Theor Chim Acta (1993) 85: 423–440

Theoretica
Chimica Acta

© Springer-Verlag 1993

The reduced multiplication scheme of the Rys–Gauss quadrature for 1st order integral derivatives

Roland Lindh

Department of Theoretical Chemistry, Chemical Center University of Lund, PoB 124, S-22100 Lund, Sweden

Received July 15, 1992/Accepted October 5, 1992

Summary. An implementation of the reduced multiplication scheme of the Rys–Gauss quadrature to compute the gradients of electron repulsion integrals is discussed. The study demonstrates that the Rys–Gauss quadrature is very suitable for efficient utilization of simplifications as offered by the direct computation of symmetry adapted gradients and the use of the translational invariance of the integrals. The introduction of the so-called intermediate products is also demonstrated to further reduce the floating point operation count. Two prescreening techniques based on the 2nd order density matrix in the basis of the uncontracted Gaussian functions is proposed and investigated in the paper. This investigation gives on hand that it is not necessary to employ the Cauchy–Schwarz inequality to achieve efficient prescreening. All the features mentioned above were demonstrated by their implementation into the gradient program ALASKA. The paper offers a theoretical and practical assessment of the modified Rys–Gauss quadrature in comparison with other methods and implementations and a detailed analysis of the behavior of the method as suggested above as a function of changes with respect to symmetry, basis set quality, molecular size, and prescreening threshold.

Key words: Integral derivatives – Rys–Gauss quadrature – Reduced multiplication scheme – Gradients – Symmetry adapted gradients – Prescreening

1 Introduction

The recent year's outburst of publications on the subject of the computation of two-electron integrals has not only presented quite a number of new methods or algorithms in this field but has also made it more or less impossible to compose an introduction to the subject which will not give the initiated reader a feeling of *deja vu* (this statement holds also for the section of definitions). However, a small summary of previous work and a motivation for the presented work is mandatory.

The fuse to the new interest in novel electron repulsion integral (ERI) algorithms was the publication in 1986 of the Obara–Saika [1] (OS) recurrence relation. This expression facilitates a speedy evaluation of the primitive ERI's when implemented on a vector computer. This work was followed by Head-Gordon and Pople [2] (HGP) suggesting the new recurrence relation to be combined with the use of the transfer equation [3] (denoted the horizontal recurrence relation (HRR) in their work). With the HGP method one initially generates a set of intermediate ERI's in the primitive base with the OS recurrence relation (the relation denoted the vertical recurrence relation (VRR) by HGP is a special case of the OS recurrence relation). The second phase of the HGP is then performed in the contracted basis where the application of the transfer equation (HRR) will produce both the ERI's and the first order derivatives with respect to the centers of the basis functions. An addition of a third recurrence relation was introduced independently by Hamilton and Schaefer [4] (HS), and by Lindh et al. [5] (LRL), to improve the performance even further. Quite recently Ryu et al. [6] suggested further improvements of the transfer equation for integrals of high total angular momentum by the observation that some of the steps in the algorithm are redundant and can be eliminated. In parallel Gill et al. [7] suggested yet a new method which is a synthesis of the McMurchie–Davidson [8] scheme and the transfer equation as applied by HGP. In this method, which is suggested to be superior to the modified HGP (HS or LRL) method, the authors introduced the concepts of early and late contraction (transformation from the primitive basis to the AO basis), respectively. This has recently been extensively used in the BRACKET method of Gill et al. [9] and in the PRISM method by Gill and Pople [10]. These new methods are flexible such that optimal performance is achieved regardless of the degree of contraction. The methods mentioned so far have the following ingredients in common; *i*) they produce contracted ERI's and/or derivatives of ERI's, and *ii*) the efficiency is associated with the recurrence relation using the redundancy when all the elements in a shell quadruplet are produced.

As a contrast to these methods we have the Rys–Gauss quadrature [3] which is based on a simple recurrence relation for the so-called two dimensional (2D) integrals (also called subsidiary functions) and an assembling of the primitive integrals from these 2D-integrals. This method has over the years earned a good reputation due to a number of successful implementations such as the HONDO [11], CADPAC [12], and GRADSCF [13] quantum chemistry program packages and now recently in the integral program SEWARD [5]. The Rys–Gauss quadrature is known for its simple structure. However, the method does not offer any flexibility such as early and late contraction. Furthermore, the method was known to perform poorly for ERI batches of low total angular momentum. LRL, however, showed in their paper in 1991 that with the introduction of the reduced multiplication scheme the method is very competitive and is outstanding in the case of low degree of contraction [5].

In the computation of molecular gradients the force is computed as the trace of the *effective* (also called the *variational*) density/Fock matrices and the integral derivatives [14]. It is important to note that the basis in which the trace is performed can be any in the range from the primitive basis of Gaussian or Hermite functions to the symmetry adapted molecular orbital basis. Combined with the information that the contraction step in ordinary ERI evaluation takes 15–25 percent of the total CPU time one can argue if the integral derivatives should be traced in the contracted or the primitive base, i.e. if the gradient should be contracted to the AO basis or if the 2nd order density matrix should be backtransformed to the primitive basis (decontraction). Following the HGP method one has to contract the intermediate integrals in four different ways (the differentiation puts the Gaussian exponent in front of some integrals) whereas the calculation of the primitive integral derivatives would require one decontraction of the 2nd order density matrix (in the case of a correlated wavefunction).

The list of intermediate integrals is somewhat smaller than the list of the elements in the 2nd order density matrix. However, the difference is small (especially for batches of low total angular momentum) and will not compensate for the fact that the contraction step will be about four times as expensive as the corresponding decontraction. The situation in the comparison to the BRACKET and PRISM method is similar although the set of intermediate entities is somewhat smaller as compared to the HGP method. In addition to this reason for the decontraction of the second order density matrix we note that the atomic natural orbital (ANO) basis sets [15] as initially introduced by Almlöf and Taylor [16] are of Raffanetti type [17], i.e. the contracted basis functions have the same set of Gaussian exponents in common. Hence, an efficient prescreening can only be achieved if executed in the uncontracted basis. This matter is actually quite crucial since efficient screening will reduce the CPU time with a factor of about two or more [18]. The conclusion from these facts is that methods which generate the primitive integral derivatives fast will have an advantage as compared to other methods. The Rys–Gauss quadrature is such a method.

In the computation of gradients there are a few relations and constraints which can be used to reduce the computational effort. These are symmetry, and translational invariance of the integrals. In methods in which the efficiency of the gradient evaluation depends on the use of complete shells these simplifications will be difficult to fully exploit. This is not the case with the Rys–Gauss quadrature. In addition to this the Rys–Gauss quadrature does not require as much memory as other methods. This is a very critical point especially for the gradient calculations of ANO type basis sets.

