Theor Chim Acta (1995) 92: 339–349

Theoretica Chimica Acta

An efficient data compression method for the Davidson subspace diagonalization scheme

Holger Dachsel*, Hans Lischka

Institut für Theoretische Chemie und Strahlenchemie der Universität Wien, Währingerstraße 17, A-1090 Wien, Austria

Received December 2, 1994/Accepted May 23, 1995

Summary. A flexible and efficient compression scheme for the expansion and product vectors Hamiltonian matrix times expansion vectors is presented within the Davidson diagonalization method. Our approach is based on an error analysis of the energy in terms of the aforementioned vectors and on a compression scheme for representing floating point numbers with a variable length mantissa. For a selection of typical quantum chemical test cases total saving factors of up to ten are reported. The method is expected to work especially well for extended multi-reference CI and full CI cases. As a general outcome of our analysis we obtain limits of possible sizes of a CI expansion within the Davidson procedure in relation to the energy and the desired accuracy of the energy assuming the usual IEEE floating point standard.

Key words: Davidson procedure – Data compression scheme – Error analysis – Variable floating point representation

1 Introduction

The processing of large amounts of data which arise in *ab initio* calculations have always been a severe bottleneck for these methods. This problem is of increasing importance since the computational power in terms of floating-point operations is strongly enhanced from year to year whereas disk space, I/O bandwidth and available central memory do not keep up with these developments. Therefore, several ways out of this dilemma have been sought. The most straightforward one is to avoid these data totally by developing new algorithms. The "direct" approach has been extremely successful with circumventing the storage of the two-electron integrals on disk for SCF and MP2 calculations [1–3]. Unfortunately, the situation is much more complex in cases like CI. There, the two-electron integrals have to be transformed and stored and CI vectors have to be processed and stored as well. Expansion lengths of several millions are routine nowadays for MRCI calculations and of several billions for FCI calculations [4, 5]. The lowest eigenvalues

* *Present address*: Pacific Northwest Laboratories, Richland, Washington 99352, USA

and eigenvectors of the Hamiltonian matrix are usually determined by the subspace expansion method developed by Davidson [6]. Storage of the necessary vectors (preferentially in central memory) poses a problem because of the size of the expansion lengths. Even though the numerical operations are usually done in 64 bit arithmetic, it is not necessary to store the resulting data in full precision. Therefore, various data compression schemes have been developed. Harrison and Handy [7] and Knowles [8], and later Olsen et al. [4] and Mitrushenkov [5] have used fixed-point truncation schemes in their FCI calculations. Shepard [9] has analyzed the truncation error within first-order iterative schemes and the speed of convergence in more detail in dependence on the number of bits used for the expansion vectors. The situation is more difficult for the product vectors Hamiltonian-matrix times expansion vectors (which are equally important in terms of data size). They were not compressed at all [9] or only after having computed the respective scalar products with the previous expansion vectors in full precision [8]. This fact makes the storage of at least one product vector in full length necessary. Data compression schemes have been developed for the two-electron integrals as well [10–12] since they constitute, of course, an important factor in terms of storage requirements in *ab initio* calculations. However, the aspects of two-electron integral compression will not be discussed further here.

The computational aspects of the Davidson method, especially with regard to I/O, have been discussed by Shepard [9]. As has been pointed out by him, the subspace manipulation section of the whole Davidson procedure is easily I/O bound. Thus, by efficiently compressing the expansion vectors this I/O can be reduced substantially or removed completely if the compressed vectors can be kept completely in central memory. On parallel computers, I/O is even a much more severe bottleneck than on sequential ones. Thus, data reduction schemes will be even more effective in parallel computing. Even though the compression scheme developed by Shepard works efficiently, there are a number of serious drawbacks in his method. Most importantly, as has already been mentioned, only half of the relevant vectors are compressed. This fact severely limits the overall savings which can be achieved to a maximum factor of two. Moreover, his scheme only handles a fixed overall bit-length representation irrespective of the size of the individual vector elements. In order to automatically determine the proper truncation level some knowledge of the full-precision convergence rate is required which makes actual implementations difficult.

In this work we want to present an improved data compression approach within the Davidson diagonalization method. We want to develop a scheme where we do not have one single cutoff value but where the number of digits used to represent a given vector element varies smoothly according to its magnitude. We also want to include the products Hamiltonian-matrix times trial vector.

