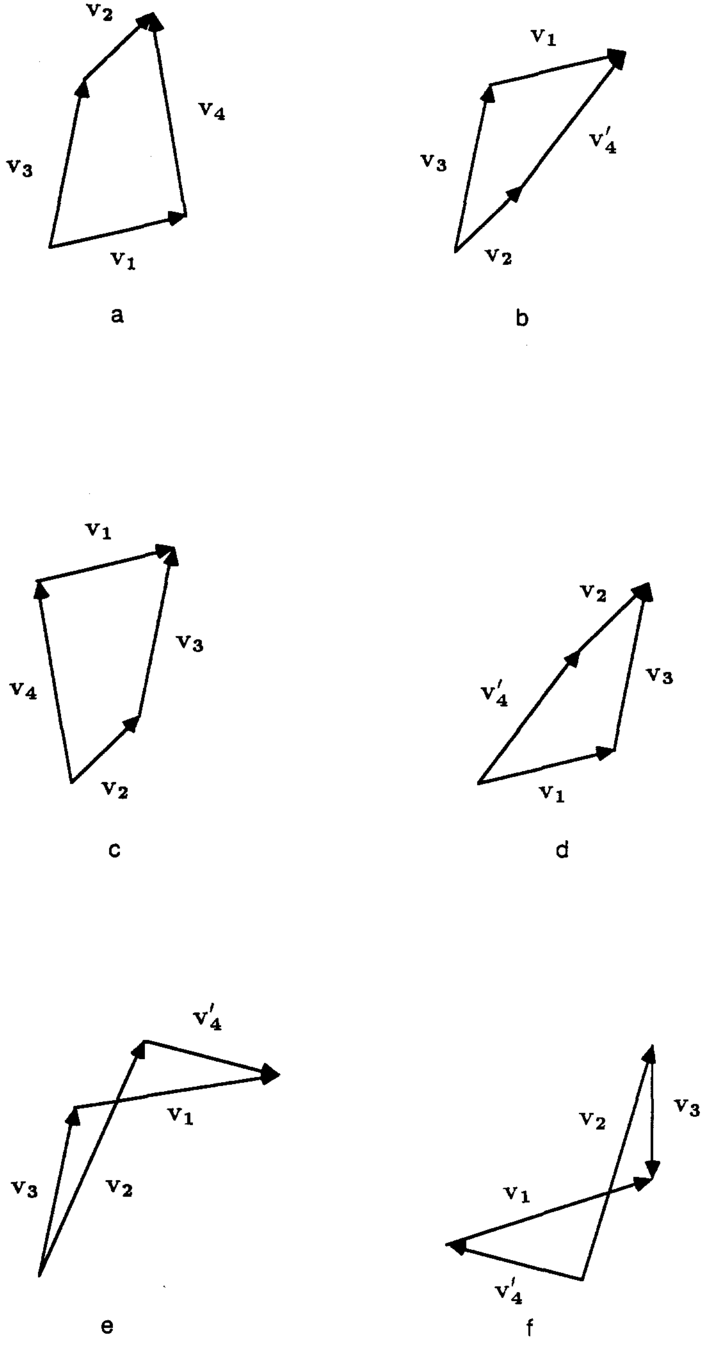

Fig. 1. Shapes of the 2-electron integrals derived from the unique lattice displacement vectors v_1, v_2 and v_3. **a** Direct shape; **b** Another direct shape but with v_1 and v_2 interchanged; **c** Inverted shape of **a**; **d** Inverted shape of **b** but which is equivalent to **a**; **e** Another type **b** direct shape but with $v_4' < 0$; **f** Inverted shape of **e** which does not produce a type **a** direct shape

The actual *compute unique integral* $I(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ refers to a subroutine which checks if the displacement vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ match atomic positions on the cluster. If the integral is allowed then the shell numbers of the basis functions located in the appropriate set of unit cells are passed to the integral evaluation subroutines. The integrals are stored using the packed basis function labels to identify the integral. This completes the description of computing and storing the unique 2-electron integrals.