We will in the coming section show how the Rys–Gauss quadrature in combination with efficient reuse of intermediate products, full utilization of symmetry constraints, as well as the translational invariance of the primitive integrals, the decontraction of the 2nd order density matrix, and efficient prescreening offers an attractive way of computing molecular gradients. The presented variation of the Rys–Gauss quadrature for the evaluation of the integral derivatives will mainly be in line with the method as suggested by Amos [19] and Dupuis [20] as compared to the approaches suggested by Saxe et al. [21] and Schlegel and coworkers [22]. The present method has been implemented in the integral gradient program ALASKA which has been interfaced with the quantum chemistry package MOLCAS-3 [23].

2 Theory

We will in this section assume that the reader is familiar with the normal notation used for the Rys–Gauss quadrature, ERI's and Gaussian functions. Otherwise we recommend the reader to study the theory section in LRL [5].

2.1 Integral gradient evaluation

In the Rys–Gauss quadrature the ERI's are expressed as:

$$(\boldsymbol{ab}\,|\,\boldsymbol{cd}) = 2\left(\frac{\varrho}{\pi}\right)^{\frac{1}{2}} \kappa_{AB}\kappa_{CD}\left(\frac{\pi}{\zeta}\right)^{\frac{3}{2}}\left(\frac{\pi}{\eta}\right)^{\frac{3}{2}} \times \sum_{\alpha=1}^{n_{Rys}} I'_x(a_x, b_x, c_x, d_x : t_\alpha) I'_y(a_y, b_y, c_y, d_y : t_\alpha) I'^{*}_z(a_z, b_z, c_z, d_z : t_\alpha). \quad (1)$$

where $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$ are the angular momentum vectors of the Gaussian basis functions (the exponents α, β, γ, and δ or the centers $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are not explicitly referenced), ϱ, ζ, and η are functions of the Gaussians exponents, κ_{AB} and κ_{CD} are factors due to the products of the two Gaussian functions of each charge distribution, and I'_λ are the 2D-integrals (the superscript asterisk on the z-component indicate that these 2D-integrals have been multiplied with the weights associated with the Rys–Gauss quadrature). The corresponding gradient of the ERI is computed by substituting the appropriate 2D-integral by its first order derivative. The first order derivative of a 2D-integral with respect to a nuclear coordinate, $\boldsymbol{E}$, is computed as [19, 20]:

$$\begin{aligned}\frac{\partial I'_\gamma(a_\lambda, b_\lambda, c_\lambda, d_\lambda)}{\partial E_\lambda} &= \delta_{AE}[a_\lambda I'_\lambda(a_\lambda - 1, b_\lambda, c_\lambda, d_\lambda) - 2\alpha I'_\lambda(a_\lambda + 1, b_\lambda, c_\lambda, d_\lambda)] \\ &+ \delta_{BE}[b_\lambda I'_\lambda(a_\lambda, b_\lambda - 1, c_\lambda, d_\lambda) - 2\beta I'_\lambda(a_\lambda, \beta_\lambda + 1, c_\lambda, d_\lambda)] \\ &+ \delta_{CE}[c_\lambda I'_\lambda(a_\lambda, b_\lambda, c_\lambda - 1, d_\lambda) - 2\gamma I'_\lambda(a_\lambda, b_\lambda, c_\lambda + 1, d_\lambda)] \\ &+ \delta_{DE}[d_\lambda I'_\lambda(a_\lambda, b_\lambda, c_\lambda, d_\lambda - 1) - 2\delta I'_\lambda(a_\lambda, b_\lambda, c_\lambda, c_\lambda, d_\lambda + 1)], \\ &\lambda = x, y, z. \end{aligned} \tag{2}$$

At this point we notice that in contrast to the recursive methods (HGP, PRISM and BRACKET) in which derivatives of ERI's of three or fewer centers can not be computed directly the Rys–Gauss quadrature will compute these gradients in one step as:

$$\begin{aligned}\frac{\partial(\boldsymbol{ab}\,|\,\boldsymbol{cd})}{\partial E_x} &= 2\left(\frac{\varrho}{\pi}\right)^{\frac{1}{2}}\kappa_{AB}\kappa_{CD}\left(\frac{\pi}{\zeta}\right)^{\frac{3}{2}}\left(\frac{\pi}{\eta}\right)^{\frac{3}{2}}\sum_{\alpha=1}^{n_{Rys}}\frac{\partial I'_x(a_x, b_x, c_x, d_x : t_\alpha)}{\partial E_x} \\ &\times I'_y(a_y, b_y, c_y, d_y : t_\alpha) I'^*_z(a_z, b_z, c_z, d_z : t_\alpha), \end{aligned} \tag{3}$$

whereas the recursive methods will employ:

$$\begin{aligned}\frac{\partial(\boldsymbol{ab}\,|\,\boldsymbol{cd})}{\partial E_x} &= \delta_{AE}\frac{\partial(\boldsymbol{ab}\,|\,\boldsymbol{cd})}{\partial A_x} + \delta_{BE}\frac{\partial(\boldsymbol{ab}\,|\,\boldsymbol{cd})}{\partial B_x} \\ &+ \delta_{CE}\frac{\partial(\boldsymbol{ab}\,|\,\boldsymbol{cd})}{\partial C_x} + \delta_{DE}\frac{\partial(\boldsymbol{ab}\,|\,\boldsymbol{cd})}{\partial D_x} \end{aligned} \tag{4}$$

The efficiency of the Rys–Gauss quadrature is associated with the much lower expense of computing the derivative of the 2D-integrals as compared to the ERI's. Furthermore, although some reduction in the FLOP count can be achieved when all gradients of a shell quadruplet are computed as will be discussed below we would like to stress that the Rys–Gauss quadrature is not as dependent as the recursive method on this situation in order to achieve a low CPU expense per gradient. This allows for a simple and efficient utilization of symmetry constraints. We note in particular that we only need the gradients with respect to those centers which will form symmetric displacements. This observation holds equally true if the calculation is performed in a double coset formalism [24] or in a petite list format [25]. The symmetry restriction will be of use since it will avoid the computation of quite a number of integral gradients (e.g. for a planar molecule we do not need the displacements out of the plane). For each redundant gradient which will not be computed we will reduce the computational effort of assembling the gradients of a shell quadruplet with about 1/9 as compared to computing all 9 gradients.