2 Formalism

2.1 The Davidson scheme

In order to develop our compression scheme let us start with the matrix eigenvalue equation

$$\boldsymbol{Hc} = E\boldsymbol{c}, \tag{1}$$

where $\boldsymbol{H}$ is the Hamiltonian matrix of dimension N_{CI} and $\boldsymbol{c}$ and E are its eigenvector and eigenvalue, respectively. In the subspace expansion method of Davidson [6],

the eigenvector $\boldsymbol{c}$ is approximated by a vector $\boldsymbol{u}$ which is expanded into a linear combination of correction vectors $\boldsymbol{v}_i$:

$$\boldsymbol{u} = \sum_{i=1}^{N} \alpha_i \boldsymbol{v}_i, \tag{2}$$

where N is the dimension of the subspace and the expansion coefficients α_i are determined from the small eigenvalue problem

$$\bar{\boldsymbol{H}}\alpha = \bar{E}\boldsymbol{S}\alpha, \tag{3}$$

where $\bar{H}_{ij} = \boldsymbol{v}_i^{\dagger}\boldsymbol{H}\boldsymbol{v}_j$ and $S_{ij} = \boldsymbol{v}_i^{\dagger}\boldsymbol{v}_j$. We use an orthogonal set of expansion vectors $\boldsymbol{v}_i$ which are, however, not normalized. From the residuum vector

$$\boldsymbol{r} = (\boldsymbol{H} - \bar{E})\boldsymbol{u} \tag{4}$$

a new expansion vector $\boldsymbol{v}_{N+1}$ is computed (for details see [6]) and an improved approximation $\boldsymbol{u}$ according to Eq. (2) is determined. This iteration scheme is continued until a given convergence limit is reached.

In large-scale CI calculations the most time-consuming step is the calculation of the matrix-vector product

$$\boldsymbol{w}_i = \boldsymbol{H}\boldsymbol{v}_i. \tag{5}$$

As has already been pointed out in the Introduction, the dimension of the eigenvalue problem (1) can easily reach several million configurations. Thus, $2N$ vectors of that expansion length have to be stored. This can be either done in central memory or else, if this is not sufficient, on disk. In either case it is highly desirable to reduce the actual amount of data. In the first case, larger CI expansion sets can be held completely in central memory, and in the second case disk I/O is reduced.

Our aim is to derive relations between a given error

$$\Delta\bar{E} = \bar{E} - E \tag{6}$$

in the energy and errors in the $\boldsymbol{v}$ and $\boldsymbol{w}$ vectors. We do so by looking first at errors in $\boldsymbol{u}$ and in

$$\boldsymbol{p} = \boldsymbol{H}\boldsymbol{u} \tag{7}$$

and relate these errors subsequently to those in the $\boldsymbol{v}_i$'s and $\boldsymbol{w}_i$'s. Then, these relations will be used to derive a compression scheme for these vectors.

2.2 Error analysis for the u vector

Let us start with the error analysis with respect to the $\boldsymbol{u}$ vector. We write the subspace energy as

$$\bar{E} = \frac{\sum_i \sum_j u_i H_{ij} u_j}{\sum_i u_i^2} \tag{8}$$

The summations in Eq. (8) and in all subsequent formulas extend – if not otherwise stated – over the dimension N_{CI} of the CI expansion. $\Delta_u\bar{E}$ (the index u in the following Eq. (9) is used to emphasize the expansion into the components of $\boldsymbol{u}$) is expanded into a Taylor series around the exact vector $\boldsymbol{c}$:

$$\Delta_u\bar{E} = \sum_i \left(\frac{\partial\bar{E}}{\partial u_i}\right)_c \Delta u_i + \frac{1}{2}\sum_i\sum_j \left(\frac{\partial^2\bar{E}}{\partial u_i \partial u_j}\right)_c \Delta u_i \Delta u_j + \cdots \tag{9}$$

with

$$\Delta \boldsymbol{u} = \boldsymbol{u} - \boldsymbol{c}. \tag{10}$$

The first derivatives are zero because of the stationarity property of Eq. (1). By straightforward differentiation we get for the second-order term