Forming the Fock matrix involves expanding the unique 2-electron integral list to the full set of integrals. Like most *ab initio* packages the basis function labels are unpacked first, but now these labels are also used to obtain the positive displacement vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ associated with each integral. The program then finds all the different positions where the shape $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ can fit on the cluster and generates appropriate basis function labels to be used in adding the 2-electron integral to the Fock matrix elements. It is this step of locating the integral shapes on the cluster which adds a time overhead to the SCF calculation. However, because of the shell structure of the integrals, several integrals with the same displacement vectors can be processed simultaneously. Generally, for each unique integral shape there are two types of contributions to the Fock matrix. There is always the direct shape, but in many cases a contribution from the inverted shape is also needed. The inverted shape, which arises from the 'pseudo-inversion symmetry' of the cluster, is not added to the Fock matrix when it is the same as the direct shape. This occurs when either $\mathbf{v}_1 = \mathbf{v}_2$ or $\mathbf{v}_3 = -\mathbf{v}_4$. The sign of the integral value of the inverted shape is changed for each p or f basis function in the integral.

The method has been programmed into the HONDO package [8]. Changes were made in the integral driver routine and the Fock formation routine.

3. Results

We now illustrate the effectiveness of pseudo-lattice symmetry in reducing the number of 2-electron integrals required to perform a SCF calculation on a cluster. We have performed calculations on different sized H and Be clusters and for reference make comparisons with the number of 2-electron integrals needed when using various point group symmetries. For each cluster, identical energies are obtained at each SCF iteration, confirming that the pseudo-lattice symmetry approach does neither approximate nor neglect any of the 2-electron integrals.

For calculations with only a single basis function at a lattice point, Table 1 shows that the pseudo-lattice method gives an appreciable reduction in the integral storage requirements. Table 1 lists results using pseudo-lattice symmetry and point group symmetry for different H atom clusters which model the (100) surface of a simple cubic lattice. Each H has a single contracted STO-2G s basis function centered on it and the nearest H–H distance is 0.6 Å. In the single layer clusters, the H_4, H_{16} and H_{36} pseudo-lattice calculations are competitive with the C_s, C_{2v} and C_{4v} symmetries, respectively. Similarly, for the multiple 4 × 4 layer H clusters, the number of different 2-electron integrals needing to be computed in pseudo-lattice procedure does not increase as rapidly with cluster size as it does for the calculations using point group symmetry. Obviously the larger a cluster, the more times a 2-electron integral value will be repeated. For example, in the 2 layer H_{32} pseudo-lattice cluster calculation the set of 2219 unique 2-electron integrals, evaluated for the single layer H_{16} cluster, would also be found only

Table 1. Comparison of the number of 2-electron integrals computed using pseudo translational symmetry only with calculations using point group symmetry for single and multiple layer clusters composed of H atoms arranged as fragments of a simple cubic lattice

Single layers	2 × 2	4 × 4	6 × 6
Pseudo lattice	28	2219	29,914
C_1	55	9316	222,111
C_s	31	4708	111,303
C_{2v}	19	2404	55,899
C_{4v}	13	1272	28,314

4 × 4 Layers	4 × 4 × 1	4 × 4 × 2	4 × 4 × 3
Pseudo lattice	2219	30,542	129,269
C_1	9316	139,656	692,076
C_s	4708	70,024	346,476
C_{2v}	2404	35,208	173,676
D_{2h}	2404	17,800	89,164

Table 2. Comparison of the number of 2-electron integrals computed using pseudo translational symmetry only with calculations using point group symmetry for various *hcp* Be clusters