2.2 The reduced multiplication scheme

At this point we will introduce the intermediate products and the reduced multiplication scheme which will reduce the FLOP count further as compared to the traditional implementation of the Rys–Gauss quadrature. In the case of a four center ERI we will compute the gradients of three centers. The translational invariance of the integral gives the fourth gradient. The most efficient utilization of this restriction is not to compute the missing gradient directly but rather its contribution after contraction with the density matrix (all 12 integral gradients are contracted with the same 2nd order density matrix). Hence, for the four center cases we will for each cartesian component compute three different gradients. These gradients have the same undifferentiated part in common (the product of two 2D-integrals), see Eq. (3). This will form our intermediate product. Thus the FLOP count for assembling of the three gradients of a cartesian component of a shell quadruplet will reduce from $(9 * n_{Rys} - 3)$ to $(7 * n_{Rys} - 3)$, i.e. the savings are 50, 36, and 33% for the 1st order derivatives of the integral batches up to and equal to $(ss|ss)$, $(pp|ss)$, and $(pp|pp)$, respectively. In addition to this some caution will have to be applied to the formation of the intermediate product since the 2D-integrals with a total angular index of zero have the value of unity. In these cases the intermediate product is directly defined by the 2D-integral with a total angular momentum different from zero and in some cases the integral derivative is simply the sum of the differentiated 2D-integral when both the undifferentiated 2D-integrals have a total angular momentum of zero. Some restriction of this rule has to be applied since the weight is carried by the z-component 2D-integral, i.e. the z-component will always have to be included in the assembling of the integral derivative.

2.3 Prescreening

The general molecular gradient formula is expressed in the molecular orbital (MO) basis as [14]:

$$E^{(1)} = \sum_{pq} D_{pq} h_{pq}^{(1)} + \tfrac{1}{2} \sum_{pqrs} P_{pqrs} (pq|rs)^{(1)} - \sum_{pq} F_{pq} (p|q)^{(1)} \qquad (5)$$

where D_{pq}, P_{pqrs}, and F_{pq} are the wavefunction dependent *effective* first order density matrix, second order density matrix, and Fock matrix, respectively. $H_{pq}^{(1)}$, $(pq|rs)^{(1)}$, and $(p|q)^{(1)}$ are the first order derivatives of the one-electron hamiltonian, the two-electron integral and the overlap integral, respectively, and all are independent of the wavefunction. The time consuming part of the evaluation of the molecular gradient is of course due to the contributions from the two-electron integrals. This contribution to the total CPU time, however, can be significantly reduced by proper prescreening. This has successfully been implemented for the SCF type of wavefunction [18] by Horn et al. In their implementation the Cauchy–Schwarz inequality was used to estimate the integrals and their associated gradients. These estimates combined with the first order density matrix formed the base for the prescreening. In a correlated wavefunction, however, we have to combine the estimated integrals and gradients with the 2nd order density matrix in order to achieve the most sensitive prescreening. This requires some more delicate handling as compared to implementations of gradients of SCF wavefunctions in order for the prescreening itself not just to form a new bottleneck in the gradient calculation. We will below sketch two such procedures based on the second order density matrix. It would, of course, be possible in the case of a correlated wavefunction to approximate the 2nd order density matrix in the prescreening as a product of first order densities and to maintain a prescreening procedure similar to that one in a SCF wavefunction. However, such a scheme would assign the value of zero to all those elements of the 2nd order density matrix which are not total symmetric on the two first indices. It is most likely that such an approximation will at least in some cases be impossible.

The decontraction of the 2nd order density matrix as discussed in the introduction is a critical part of the efficiency of the implementation. This also leaves us with the possibility to multiply the prefactor (the factor before the summation sign) with the integral derivative as suggested above (see Eq. (3)) or to modify the decontracted 2nd order density matrix with the same factor. For the same reason that we chose to decontract the 2nd order density matrix we will multiply it with the prefactor. This is also a step in the direction towards an efficient prescreening. For reasons of clarity let us reformulate the summation over the two-electron integral contribution to the gradient. This is in order to introduce a more appropriate notation since the actual calculation for reasons of efficiency is carried out in the primitive basis and the integral batches are computed as complete shell quadruplets. Hence, we will introduce a reference to the Gaussian exponent and to the component of the angular part of the Gaussian basis function. Thus the two-electron gradient contribution is formulated as:

$$\cdots + \tfrac{1}{2} \sum_{\zeta\eta} \sum_{abcd} P_{\zeta\eta,abcd}(ab\,|\,cd)^{(1)}_{\zeta\eta} \cdots \tag{6}$$

where a, b, c, and d are the indices of the components of the angular function and ζ and η are the Gaussian exponents of the charge distributions due to the Gaussian basis functions of center A and B, and C and D, respectively. In the formalism of the Rys–Gauss quadrature we would express the same formula as:

$$\begin{aligned}\cdots + \tfrac{1}{2} \sum_{\zeta\eta} \sum_{abcd} & P'_{\zeta\eta,abcd} \\ & \times \frac{\partial}{\partial E_\lambda} \sum_{\alpha=1}^{n_{Rys}} I'_x(a_x, b_x, c_x, d_x : t_\alpha) I'_y(a_y, b_y, c_y, d_y : t_\alpha) I'^{*}_z(a_z, b_z, c_z, d_z : t_\alpha) \cdots\end{aligned} \tag{7}$$

where the modified 2nd order density matrix is defined as:

$$P'_{\zeta\eta,abcd} = 2\left(\frac{\varrho}{\pi}\right)^{\frac{1}{2}} \kappa_{AB}\kappa_{CD} \left(\frac{\pi}{\zeta}\right)^{\frac{3}{2}} \left(\frac{\pi}{\eta}\right)^{\frac{3}{2}} \times P_{\zeta\eta,abcd} \tag{8}$$

Let us also note that in order for the prescreening on the second order density matrix to be efficient there has to be a compromise between a straightforward implementation and a maybe less strict approach. The former will be hard to adapt to a vectorizable implementation whereas the latter can be formulated with the aim of not only facilitating some sort of prescreening but also to constitute an algorithm tailored for a computer implementation.