$$\Delta_u^2 \bar{E} = \frac{\sum_i \sum_j (H_{ij} - E\delta_{ij}) \Delta u_i \Delta u_j}{\sum_i c_i^2}, \tag{11}$$

which reduces to

$$\Delta_u^2 \bar{E} = \sum_i \sum_j (H_{ij} - E\delta_{ij}) \Delta u_i \Delta u_j \tag{12}$$

because the exact eigenvector $\boldsymbol{c}$ of Eq. (1) is normalized. In Eq. (12) we approximate the matrix $\boldsymbol{H}$ by its diagonal elements, E by $\bar{E}$ and replace the error Δu_i of each element u_i by a constant overall value Δu with

$$(\Delta u_i)^2 \leqslant (\Delta u)^2 \tag{13}$$

and get an approximate expression

$$\Delta_u^2 \bar{E}_{\text{approx}} = (\Delta u)^2 \sum_i (H_{ii} - \bar{E}) \tag{14}$$

for $\Delta_u^2 \bar{E}$. From the requirement that $|\Delta_u^2 \bar{E}_{\text{approx}}|$ is less than or equal to a given error threshold $|\Delta_u E|$ we obtain the following relation between $|\Delta u|$ and $|\Delta_u E|$:

$$|\Delta u| \leqslant \sqrt{\frac{|\Delta_u E|}{|\sum_i (H_{ii} - \bar{E})|}}. \tag{15}$$

This expression will be used later for the development of the compression scheme for the $\boldsymbol{v}_i$ vectors. The approximation of $\boldsymbol{H}$ by its diagonal elements is necessary in order to obtain manageable formulas. The same approximation is used in the Davidson method for calculating a new trial vector. Thus, in cases where $\boldsymbol{H}$ is not dominated by its diagonal elements anymore general problems with the entire Davidson scheme are to be expected (see also Sect. 3). It is important to note that we use this approximation only for the compression of the trial vectors and not for the vectors $\boldsymbol{w}_i$. Therefore, the variational character of the Davidson procedure will not be affected by this approximation.

2.3 Error analysis for the p vector

If one compresses the vector $\boldsymbol{p}$ for the current (compressed) vector $\boldsymbol{u}$ an additional error is obtained. The numerical operations involved in the matrix multiplication $\boldsymbol{H}\boldsymbol{v}_i$ itself leading to $\boldsymbol{p}$ are always carried out in full precision. No error analysis is attempted for that step.

In order to evaluate $\Delta_p \bar{E}$ we write $\bar{E}$ as

$$\bar{E} = \frac{\sum_i u_i p_i}{\sum_i u_i^2}, \tag{16}$$

and, upon expansion of $\boldsymbol{p}$ as

$$\boldsymbol{p} = \boldsymbol{p}_{\text{exact}} + \Delta \boldsymbol{p}, \tag{17}$$

obtain the result

$$\Delta_p \bar{E} = \sum_i u_i \Delta p_i = \sum_i u_i \operatorname{sign}(\Delta p_i) |\Delta p_i|, \tag{18}$$

since $\boldsymbol{u}$ is normalized. In analogy with Eq. (13), replacing $|\Delta p_i|$ by a constant value $|\Delta p|$ we get

$$\Delta_p \bar{E}_{\text{approx}} = |\Delta p| \sum_i \operatorname{sign}(\Delta p_i) u_i \tag{19}$$

and its absolute value is bounded from above:

$$|\Delta_p \bar{E}_{\text{approx}}| \leqslant |\Delta p| \sum_i |u_i|. \tag{20}$$

Thus, we require that for a given error threshold $\Delta_p E$

$$|\Delta_p E| \geqslant |\Delta p| \sum_i |u_i| \tag{21}$$

which leads to

$$|\Delta p| \leqslant \frac{|\Delta_p E|}{\sum_i |u_i|}. \tag{22}$$

2.4 Error analysis for the v and w vectors

Up to now we have obtained an error analysis related to the vectors $\boldsymbol{u}$ and $\boldsymbol{p}$ (Eqs. (15) and (22)). However, the basic quantities are the $\boldsymbol{v}_i$'s and $\boldsymbol{w}_i$'s and not the $\boldsymbol{u}$ and $\boldsymbol{p}$ vectors. In order to apply this analysis to the $\boldsymbol{v}_i$ and $\boldsymbol{w}_i$ vectors independently, we only have to make sure that the linear combination coefficients α_i of Eq. (3) do not increase the compression error. This could happen, e.g. by some arbitrary rescaling of the $\boldsymbol{v}_i$ vectors before compression. Normalization of the $\boldsymbol{v}_i$'s is not advisable since it would lead to degradation of our compression scheme. Since we want to achieve maximum efficiency in saving storage space and numerical stability, the vector $\boldsymbol{v}_{N+1}$ (N being the current subspace dimension) is rescaled before using it to compute $\boldsymbol{w}_{N+1}$. For that purpose we make the two-dimensional variational ansatz