Cluster	2 × 2	3 × 3	4 × 4	3 × 3 × 2
Pseudo lattice	6,033	93,349	503,123	2,166,840
C_1	11,656	272,691	2,269,651	6,967,168
C_s	6,294	139,702	1,165,525	3,556,408
C_{2v}	3,445	71,965	597,703	—

once. The remaining 28,323 H_{32} integrals are for integrals whose component orbitals always span the 2 layers of the cluster. When point group symmetry is used, apart from the D_{2h} calculation, the set of single layer H_{16} 2-electron integrals are duplicated twice in the double layer H_{32} cluster, once on the top layer and once on the bottom layer. The third mirror plane of the D_{2h} point group parallel to the H_{16} layer eliminates this repetition of these 2404 single layer integrals in the double layer H_{32} calculation; however, in the triple layer H_{48} cluster the 2404 single layer integrals are repeated twice, once by the outer H_{16} clusters and again by the middle H_{16} layer.

A number of cluster [9–14] and extended 2-dimensional slab [15–17] calculations which model the Be(0001) surface of the *hcp* lattice have been reported previously. Differences in the preferred site and binding energy for H adsorption on the Be(0001) surfaces predicted by cluster and slab calculations do occur. However, in a recent cluster study involving multiple H adsorption we obtained results similar to the slab calculations when the cluster's H coverage was simulated as a monolayer [15]. Work is in progress to extend this study of multiple H adsorption on Be(0001) by using the present pseudo-lattice method.

In Table 2 we present a comparison of the number of 2-electron integrals computed for different sized clusters consisting of only Be atoms. The calculations are performed with the minimal STO-3G basis set developed by Bauschlicher et al. [9] which consists of 2*s* and 1*p* orbitals resulting in 5 basis functions per Be atom. 2-Electron integrals with values less than 10^{-9} were neglected. The pseudo-lattice calculation for the multiple layer clusters can be performed in several ways. The smallest unit cell for a *hcp* lattice contains 2 atoms and the pseudo-lattice method could still work with a cluster built up from these unit cells. However, in the double layer Be_{18} cluster listed in Table 2, the top layer (A) lattice points are generated by

$$\boldsymbol{g}_A = 3n_1\boldsymbol{a}_1 + 3n_2\boldsymbol{a}_2 + n_3\boldsymbol{a}_3 \tag{6a}$$

and the bottom layer (B) by

$$\boldsymbol{g}_B = (3n_1 + 1)\boldsymbol{a}_1 + (3n_2 + 1)\boldsymbol{a}_2 + (3n_3 + 1) + n_3\boldsymbol{a}_3 \tag{6b}$$

where the primitive vectors are

$$\begin{aligned} \boldsymbol{a}_1 &= \frac{1}{3}\left(\frac{\sqrt{3}}{2}a, \frac{a}{2}, 0\right) \\ \boldsymbol{a}_2 &= \frac{1}{3}\left(\frac{\sqrt{3}}{2}a, -\frac{a}{2}, 0\right) \\ \boldsymbol{a}_3 &= (0, 0, c) \end{aligned} \tag{7}$$

with the in-layer Be–Be separation, $a = 2.2866$ Å, and the A–B interlayer distance, $c = 1.792$ Å, being taken from the bulk value [19]. The single layer Be cluster coordinates are produced by Eq. (6a). The coordinates of a Be cluster with many layers can be generated by the appropriate repetition of the A and B layers given by Eqs. (6a) and (6b). An important feature of the pseudo-lattice method is that it only processes the displacement vectors between lattice points. This means again, like the H clusters, that the 93,349 2-electron integrals evaluated for the single layer Be_9 cluster are only computed once for the double layer Be_{18} cluster.

Table 2 shows that the pseudo-lattice method compares favorably with the higher point group symmetry calculations, with the pseudo-lattice method being more effective at reducing the integral storage requirements as the Be clusters become larger. The double layer Be_{18} calculation is particularly noteworthy because the *ABAB* stacking of the a *hcp* lattice eliminates the possibility of a multiple layer cluster having C_{2v} point group symmetry. A similar point group restriction will occur in multiple layer cluster models of the (111) surface of a face centered cubic (*fcc*) lattice. Even further reduction in the integral storage could be achieved by extending the present algorithms to include local point group symmetry. However, the complexity of such an algorithm and the overhead associated with unpacking these point group symmetrized integrals would probably outweigh the advantage of using the present method.