In our first procedure we will base the prescreening directly on the value of the modified 2nd order density matrix. In this approach we will assume that the sum over weighted quadrature points will add up to unity or less. To justify this statement to some degree consider the sum over weighted quadrature points for the shell quadruplet $(ss|ss)$. In this particular case the product of the subsidiary functions have the value of unity and the weight is identical to the zeroth order incomplete gamma function, $F_0(T) = \int_0^1 dx\, e^{-Tx^2}$. We observe immediately that $F_0(T) \leqslant 1$. Hence, our prescreening will form an upper bound in this case. For shell quadruplets of a total angular momentum higher than this our estimate will no longer form an upper bound to the estimated contribution to the energy/gradient. To facilitate the prescreening we will need to form:

$$W(\eta)_\zeta = \left(\sum_{\eta} \sum_{a,b,c,d} (P'_{\zeta\eta,abcd})^2 \right)^{\frac{1}{2}}. \tag{9}$$

The value of this norm is compared to the threshold and based on this comparison we will either compute all gradients associated with the index ζ or not. This procedure is repeated a second time with respect to the index η in order to complete the prescreening. This procedure will in the forthcoming sections be denoted the "2nd order density matrix only method" (2DO). Observe the similarity of this method and the traditional prescreening method which is based on the radical overlap (see Horn et al. [18] Eq. (29)). However, the present implementation does also include the factor due to the transformation of the variable of integration of the incomplete gamma function. Hence, the estimate of the integral/gradient will include some coupling between the two charge distributions.

The second method will base the gradient estimate on the Cauchy–Schwarz inequality. This means that the gradients will be approximated with an expression which is an upper bound to the exact value and is expressed as:

$$|(a^ib|cd)_{\zeta,\eta}| \leqslant |(a^ib|a^ib)_{\zeta,\zeta}|^{\frac{1}{2}} |(cd|cd)_{\eta,\eta}|^{\frac{1}{2}}, \tag{10}$$

where the superscript i indicates a differentiation. For an efficient implementation we will form the following entities:

$$(ab)_\zeta = |(ab|ab)_{\zeta,\zeta}|^{\frac{1}{2}}, \tag{11}$$

and

$$(ab)_\zeta^* = |(a^ib|a^ib)_{\zeta,\zeta}|^{\frac{1}{2}} + |(ab^i|ab^i)_{\zeta,\zeta}|^{\frac{1}{2}}. \tag{12}$$

For the last entity the asterisk indicates that we will sum 1st order differentiations in all possible directions with respect to any of the centers for displacements belonging to the total symmetric representation. These 2nd order derivative integrals are in the current implementation evaluated by 2nd order numerical differentiation. The prescreening threshold will now be compared with the value of:

$$Q(\eta)_\zeta = \left(\sum_{\eta} \sum_{abcd} (P'_{\zeta\eta,abcd} \times \{(ab)_\zeta^*(cd)_\eta + (ab)_\zeta(cd)_\eta^*\})^2 \right)^{\frac{1}{2}}, \tag{13}$$

and a similar expression with respect to the Gaussian index η. This procedure will in the forthcoming sections be denoted the "2nd order density matrix and integral estimate method" (2DI). Observe that Q reduces to W when the integral gradient estimate is replaced with unity.

Let us note a technical detail in comparison with the implementation of Horn et al. which will be of importance for the analysis of the performance of the implementation. As the prescreening is suggested here we do not look for the largest element but we rather form some sort of a norm. This will mean that in case we have a large number of contributions which one by one is below the threshold we will still include these in the summation. Hence, this approach could be expected to be somewhat more stable. The success of the presented methods is very much dependent on the implementation and below follows a short description of the loop structure of the 2-electron integral gradient driver as implemented in ALASKA. The symmetry handling in the program is the double coset representative (DCR) technique due to Davidson [24, 26].

```
LOOP OVER SYMMETRY ADAPTED SHELL A
  LOOP OVER SYMMETRY ADAPTED SHELL B
    LOOP OVER SYMMETRY ADAPTED SHELL C
      LOOP OVER SYMMETRY ADAPTED SHELL D
        GATHER $P^{\alpha\beta\gamma\delta}_{pq,abcd}$ IN SO BASE
        LOOP OVER DCR OPERATOR $\hat{R}$
             LOOP OVER DCR OPERATOR $\hat{S}$
               LOOP OVER DCR OPERATOR $\hat{T}$
                  DESYMMETRIZE $P^{\alpha\beta\gamma\delta}_{pq,abcd}$ TO AO BASE
                  PROJECT $P_{pq,abcd}$ ON THE PRIMITIVE BASE
                  MULTIPLY $P_{\zeta\eta,abcd}$ WITH PREFACTOR
                  TRANSPOSE $P'_{\zeta\eta,abcd}$ TO $P'_{\eta,abcd,\zeta}$
                  COMPUTE $W(\eta)_\zeta$ OR $Q(\eta)_\zeta$
                  LOOP OVER $\zeta$
                    SKIP IF $W_\zeta$ OR $Q_\zeta$ LESS THAN THRESHOLD
                       GATHER DATA
                  NEXT $\zeta$
                  TRANSPOSE $P'_{\eta,abcd,\zeta'}$ TO $P'_{abcd,\zeta',\eta}$
                  COMPUTE $W(\zeta')_\eta$ OR $Q(\zeta')_\eta$
                  LOOP OVER $\eta$
                    SKIP IF $W_\eta$ OR $Q_\eta$ LESS THAN THRESHOLD
                       GATHER DATA
                  NEXT $\eta$
                  TRANSPOSE $P'_{abcd,\zeta',\eta'}$ TO $P'_{\zeta',\eta',abcd}$
                  COMPUTE GRADIENTS AND CONTRACT WITH $P'_{\zeta',\eta',abcd}$
               NEXT $\hat{T}$
             NEXT $\hat{S}$
          NEXT $\hat{R}$
        NEXT SHELL D
     NEXT SHELL C
   NEXT SHELL B
NEXT SHALL A
```

We will in the coming section demonstrate the efficiency of the algorithms as outlined above. This will be both in the form of theoretical FLOP counts and as comparisons between the integral gradient program ALASKA and other implementations.

3 Performance assessment

3.1 Theoretical FLOP count

Before we make a comparison with other methods let us compare the present method with what would be a traditional implementation of the Rys–Gauss quadrature. The FLOP count for two typical shell quadruplets are listed in Table 1. From this compilation we see that the new method will reduce the total FLOP count at the uncontracted level with about 20%.