$$\boldsymbol{u}_{(N+1)} = \alpha_1 \boldsymbol{u}_{(N)} + \alpha_2 \boldsymbol{v}_{N+1} \tag{23}$$

and rescale $\boldsymbol{v}_{N+1}$ by α_2.

All matrix elements for this small eigenvalue problem are readily available except for $H_{22} = \boldsymbol{v}_{N+1}^{\mathrm{t}} \boldsymbol{H} \boldsymbol{v}_{N+1}$. In this case, $\boldsymbol{H}$ is approximated by its diagonal elements in analogy to our procedures above.

2.5 Compression scheme

The requirement for the elements of the $\boldsymbol{v}_i$ and $\boldsymbol{w}_i$ vectors not to exceed given constant overall errors means that the number of significant digits in these elements varies and that this number will decrease with decreasing absolute values of the vector components. In order to make use of this situation, we devised a special floating point representation with a variable length of the mantissa and truncate the insignificant digits according to our error analysis. The conversion of the original, uncompressed numbers to our new scheme depends, of course, on the

original floating point representation. For the purpose of the presentation of the formalism, we take reference to the IEEE standard [13] because of its widespread use. However, the adaption to other floating point representations is straightforward.

Basically, there are two possibilities to store the information about the compressed numbers. In our first version we stored exponent (seven bits) and sign of the number in eight bits and then the mantissa. A storage scheme which is slightly more efficient by one bit [14] is given by storing the number of mantissa bits instead of the exponent since the maximum number of mantissa bits can be represented by six bits.

In the original floating point number, 12 bits are used for sign and exponent whereas only seven bits (six bits for the number of mantissa bits and one bit for the sign) are taken for the compressed numbers. Consecutive zeros are stored in a compact way by using a repetition factor. To find out how many bits in the mantissa are actually needed, the maximum error in dependence on the number of bits is calculated. We want to truncate all numbers such that the absolute errors in the $\boldsymbol{u}$ and $\boldsymbol{p}$ vectors do not exceed the absolute errors given by Eqs. (15) and (22). In Scheme 1, all numbers in absolute values which can be represented within the IEEE standard are listed in rows. If the absolute value of a number x_i is located in the interval given in the first line of Scheme 1, we utilize only the leading n_0 bits of the mantissa. Since in the following lines of Scheme 1 the limits for the numbers are always reduced by a factor of two, the number of bits used for the representation of the mantissa is reduced correspondingly by one. Conversely, if the limits increase by a factor of two then the number of bits is increased by one. All numbers $\leqslant 2^{-n_0-1}(2-2^{-52})$ in absolute value are set to zero. Thus, for a given number with exponent exp (see Scheme 1) we use

$$n_{\exp} = \min(\max((n_0 + \exp), 0), 52) \tag{24}$$

bits. The maximum error is found for the zero bit mantissa as $2^{-n_0}(2-2^{-52})$. Instead of truncating, the mantissa is actually rounded. This reduces the maximum error to $2^{-n_0}(1.5-2^{-52})$.

Substituting the maximum error defined in the previous paragraph for $|\Delta u|$ in Eq. (15) we compute a value $n_{0,u}$ as

$$n_{0,u} = -\log_2\left(\frac{\sqrt{|\Delta_u E|}}{(1.5-2^{-52})\sqrt{|\sum_i (H_{ii}-\bar{E})|}}\right). \tag{25}$$

Similarly from Eq. (22), $n_{0,p}$ is given as

$$n_{0,p} = -\log_2\left(\frac{|\Delta_p E|}{(1.5-2^{-52})\sum_i |u_i|}\right). \tag{26}$$

The actual number of bits for the mantissas are chosen as the next larger integers of $n_{0,u}$ and $n_{0,p}$, respectively.