4. Conclusion

Clusters calculations using the well developed techniques of quantum chemistry are a convenient method for modelling low adsorption sites on a surface. Often, however, the low symmetry of the cluster makes the calculations computationally demanding. In this paper we present a new method for reducing the number of 2-electron integrals in the calculation. This 'pseudo-lattice' method takes advantage of the ordered arrangement of atoms often used to form the cluster. The reduction in the number of 2-electron integrals is achieved by processing the displacements between the lattice points on the cluster. Providing the cluster has an underlying set of primitive translation vectors connecting the different atoms composing the cluster a reduction in the number of 2-electron needing to be computed can be achieved regardless of the symmetry of the cluster.

Acknowledgements. We are grateful for partial financial support from the Materials and Applied Sciences Program at the University of Hawaii.

References

1. Sauer J (1989) Chem Rev 89:199; Zhidomirov GM, Kazansky VB (1986) Advan in Catalysis 34:131
2. Cohen ML, Louie SG (1984) Ann Rev Phys Chem 35:537; Pisani C, Dovesi R (1987) Theoret Chim Acta 72:277; Boettger JC, Trickey SB (1986) J Phys F: Met Phys 16:693; Hoffmann R (1988) Rev Mod Phys 60:601
3. Dacre PD (1970) Chem Phys Lett 7:47; Elder M (1973) Intern J Quantum Chem 7:75; Dupuis M, King HF (1977) Intern J Quantum Chem 11:613; Takada T, Dupuis M (1981) J Chem Phys 75:332
4. Head JD, Silva SJ (1989) Intern J Quantum Chem Symp 23:231
5. Ermler WC, Ross RB, Christiansen PA (1988) Adv Quant Chem 19:139
6. Almlof J, Faegri K, Korsell K (1982) J Comput Chem 3:385; Almlof J, Taylor PR (1984) In: Dykstra CE (ed) Advanced theories and computational approaches to the electronic structure of molecules. Reidel, Dordrecht, p 107
7. Pisani C, Dovesi R, Roetti C (1988) Hartree–Fock *ab initio* treatment of crystalline systems. Springer, Berlin Heidelberg New York
8. HONDO, Dupuis M, Watts JD, Villar HO, Hurst GJB (1987) Version 7.0, IBM, Scientific and Engineering Computations, Kingston, NY
9. Bauschlicher CW, Bagus PS, Schaefer, HF (1978) IBM J Res Develop 22:213; Bagus PS, Schaeffer HF, Bauschlicher CW (1983) J Chem Phys 78:1390
10. Cox BN, Bauschlicher CW (1981) Surf Sci 102:295
11. Bauschlicher CW, Bagus PS (1982) Chem Phys Lett 90:335
12. Pacchioni G, Pewestorf W, Koutecky J (1984) Chem Phys 83:261
13. Rubio J, Illas F, Ricart JM (1986) J Chem Phys 84:3311
14. Ross RB, Ermler WC, Pitzer RM, Kern CW, (1987) Chem Phys Lett 134:115; Ermler WC, Ross RB, Kern CW, Pitzer RM, Winter NW (1988) J Phys Chem 92:3042; Marino MM, Ermler WC, Tompa GS, Seidl M (1989) Surf Sci 208:189
15. Dillon IP (1990) MS thesis, University of Hawaii; Dillon IP, Head JD, submitted to Surf Sci
16. Wu JZ, Trickey SB, Boettger JC (1990) Phys Rev B 42:1633
17. Yu R, Lam PK (1989) Phys Rev B39:5035
18. Angonoa G, Koutecky J, Ermoshkin AN, Pisani C (1984) Surf Sci 138:51
19. Donohue J (1974) The structure of the elements. Wiley, New York