In the traditional analysis of the FLOP count it is standard to divide the algorithm into three sublevels according to the procedure suggested by Hegarty and van det Velde [27]. These levels are the K^4, the K^2, and the K^0 levels which correspond to the uncontracted, the half contracted, and fully contracted integrals, respectively. To each of these levels are assigned constants x, y, and z, respectively, which represent the FLOP count associated with manipulating the integrals in each level of contraction. At the time of developing this procedure to analyse the benefits of a given integral method it was assumed that basis sets were of the segmented type. This explains why the FLOP count of the contraction step, $c * (m^4M + m^3M^2 + m^2M^3 + mM^4)$ where m is the number of Gaussian exponents of a shell and M is the number of contracted functions of the same shell, dose not put an explicit M dependency into x (M = 1 for a segmented basis set). To correct for this deficiency of earlier investigations we will in this paper explicitly list each contribution. The FLOP counts of the present method and the HGP method are presented in Table 2. The comparison is limited to these two methods due to the limited data available in the literature. It is quite clear, however, that the earlier conclusion still holds that the BRACKET and PRISM methods will suffer when state of the art basis sets are used. In the generation of the HGP numbers in Table 2 we have from the x parameter subtracted the contribution from the contraction. Assuming that the integrals have been premultiplied with the contraction coefficient in a K^2 loop we associate the contraction with 1 FLOP per entity to contract. In the new contraction, however, we will assume no premultiplication and the contraction step will carry 2 FLOP per entity to contract. The number of entities to contract in the HGP

Table 1. FLOP count for a traditional implementation of the Rys–Gauss quadrature and the reduced multiplication scheme for some shell quadruplets. The FLOP count is for the computation of 9 of the 12 possible cartesian gradients of a shell quadruplet

Shell quadruplet:	$(ss\|ss)$		$(pp\|pp)$	
Step	a	b	a	b
Roots and weights	13	13	78	78
2D-integrals	39	39	246	246
Transfer equation	6	6	532	532
Diff. 2D-integrals	3	3	216	216
Assembling of gradient	24	6	5832	4269
Total	85	67	6904	5341

[a] Traditional implementation; [b] present method

Table 2. FLOP count for the HGP method and the reduced multiplication scheme of the Rys–Gauss quadrature for two typical shell quadruplets. The FLOP count is for the computation of 9 of the 12 possible cartesian gradients of a shell quadruplet[a]

Shell quadruplet:	$(ss\,\vert\,ss)$		$(pp\,\vert\,pp)$	
Scaling level	HGP	Present	HGP	Present
Mm^4	20	2	1116	162
m^4 (x)	$85+1S$	$67+1S$	$2742+1S$	$5341+1S$
M^2m^3	20	2	1116	162
M^3m^2 (y)	20	2	1116	162
M^2m^2	0	0	0	0
M^4m	20	2	1116	162
M^4 (z)	3	6	3100	6

[a] S indicates a square root

methods were computed to be 10, and 558, respectively, for the shell quadruplets of $(ss\,|\,ss)$, and $(pp\,|\,pp)$, respectively. The first thing to note from Table 2 is obviously that the most expensive step in the gradient calculation scales as m^4M rather than m^4. To further analyse the merits of the two methods it will be illustrative to compile the actual flop counts for the two shell quadruplets for the basis sets $(6s3p/2s1p)$, and $(14s9p/6s5p)$, respectively. These basis sets correspond to the STO-3G and ANO basis of carbon and would represent the two extremes of a 1st row basis set. For the $(ss\,|\,ss)$ shell quadruplet we have 185 and 95 kFLOP for the HGP and the present method, respectively, for the STO-3G type of basis. The same numbers for the ANO basis are 11.1 and 3.4 MFLOP, respectively. For the $(pp\,|\,pp)$ shell quadruplet of STO-3G we have 359 and 452 kFLOP for the HGP method and the present method, respectively. As the corresponding numbers for the basis of ANO type we find 94 and 46 MFLOP, respectively. These figures show that at least theoretically the reduced multiplication scheme of the Rys–Gauss quadrature is competetive in the STO-3G range of basis sets and gets increasingly superior for larger basis sets.

Let us also mention an advantage of the Rys–Gauss quadrature in conjunction with direct implementations of correlated methods. Here one will generate a partial integrals list with at least one SO index fixed. In the case of a basis set of general contraction type this will be rather inefficient and result in much redundant work since some of the basis functions will have the same primitive set in common. Hence this fact advocates that the fixed index should be kept in the primitive basis and presumably not transformed to the contracted basis until the summation over the free indices is performed. Judging from the superior theoretical FLOP count of the Rys–Gauss quadrature for primitive basis set it must be regarded as strong candidate for the implementation in such a direct scheme.

3.2 Comparison with other implementations

In this section we will do a brief comparison with some other implementations of *ab initio* gradient techniques as implemented on an IBM RISC System/6000

550 machine. The implementations to compare with are the TURBOMOLE [28] and ABACUS [29] quantum chemistry programs. The former is based on the Rys–Gauss quadrature and the latter is based on the McMurchie–Davidson scheme [30]. The programs also have two different approaches towards symmetry. While TURBOMOLE works with the petite list [31] approach ABACUS and ALASKA employ the double coset representative approach [26]. The reason for this difference is primarily historic. Whereas TURBOMOLE is especially tailored for the SCF type wavefunction both ABACUS and ALASKA are developed with the aim for calculation of correlated wavefunctions. The two-electron integral transformation which is an essential step in most post-SCF procedures was not practically possible with petite list until quite recently [32, 33]. Hence, the different choice of incorporating symmetry in the package. The analysis here is not to judge these aspects of the computation. Thus, we will primarily compare the integral times excluding time due to symmetrization when we compare TURBOMOLE with the present implementation while comparisons with ABACUS will include the symmetrization time too. Furthermore, when prescreening is of importance we will select the same prescreening thresholds for the different packages. This will not make the comparison clear cut since the program packages employ different prescreening methods. As a matter of fact, for molecules for which the prescreening is of importance a more elaborate comparison involving several thresholds and a comparison of the resulting errors in the computed gradients would have to be employed in order to do a fair comparison if we were to monitor how effective the prescreening is or a calculation of the gradients with a zero threshold would have to be employed if we were to compare how effective the gradient formation is for molecules where the four center integrals dominated the ERI list. We will not in this paper, however, involve us with too complicated comparisons but do rather present the CPU times as they are if default thresholds are chosen. The calculations will be made for the four molecules acetylene, cubane, 2-pentenal, and chlorofluoromethanol. The first two molecules exhibit high symmetry and the latter two have no symmetry at all. The calculations of cubane and 2-pentenal will be dominated by the four-center integrals. The 2-pentenal molecule is in the *trans* 2-pentenal ($CH_3CH_2CH{=}CHCHO$, methyl *gauche* and CO *syn* the double bond) configuration. The basis sets employed are the STO-3G [34], the DZP [35], the TZ2P [36], and the average ANO [15] basis sets. The ANO basis sets will be used with two different shell extensions. The molecules have all been relaxed with respect to the SCF energy. The SCF energies and some geometrical parameters of the molecules mentioned above are listed in Tables 3–6. Observe that we employ the

Table 3. The SCF energy and the geometrical parameters of acetylene, C_2H_2, for some basis sets