So far, $\Delta_u E$ and $\Delta_p E$ have been treated independently. However, in practical applications one wants to specify only one energy threshold ΔE. We set $|\Delta_u E| = a|\Delta E|$ and $|\Delta_p E| = b|\Delta E|$ with $a+b=1$ and $a>0$, $b>0$ and optimize the target function

$$f(a,b) = n_{0,u}(a) + n_{0,p}(b). \tag{27}$$

Straightforward calculation gives $a = \frac{1}{3}$ and $b = \frac{2}{3}$.

$$
\begin{aligned}
&\vdots \\
&2^0(2-2^{-52}) \geqslant |x_i| \geqslant 2^0 && (\exp = 0),\ \text{mantissa length } n_0 \text{ bits} \\
&2^{-1}(2-2^{-52}) \geqslant |x_i| \geqslant 2^{-1} && (\exp = -1),\ \text{mantissa length } n_0 - 1 \text{ bits} \\
&\vdots \\
&2^{-n_0+1}(2-2^{-52}) \geqslant |x_i| \geqslant 2^{-n_0+1} && (\exp = -n_0 + 1),\ \text{mantissa length 1 bit} \\
&2^{-n_0}(2-2^{-52}) \geqslant |x_i| \geqslant 2^{-n_0} && (\exp = -n_0),\ \text{mantissa length 0 bit} \\
&\vdots
\end{aligned}
$$

Scheme 1

2.6 Analysis of the residuum

The analysis of the residuum follows the above lines. We derive a relation between r^2 and the energy convergence threshold ΔE. In analogy to Sect. 2.2, r^2 is expanded into a Taylor series in $\Delta \boldsymbol{u}$ resulting in the following expression:

$$r^2 = \sum_i \sum_j \sum_k (H_{ik} - E\delta_{ik})(H_{jk} - E\delta_{jk})\,\Delta u_i\,\Delta u_j + \cdots \tag{28}$$

Truncating after the term of second order, approximating $\boldsymbol{H}$ by its diagonal and substituting the Δu_i's by a constant value Δu gives

$$r^2_{\text{approx}} = (\Delta u)^2 \sum_i (H_{ii} - \bar{E})^2. \tag{29}$$

Setting Δu equal to the maximum error $2^{-n_{0,r}}(1.5 - 2^{-52})$ as defined in Sect. 2.5 gives

$$r^2_{\text{approx}} = 2^{-2n_{0,r}}(1.5 - 2^{-52})^2 \sum_i (H_{ii} - \bar{E})^2. \tag{30}$$

The value of $n_{0,r}$ is the next larger integer of the result calculated from Eq. (25) with the total convergence threshold ΔE replacing $\Delta_u E$. On convergence, the value of r^2 calculated according to Eq. (4) is less or equal than that given by Eq. (30).

2.7 Computational considerations

The compression scheme has been implemented into the COLUMBUS program system [15, 16] and is also available as a separate program. Bit packing routines have been written in Fortran. In the uncompressed version of the COLUMBUS program, blocks of data with fixed record length were written to disk. Because of the dynamic packing scheme this record length is now variable. A dynamic indexing scheme has been developed for bookkeeping purposes of the individual records. Routines written in C are used for bite-adressable I/O. The compression and decompression steps are performed in place.

On the basis of a given energy threshold the structure of the compression scheme and convergence criteria are determined automatically at the beginning of the program. Additionally, the packing scheme for the $\boldsymbol{w}$ vectors are adjusted dynamically in the course of the iterations. Eqs. (25), (26) and (30) have to be evaluated at the beginning of the calculation. Therefore, an estimate of $\bar{E}$ has to be made. Currently, we use for it the largest diagonal element of the Hamiltonian matrix in absolute value multiplied (somewhat arbitrarily) by a factor of 1.1. If it turns out that this estimate is not valid anymore in the course of the iterations, the

packing scheme for the v vectors is adjusted. At the beginning of the calculation, $\sum_i |u_i|$ in Eq. (26) is approximated by its upper bound $\sqrt{N_{CI}}$. Later, the current value of the sum is used.

It is important to note that the number of iterations does not increase by the compression scheme as compared to an equivalent calculation with the same energy threshold but without data compression.