Basis set	E_{SCF}/hartree	R_{CC}/Å	R_{CH}/Å
STO/3G	−75.856248	1.168	1.065
DZP	−76.831724	1.192	1.061
TZ2P	−76.850024	1.180	1.055
ANO(*spd*)[a]	−76.852607	1.180	1.054
ANO(*spdf*)[b]	−76.856268	1.180	1.054

[a] Contracted to C(6*s*5*p*3*d*) and H(3*s*2*p*); [b] Contracted to *C*(6*s*5*p*3*d*2*f*) and H(3*s*2*p*3*d*)

Table 4. The SCF energy and the geometrical parameters of cubane, C_8H_8, for some basis sets

Basis set	E_{SCF}/hartree	R_{CC}/Å	R_{CH}/Å
STO-3G	−303.781400	1.562	1.086
DZP	−307.447488	1.560	1.083

Table 5. The SCF energy and some geometrical parameters of the *trans* 2-pentenal molecule, $CH_3CH_2{=}CHCHO$, for some basis sets[a]

Basis set	E_{SCF}/hartree	bond/Å	angle/deg	dihedral/deg	dihedral/deg
		C=C	CCO	CCCO	CCCC
STO-3G	−265.470279	1.316	124.1	0.3	120.2
DZP	−268.892751	1.330	125.2	0.3	120.7

[a] A full list of the geometrical parameters is available from the author

Table 6. The SCF energy and some geometrical parameters of the chlorofluoromethanol molecule, ClFCHOH, for some basis sets[a]

Basis set	E_{SCF}/hartree	R_{ClC}/Å	R_{FC}/Å	R_{CO}/Å	ClCF ∠/deg	COH ∠/deg
STO-3G	−665.012939	1.830	1.367	1.419	109.8	103.7
DZP	−672.835681	1.760	1.340	1.352	109.3	109.6
TZ2P[b]	−672.871934	1.763	1.336	1.350	109.0	109.7

[a] A full list of the geometrical parameters is available from the author; [b] The chlorine basis is contracted as (11*s*7*p*2*d*/7*s*5*p*2*d*) with the exponents of 1.13 and .38 in the *d*-shell

pure spherical harmonic components of the basis functions of higher angular momentum for ALASKA whereas the other program will use the cartesian representation.

The CPU time of the various packages and molecules are collected in Table 7. Let us here discuss the results molecule by molecule. However, before we start note that ALASKA is written for general wavefunctions and in addition to the normal integral gradient computation it also performs a backtransformation of the 2nd order density matrix from AO to primitive base and, in the case of spherical harmonics, a projection of the same to the cartesian representation. These steps normally constitute 15–30% of the total time. Furthermore, to facilitate computations with ANO basis sets and correlated wavefunctions the prescreening is performed on the 2nd order density matrix in the primitive basis. Hence, keep this in mind when comparing the timings with TURBOMOLE which is developed in particular for the SCF wavefunctions for which the 2nd order density matrix is defined by the 1st order density matrix, i.e. no four index transformations are needed, and for which the prescreening is done on the first order density matrix in the contracted basis.

First, the acetylene molecule is in the integral and gradient calculation dominated by two- and three-center integrals and a lot is to be gained in efficiency if triangularization of symmetrical matrices is properly implemented. Furthermore, we expect the influence of the prescreening on the computation

Table 7. Timing in CPU seconds on a IBM RISC System/6000 550 machine, for ERI integral and gradient evaluation for a series of molecules with different basis set. Bracketed numbers include timing due to the symmetry treatment[a]

		SEWARD/ALASKA		ABACUS[b]		TURBOMOLE	
		Integral	Gradient	Integral	Gradient	Integral[c]	Gradient
Molecule	Basis						
C_2H_2	STO-3G	0.39{0.40}	0.90{0.93}	{0.59}	{0.84}	—	2
	DZP	5.1{6.0}	10.4{11.1}	{6.2}	{11.8}	—	19
	TZ2P	20{21}	33{34}	{17}	{35}	—	66
	ANO(*spd*)	100{102}	242{244}	{78}	{188}	—	[d]
	ANO(*spdf*)	432{446}	1211{1224}	{329}	{856}	—	[d]
C_8H_8	STO-3G	26{59}	74{126}	{74}	{313}	—	75
	DZP	418{2045}	821{3260}	{2467}	[e]	—	772
2-pentenal	STO-3G	59	121	78	183	—	138
	DZP	1055	1567	1331[f]	[f]	—	1432
ClFCHCOH	STO-3G	9	26	12	30	—	30
	DZP	128	242	166	546	—	274
	TZ2P	560	733	619[f]	[f]	—	675

[a] All calculations were performed at the relaxed geometry of the molecule with respect to the chosen basis set. The prescreening threshold, when available, was set at 10^{-7}; [b] The ABACUS package does not offer prescreening; [c] TURBOMOLE is a direct SCF program and no explicit timing for computing all integrals once is available; [d] The program cannot handle general contraction explicitly; [e] The program aborted during the gradient formation due to a segmentation fault; [f] The program aborted during the formation of the super matrix file due to disc storage problems

time to be negligible. For this molecule and for all basis sets where a comparison could be made we observe almost identical performance for SEWARD/ALASKA and ABACUS while TURBOMOLE is lagging behind with about a factor of two. The performance of ALASKA versus TURBOMOLE in the calculation of the gradients of the acetylene molecule could be an indication of the improvement within the framework of the Rys–Gauss quadrature. However, it could just as well be due to no or poor implementation of triangularization in TURBOMOLE. Furthermore, the slightly better performance of ABACUS as compared to ALASKA for the acetylene is most likely due to either a better implementation of the symmetry treatment or a reflextion of a better utilization of the simplifications which arise in the calculation of 3-, and 2-center ERI integrals. Second, the cubene molecule calculation is dominated by four-center integrals and the prescreening will be of importance in this case. For this molecule we see a significant improvement by TURBOMOLE as compared to the acetylene calculation. This could be due to the fact that the calculation here will not suffer from a poor implementation of triangularization. The improved performance could also be due to much more efficient prescreening as compared with ALASKA. This is, however, not very likely. What is quite evident though is that the poorer performance of ABACUS for the molecule is mainly due to the lack of prescreening in that package. In comparing the CPU times between ALASKA or ABACUS and TURBOMOLE of the cubene molecule we get a worst case comparison between double coset decomposition and the petite list implementation of symmetry. The results here indicate that the former is so much more expensive for the higher point groups that it is worth while to implement a formation of the desymmetrized 2nd order density matrix directly from desymmetrized first order density matrices in the case of a SCF calculation. Third, for the 2-pentenal molecule conventional SCF calculations were possible, for STO-3G and DZP basis sets. Here we observe the same trend as for the cubene molecule with the exception that ABACUS and ALASKA do not carry any extra time due to symmetrization. The lack of prescreening facilities in ABACUS is making itself evident again. Fourth and final, the ClFCHOH molecule leaves us with much the same picture as the other molecules with the possible exception that it strongly advocates the usage of prescreening in compact molecular systems as well. There is no other way of explaining the poor performance of ABACUS as compared to the other implementations.