3 Results and discussion

For the purpose of comparison with the previous work of Shepard [9], we start our presentation of applications with the modified Nesbet matrices

$$H_{ij} = 1 + \delta_{ij}\gamma(2i - 2) \quad \text{for } |i - j| \leqslant w \tag{31}$$

as defined in Ref. [9]. γ is a scale factor and w is a width parameter. The fully converged energies of Eq. (1) were obtained by the Davidson procedure using full double precision (64 bit) accuracy for storing the v and w vectors. They are expected to be accurate to all digits tabulated. In Table 1 two groups of examples are shown. The first group refers to $\gamma = 1$. In this case $\boldsymbol{H}$ is dominated by the diagonal elements. For these examples the savings increase dramatically with decreasing width w because of the efficient storage of consecutive zeros. Comparing with the results obtained by Shepard [9] ($N_{CI} = 10000$, $w = 9999$ and 99, see Figs. 2 and 5 in [9]) we find that at least 16 ($w = 9999$) and 8 ($w = 99$) bits had to be used in his work if the number of subspace iterations should not increase compared to the calculation in full precision. This gives savings of 4 and 8 for the v vectors but only 1.6 and 1.8 altogether. These numbers have to be compared with our total saving of 3.6 and 97.6.

In the second group of examples, the dominance of the diagonal elements compared to the off-diagonal ones has been reduced by choosing smaller values for γ. These examples should test the validity of replacing $\boldsymbol{H}$ by its diagonal values. Up

Table 1. Results for the modified Nesbet matrices[a, b, c]

w	γ	E^d	$\bar{E}^e - E$	No. of iterations	f_v[f]	f_w[g]	f_{tot}[h]
9999	1	0.16758194	8.5×10^{-7}	3	5.6	2.5	3.6
999	1	0.19808527	5.2×10^{-7}	5	18.0	7.7	11.1
99	1	0.25504648	7.8×10^{-7}	5	133.5	73.9	97.6
9	1	0.35975609	2.3×10^{-7}	5	895.5	641.0	758.6
9999	0.05	0.00993642	5.2×10^{-7}	5	7.1	2.6	4.0
9999	0.04	0.00796467	7.0×10^{-7}	5	6.8	2.6	4.0
9999	0.03	0.00598520	7.7×10^{-7}	5	6.8	2.7	4.0
9999	0.02	0.00399796	7.3×10^{-7}	5	11.1	2.7	4.6

[a] $N_{CI} = 10000$, for the definition of the parameters see Eq. (31)
[b] energy convergence threshold $\Delta E = 10^{-6}$
[c] all energies are given in a.u.
[d] fully converged, lowest eigenvalues, see Eq. (1)
[e] converged subspace energies, see Eq. (3)
[f] saving factors for the v vectors
[g] saving factors for the w vectors
[h] total saving factors

to a value of $\gamma = 0.02$ the Davidson method converges without problems. The calculations in full precision and with compression agree within the specified limits as they should. For $\gamma = 0.01$ convergence could not be achieved even in the full precision calculation. More sophisticated update methods for the construction of the trial vectors are necessary in this case. This example illustrates very nicely the validity of our assumption that as long as $\boldsymbol{H}$ is well represented by its diagonal in the original Davidson scheme this will also be the case for our compression formalism.

In Table 2, the results of a collection of configuration interaction calculations including all single and double substitutions from a set of reference wave functions (MRCISD) are shown. The computations on the CH_3 molecule were performed using a 7 orbital/7 electron CAS reference wave function. The basis sets were taken from the compilation by Dunning [17] and consisted of cc-pVDZ, cc-pVTZ and cc-pVQZ basis sets. For more information on these examples see Ref. [18]. The electronic ground state of butadiene was calculated using Dunning's cc-pVDZ and cc-pVTZ basis sets. The SR calculations were based on a closed-shell SCF calculation and the MR calculations on a 4 orbital/4 electron CAS in the π-system.

Two energy convergence thresholds of 10^{-4} and 10^{-6} have been chosen. As in the case of the previously discussed generalized Nesbet matrices, the compression for the $\boldsymbol{v}$ vectors is much more effective as compared to the $\boldsymbol{w}$ vectors. These findings come from the well-known general fact that the error in the eigenvalue depends quadratically on the error in the CI coefficients (see Eq. (12)) and is linear in the product vector (see Eq. (18)). Savings up to a factor of twenty could be achieved for the $\boldsymbol{v}$ vectors. Total saving factors range from four to ten. The smallest

Table 2. Results for various test examples[a]

	N_{CI}	E[b]	ΔE[c]	$\bar{E} - E$[d]	f_v[e]	f_w[f]	f_{tot}[g]
CH_3pVDZ	70254	−39.71416814	10^{-6}	1×10^{-7}	9.9	4.5	6.3
			10^{-4}	4×10^{-5}	17.0	6.5	9.8
CH_3pVTZ	624334	−39.75757058	10^{-6}	2×10^{-7}	11.7	4.6	6.7
			10^{-4}	6×10^{-5}	18.7	6.2	9.7
CH_3 C: pVQZ H: pVTZ	1058400	−39.76527616	10^{-6}	2×10^{-7}	10.9	4.5	6.5
			10^{-4}	6×10^{-5}	21.9	6.3	10.4
butadiene pVDZ-SR	82772	−155.42885615	10^{-6}	1×10^{-7}	6.5	2.9	4.1
			10^{-4}	3×10^{-6}	8.1	3.9	5.4
butadiene pVTZ-SR	290380	−155.53544485	10^{-6}	1×10^{-7}	6.6	2.8	4.1
			10^{-4}	3×10^{-6}	8.3	3.7	5.3
butadiene pVDZ-MR	1551170	−155.45426453	10^{-6}	2×10^{-7}	8.2	3.9	5.4
			10^{-4}	4×10^{-5}	10.4	5.2	7.1

[a] all energies are given in a.u.
[b] fully converged, lowest eigenvalues, see Eq. (1)
[c] energy convergence thresholds
[d] converged subspace energies, see Eq. (3)
[e] saving factors for the $\boldsymbol{v}$ vectors
[f] saving factors for the $\boldsymbol{w}$ vectors
[g] total saving factors

ones are obtained for the SR calculations since there one does not find so many relatively small CI coefficients (in absolute value) as compared to the MR cases and, thus, the compression cannot be as effective.

Besides the discussion of the specific examples shown in Tables 1 and 2, we can also draw some general conclusions for the Davidson procedure from our analysis. Since the vector $\boldsymbol{u}$ is normalized, its largest element in absolute value has to be less than or equal to one. In case of convergence of the Davidson scheme, $\boldsymbol{p} \approx \bar{E}\boldsymbol{u}$ and the largest elements in $\boldsymbol{p}$ are of the order $\bar{E}$. According to Sect. 2.5, we need for the representation of those numbers $n_{0,p} + \exp_{\bar{E}}$ bits. We allow a maximum number of 48 significant bits reserving four bits for rounding errors. Therefore, we have the following two conditions:

$$n_{0,u} \leqslant 48 \tag{32}$$

and

$$n_{0,p} + \exp_{\bar{E}} \leqslant 48. \tag{33}$$

$n_{0,u}$ and $n_{0,p}$ are integer and given by Eqs. (25) and (26). In order to evaluate Eq. (25) in general terms, we assume the relation $|\sum_i (H_{ii} - \bar{E})| \leqslant N_{\mathrm{CI}}|\bar{E}|$ which is valid, e.g. for the case $\bar{E} < H_{ii} \leqslant 0$. In Eq. (26), $\sum_i |u_i|$ is replaced by its upper bound $\sqrt{N_{\mathrm{CI}}}$. The maximum CI dimension $N_{\mathrm{CI,max}}$ is obtained if at least for one of the expressions (32) and (33) equality is achieved. The relation between $N_{\mathrm{CI,max}}$, $\bar{E}$ and ΔE is illustrated in Table 3. If for a certain set of $\bar{E}$ and ΔE values the CI dimension exceeds $N_{\mathrm{CI,max}}$ the calculation has to be performed in a higher precision than the 52 bit mantissa of the IEEE standard or else the accuracy requirement has to be reduced. Usually, in practical applications N_{CI} and $\bar{E}$ are given which allows to check the desired accuracy of the energy.