As a summary of this analysis we conclude that all three packages have very similar timings and for the differences are mainly due to prescreening and how the symmetry is implemented for the SCF wavefunction. These differences could easily be removed leaving the three packages with virtually the same performance. Hence, the choice of which package to use would then be guided by other aspect than the CPU time.

In the coming subsections we will study *i*) the ability to utilize symmetry constraints, *ii*) CPU time as a function of the basis set quality, and *iii*) efficiency of prescreening of the present implementation, ALASKA.

3.3 Symmetry constraints

To monitor the ability to utilize the symmetry constraints with the presented method and its implementation into ALASKA we will study the CPU time of the integral and gradient evaluation of the acetylene molecule as we reduce the symmetry which is employed in the calculation (see Table 8). The results here show that the ratio of the CPU times for the gradient and integral evaluation increases as the utilization of the symmetry is reduced. This is exactly what we expect, i.e. the introduction of symmetry will not only reduce the shell quadruplets which we will handle but also the gradients to compute of a particular shell quadruplet. This is especially apparent when we reduce the symmetry from D_{2h} to C_{2v} by removing one of the mirror planes which is parallel to the bond of the

Table 8. The CPU time in seconds of the integral and gradient evaluation of acetylene, C_2H_2, for ANO(*spd*) basis set versus the symmetry point group utilized in the calculation. The number in parenthesis is the ratio gradient to integral CPU time

Point group	Integral	Gradient
D_{2h}	102	247 (2.42)
C_{2v}	106	346 (3.26)
C_s	113	436 (3.86)
C_1	216	795 (3.68)

molecule. In this case both calculations will have the same set of shell quadruplets and the only computational difference is that more gradients per shell quadruplets will be computed in the latter case. It is actually not until we totally eliminate the symmetry that we will increase the number of shell quadruplets to take into consideration. Here we see that the ratio of gradient to integral CPU time goes from 2.4 to a factor of about 3.7 when we reduce the symmetry.

3.4 Basis set versus CPU time

The results of Table 7 indicate the following general trends with respect to the basis set quality. For the basis set of STO-3G, DZP, TZ2P, and ANO we observe the ranges of the ratio for the gradient to integral CPU time of 2.0–2.9, 1.5–2.0, 1.3–1.7, and 2.4–2.8, respectively. The somewhat poorer performance of the STO-3G in general is most likely attributed to a larger degree of overhead as compared to the other basis sets. The results which we report for the DZP and TZ2P basis sets are exceptionally good as compared to other methods which usually lie in the range of 2–3. The ANO basis sets tend to give rise to a somewhat higher ratio. In particular we note that the ratio does not increase significantly as we include shells of higher angular momentum. This is very encouraging since it will allow us to do geometry optimizations of smaller molecules with state of the art basis sets without a too large increase in the CPU time relative the integral time.

3.5 Prescreening

The 2-pentenal molecule with the DZP basis set was chosen as the probe of the two prescreening methods suggested in this paper. The 2DO and 2DI prescreening methods were tested for various cut-off thresholds, ϑ. In Table 9 is recorded the relative CPU time (relative to no prescreening) of the calculation versus the threshold and the root-mean-square (RMS) error of the gradient as compared to the gradient computed with no prescreening. Furthermore, in Fig. 1 we plot the relative CPU time of the calculation versus $-\log(\text{RMS})$ in order to monitor the efficiency of the methods suggested here. The first impression from these calcula-

Table 9. The relative CPU time for the gradient evaluation of the 2-pentenal molecule for a DZP basis set at the corresponding relaxed geometry versus the cut-off threshold, ϑ, and the root-mean-square (RMS) error of the gradient for the 2DO and 2DI prescreening methods.

	2DO		2DI	
ϑ	RMS	Percent CPU time	RMS	Percent CPU time
10^{-5}	$1.91E-3$	22.0	$5.00E-4$	26.5
10^{-6}	$2.79E-4$	26.4	$5.66E-5$	31.6
10^{-7}	$2.13E-5$	31.1	$7.11E-6$	36.0
10^{-8}	$2.16E-6$	35.6	$5.42E-7$	40.6
10^{-9}	$4.21E-7$	39.4	$1.04E-7$	44.3
10^{-10}	$3.29E-8$	43.1	$1.51E-8$	49.3

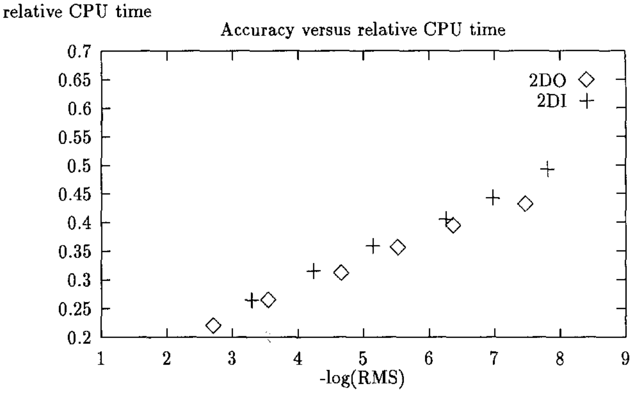

Fig. 1. The accuracy of the gradient of the 2-pentenal molecule with a DZP basis set versus the relative CPU time for the evaluation of the gradient using the 2DO and 2DI prescreening methods. The accuracy of the gradient is evaluated as the root-mean-square of the error of the gradient as compared to no prescreening

tions is that prescreening is of such an importance that it would be rather foolish not to use it in one form or another. However, it is noted quite surprisingly that the more sophisticated prescreening (2DI) is not more efficient than the the 2DO prescreening. This observation is in conflict with the study of Horn et al. who found that the use of a bounded integral gradient estimate will give a more efficient prescreening. Let us here point out the major differences in the two implementations in our search of a possible explanation to this disagreement. First, the prescreening by Horn et al. is implemented for the contracted 1st order density matrix whereas we apply the prescreening based on the uncontracted second order density matrix. Second, the Horn et al. implementation is based on finding the largest element of a given batch whereas the approach suggested here is using a norm technique. Third, the radial overlap approximation differs slightly between the two implementations. With the data given by this investigation it is not clear which of these factors give the major contribution to the difference. However, let us note that in accordance with Horn et al. the prescreening based on the Cauchy–Schwarz inequality has a closer correlation between the chosen threshold and the actual accuracy of the computed gradient as compared to the prescreening based on the radial overlap. The relation between the threshold and the actual accuracy of the 2DI prescreening is as good as in the implementation of Horn et al. Hence, this excludes any errors in the implementation. Furthermore, the curve for the 2DO method is much smoother than the radial overlap approximation of Horn and coworkers. This could possibly either indicate some problems with the implementation in TURBOMOLE or as a sign that the version of the radial overlap approximation suggested here is superior. In addition, it is noted that the use of an upper bound for the integral etimated does not imply that the prescreening will be more efficient. The only obvious effect of using such an estimate is that the prescreening will be safer.