The importance of I/O bottlenecks in the Davidson scheme has already been discussed at the beginning. By means of the data compression, I/O related to the $\boldsymbol{v}$ and $\boldsymbol{w}$ vectors is reduced at the cost of CPU time. In case that thereby all vectors

Table 3. Maximum CI dimensions $N_{\mathrm{CI,max}}$ in dependence on energies and energy thresholds

$\bar{E}^{a,b}$	ΔE^{c}	$N_{\mathrm{CI,max}}$
-10^2	10^{-11}	382
	10^{-10}	38 208
	10^{-9}	3 820 802
	10^{-8}	382 080 259
-10^3	10^{-10}	597
	10^{-9}	59 700
	10^{-8}	5 970 004
	10^{-7}	597 000 404
-10^4	10^{-9}	233
	10^{-8}	23 320
	10^{-7}	2 332 032
	10^{-6}	233 203 283

[a] energies in a.u.
[b] see Eq. (3)
[c] energy convergence thresholds

can be held in central memory this part of the I/O is, of course, removed completely. In order to illustrate the situation CPU and I/O times have been measured separately. The rates for I/O and compression have been determined on an IBM RS/6000 model 7012-3BT with the COLUMBUS program for the three cases with the largest CI dimensions given in Table 3. We obtain an average rate of I/O (for the $\boldsymbol{v}$ and $\boldsymbol{w}$ vectors) of about 2 MB/s and an average rate for compression of 5–10 MB/s depending on the case. The fact that the CPU efficiency is certain to increase further in relation to disk I/O will favor our compression scheme even more in the future.

4 Conclusions

A general and efficient compression algorithm for the subspace expansion vectors $\boldsymbol{v}_i$ and their matrix products $\boldsymbol{w}_i = \boldsymbol{H}\boldsymbol{v}_i$ within the Davidson diagonalization approach has been presented. In case of SRCI-SD and MRCI-SD calculations, typical saving factors between 4–7 are obtained for an accuracy of 10^{-6} a.u. in the energy. For larger thresholds (e.g. 10^{-4} a.u.), even much larger savings are found. This scheme is expected to be especially profitable with increasing number of small contributions to the CI vector as is the case with larger MR spaces and FCI calculations. The advantage relative to previous compression schemes is the inclusion of both the subspace expansion vectors $\boldsymbol{v}_i$ and all the product vectors $\boldsymbol{H}\boldsymbol{v}_i$ and an analysis of the errors due to compression and automatic control of the Davidson iteration process. The number of necessary iteration steps is not increased by the compression.

Acknowledgements. This work was sponsored by the "Fonds zur Förderung der wissenschaftlichen Forschung", project no. P9032-CHE. We thank Prof. I. Shavitt for his interest in our work.

References

1. Almlöf J, Fægri J Jr, Korsell K (1982) J Comp Chem 3:385
2. Häser M, Ahlrichs R (1989) J Comp Chem 10:104
3. Haase F, Ahlrichs R (1993) J Comp Chem 14:907
4. Olsen J, Jørgensen P, Simons J (1990) Chem Phys Lett 169:463
5. Mitrushenkov AO (1994) Chem Phys Lett 217:559
6. Davidson ER (1975) J Comp Phys 17:84
7. Harrison RJ, Handy NC (1983) Chem Phys Lett 96:386
8. Knowles PJ (1989) Chem Phys Lett 155:513
9. Shepard R (1990) J Comp Chem 11:45
10. McLean AD (1978) in: Proc 1978 IBM Symp on Mathematics and Computation, Vol 1, San Jose, CA, pp 133–142
11. Fülscher MP, Widmark P-O (1993) J Comp Chem 14:8
12. Stradella OG, Corongiu G, Clementi E (1993) J Comp Chem 14:673
13. IEEE Standard 754-1985, IEEE Standard for Binary Floating-Point Arithmetic, IEEE, New York, 1985
14. Shavitt I, private communication
15. Lischka H, Shepard R, Brown F, Shavitt I (1981) Int J Quantum Chem S15:91
16. Shepard R, Shavitt I, Pitzer RM, Comeau DC, Pepper M, Lischka H, Szalay PG, Ahlrichs R, Brown FB, Zhao JG (1988) Int J Quantum Chem S22:149
17. Dunning Jr TH (1989) J Chem Phys 90:1007
18. Schüler M, Kovar T, Lischka H, Shepard R, Harrison RJ (1993) 84:489