4 Conclusions

The reduced multiplication scheme of the Rys–Gauss quadrature has been extended to the evaluation of the ERI 1st derivatives needed for the evaluation of the molecular gradient of *ab initio* energy functionals. Furthermore, simplifications in the calculation of the 1st order derivatives of ERIs of two- and three-central integrals by the Rys–Gauss quadrature have been introduced. Finally, two approaches to prescreening based on the 2nd order density matrix in the primitive basis have been presented. All the new methods have been implemented in the integral gradient program ALASKA.

The major highlights of the study here is, *i*) the integral gradient evaluation scales as Mm^4 rather than as m^4 in the case of basis set of general contraction type as compared to segmented basis sets, *ii*) the reduced multiplication scheme of the Rys–Gauss quadrature for the calculation of the integral derivatives is efficient and flexible as compared to other methods and implementation, *iii*) the method suggested here offers significant speedup in connection with the presence of symmetry in the molecule under study, *iv*) prescreening without bounded estimate of the integral gradient is demonstrated to be as efficient as other methods based on bounded estimates (Cauchy–Schwarz), and *v*) the method suggested here is well suited for correlated wavefunctions and for basis sets of general contraction type.

Steps to extend the presented implementation to the MP2, MCSCF, and CASPT2 energy functionals are currently under way at this laboratory.

Acknowledgements. The author wants to express his gratitude to Dr. T. Helgaker for the use of the ABACUS quantum chemistry package. This project has been supported by a grant from the Swedish Natural Research Council (NFR) and IBM Sweden under a joint study contract.

References

1. Obara S, Saika A (1986) J Chem Phys 84:3963
2. Head-Gordon M, Pople JA (1988) J Chem Phys 89:5777
3. King HF, Dupuis M (1976) J Comput Phys 21:144; Dupuis M, Rys J, King HF (1976) J Chem Phys 65:111; Rys J, Dupuis M, King HF (1983) J Comput Phys 4:154
4. Hamilton TP, Schaefer III HF (1991) Chem Phys 150:163
5. Lindh R, Ryu U, Liu B (1991) J Chem Phys 95:5889
6. Ryu U, Lee YS, Lindh R (1991) Chem Phys Lett 185:562
7. Gill PMW, Head-Gordon M, Pople JA (1989) Int J Quantum Chem Symp 23:269
8. McMurchie LE, Davidson ER (1978) J Comput Phys 26:218
9. Gill PMW, Head-Gordon M, Pople JA (1990) J Chem Phys 94:5564
10. Gill PMW, Pople JA (1991) Int J Quantum Chem 40:753
11. "HONDO A General Atomic and Molecular Electronic Structure System", by Dupuis M, Maluendes SA (1991) Chap. 12, In: Clementi E (ed)., "MOTECC-91: Modern techniques in computational chemistry", ESCOM, Leiden
12. Amos RD, Rice JE (1987) CADPAC: the Cambridge Analytical Derivative Package, Issue 4.0, Cambridge, UK
13. GRADSCF, Komornicki A, Polyatomics Research Institute, Mountain View, California
14. Rice JE, Amos RD (1985) Chem Phys Lett 122:585
15. Widmark PO, Malmqvist, P-Å, Roos BO (1990) Theoret Chim Acta 77:291; Widmark PO, Persson BJ, Roos BO (1991) ibid 79:419
16. Almlöf J, Taylor PR (1987) J Chem Phys 86:4070
17. Raffenetti RC (1973) J Chem Phys 58:4452
18. Horn H, Weiß H, Häser M, Ehrig M, Ahlrichs R (1991) J Comput Chem 12:1058
19. Amos RD (1984) Chem Phys Lett 108:347
20. Dupuis M (1986) In: Jørgensen P, Simons J (eds) Geometrical derivatives of energy surfaces and molecular properties. Reidel, Dordrecht
21. Saxe P, Yamaguchi Y, Schaefer HF (1982) J Chem Phys 77:5647
22. Schlegel HB (1982) J Chem Phys 77:3676; Schlegel HB, Binkley JS, Pople JA (1984) ibid 80:1976

23. MOLCAS-3 is an experimental extension of the quantum chemistry package MOLCAS-2, MOLCAS version 2, Andersson K, Fülscher MP, Lindh R, Malmqvist P-Å, Olsen J, Roos BO, Sadlej AJ (1991) Univ of Lund, Sweden, and Widmark PO, IBM Sweden
24. Taylor PR (1986) Theoret Chim Acta 69:447; (1989) ibid 76:147
25. Dupuis M, King HF (1976) Int J Quantum Chem 11:613; Takada T, Dupuis M, King HF (1981) J Chem Phys 75:332; Takada T, Dupuis M, King HF (1983) J Comput Chem 4:234
26. Davidson ER (1975) J Chem Phys 62:400
27. Hegarty D, van der Velde G (1983) Int J Quantum Chem 23:1135
28. Häser M, Ahlrichs R (1989) J Comput Chem 10:104
29. Helgaker TU, Almlöf J, Jensen HJA, Jørgensen P (1986) J Chem Phys 84:6266
30. McMurchie LE, Davidson ER (1978) J Comput Phys 26:218
31. Dupuis M, King HF (1977) Int J Quantum Chem 11:613
32. Häser M, Almlöf J, Feyereisen MW (1991) Theoret Chim Acta 79:115
33. Hollauer E, Dupuis M (1992) J Chem Phys 96:5220
34. Hehre WJ, Stewart RF, Pople JA (1969) J Chem Phys 51:2657; Collins PB, Schleyer PvR, Binkley JS, Pople JA (1976) ibid 64:5142
35. Dunning TH, Hay PJ (1976) in: Modern theoretical chemistry. Plenum, NY, Chap 1, p 1
36. Exponents: Huzinaga S (1965) J Chem Phys 42:1293; coefficients: Dunning TH (1971) J Chem Phys 55:716; (1986) J Chem Phys 84:6266