Theor Chim Acta (1991) 79: 241–269

Theoretica Chimica Acta

Variational reactive scattering calculations: computational optimization strategies

David W. Schwenke[1], Steven L. Mielke[2], and Donald G. Truhlar[2]

[1] NASA Ames Research Center, MS 230-3, Moffett Field, CA 94035, USA

[2] Department of Chemistry, Chemical Physics Program, and Supercomputer Institute, University of Minnesota, Minneapolis, MN 55455, USA

Received September 1, 1990/Accepted November 13, 1990

Summary. We have developed efficient and accurate techniques for the calculation of quantum mechanical reaction probabilities of atom-diatom exchange reactions in the gas phase, and we have optimized a computer code employing these techniques and applied it sucessfully to several systems. In this paper we consider further strategies for improving the algorithm to allow even more demanding applications. In this context, improvement means that equivalent results can be obtained using fewer computational resources (computer time or storage) or that an equivalent expenditure of resources can yield higher accuracy. The new strategies discussed here lead to improvement in both of these areas. Two areas of special focus in the present paper are (i) the finite difference boundary value method used for calculating distorted wave Green's functions and regular solutions for scattering by the distortion potential and (ii) the choice of the distortion potential itself. Among other results included here is the first application of the outgoing wave or scattered wave variational principle to reactive scattering.

Key words: Chemical reactions – Quantum dynamics – Memory management – Finite difference boundary value method – Outgoing wave variational principle

1. Introduction

A recent exciting advance in the field of chemical reaction dynamics has been the development of efficient and general algorithms [1–13] capable of computing accurate reactive transition probabilities and state-to-state cross sections for atom-diatomic molecule reactive scattering events without any dynamical approximations. Several of our own applications of these algorithms are discussed in a recent overview paper [14] along with a complete (through early 1990) set of references to work by others. These calculations are difficult, and specialized techniques and thoughtful choices among implementation alternatives are required to make the most demanding calculations feasible. Some "tricks of the trade" are very dependent for their efficacy on the particular algorithm employed, but others are more general in their potential usefulness. In some cases the algorithmic enhancements are very subtle, but they can nevertheless have a

significant effect on the accuracy and efficiency of an algorithm. In this paper we will explore some algorithmic considerations that affect the generalized Newton variational principle (GNVP) algorithm that we have presented elsewhere [11, 12], and we discuss various ways to improve the efficiency of such calculations. We also present the computational details necessary to extend the scattered wave variational principle (SWVP) for the T matrix [15–17] and its S matrix analog, the outgoing wave variational principle (OWVP) [16], to reactive scattering. The present paper plus three others [11, 12, 18], taken together along with the description [3] of the method of moments for the amplitude density, include the major computational considerations that have guided the development of our current computer code.

Section 2 outlines the general theoretical and computational considerations which form the framework of our algorithm, Sect. 3 describes the specialized techniques and implementation details not discussed in our previous publications, Sect. 4 contains illustrative results of applications of these techniques to the $D + H_2$ and $F + H_2$ reactions, and Sect. 5 contains our conclusions.

2. Theoretical framework

In our original work on accurate quantum dynamical calculations for three-dimensional chemical reactions we employed the method of moments for the amplitude density (MMAD) [2, 3, 14, 19–22], and in later work we employed the generalized Newton variational principle (GNVP) for both real and complex boundary conditions [10–12, 14, 18, 21–23]. More recently we have demonstrated that the GNVP for complex T matrix boundary conditions is a special case of a more general variational principle called the scattered wave variational principle (SWVP) [16, 17], which was originally stated by Schlessinger [15]. When implementing the more general variational principles, one has the choice of using either the SWVP, which yields directly the transition (T) matrix, or the OWVP, which yields directly the scattering (S) matrix. Because of the linear relation between the T and S matrices, these two variational principles will give identical results for a given choice of basis set and numerical parameters, so the choice is a question of convenience. In this section we give the details required to extend our procedures to the more general basis functions allowed by the SWVP or OWVP. Details not covered are the same as in our previous work [11, 12, 18]; thus the reader is directed to these previous papers for additional information.

In the present discussion we explicitly consider GNVP and OWVP calculations in which we directly form the scattering matrix, although direct evaluation of the reactance or transition matrix proceeds similarly. The scattering matrix elements are labeled S_{nn_0}, where n denotes a full set of atom-diatom quantum numbers, in particular: α_n, which specifies the asymptotic arrangement of the atoms, v_n, the number of nodes in the radial vibrational wave function, j_n, the internal rotational quantum number, l_n, the quantum number for orbital angular momentum of relative translational motion, J, the total angular momentum, M, the projection of the total angular momentum on the laboratory fixed z axis, and P, the parity. Although the arrangement channel quantum number α is included in n, we find it helpful in making the structure of the equations more clear to give at times both n and α, although this is redundant.

For convenience, we include the quantum numbers J, M, and P in n, and we take advantage of the rigorous decoupling of wave functions labeled by different

J, M, and P values to solve for the JMP blocks separately. The results for a given J and P are independent of M so we will label the blocks by their J and P values. (Only one P block occurs for $J = 0$ so this JP block can be labeled by J alone.) Quantities labeled only by the quantum numbers α, v, and j will be called "states", and the specification of both the state and the quantum numbers l, J, M, and P yields a "channel", labeled n.

We will split the Hamiltonian into two parts: a zero order part H_α^D, which contains all of the kinetic energy terms and the diatomic binding potential, but only part of the interaction potential energy between the atom and diatom, and a second part consisting of the rest of the interaction potential. The part of the interaction potential included in H_α^D is called the distortion potential or $V^{D,\alpha}$, and the rest is called the coupling potential V_α^C. Thus we have:

$$H = H_\alpha^D + V_\alpha^C, \tag{1}$$

and we use solutions of $H_\alpha^D - E = 0$ in constructing the scattering matrix for the full Hamiltonian. We define H_α^D such that the scattering problem decouples into blocks diagonal in α and sometimes in other indices as well. Thus:

$$V^{D,\alpha} = \sum_{nn'} \Delta_{nn'}^{\alpha} |\phi_{\alpha n}\rangle V_{nn'}^{\alpha} \langle \phi_{\alpha n'}|, \tag{2}$$

where $\Delta_{nn'}^{\alpha}$, is unity if channels n and n' belong to the same distortion potential block and is zero otherwise, $\phi_{\alpha n}$ is a channel vibrational-rotational-orbital function, and $V_{nn'}^{\alpha}$ is a matrix element of the full interaction potential between the atom and diatom in the $\phi_{\alpha n}$ basis. Then we compute the coupling potential from the equation $V_\alpha^C = H - H_\alpha^D$.

The scattering matrix is written as:

$$S_{nn_0} = \delta_{\alpha_n \alpha_{n_0}} {}^0S_{nn_0} + \mathscr{S}_{nn_0}, \tag{3}$$

where ${}^0S_{nn_0}$ is an element of the scattering matrix due to the Hamiltonian H_α^D and $\mathscr{S}_{nn_0}$ is an element of the correction due to V_α^C, which is given in the GNVP by the expression:

$$\underline{\underline{\mathscr{S}}} = \underline{\underline{\mathscr{S}}}^B + \tilde{\boldsymbol{B}}^T \tilde{\boldsymbol{C}}^{-1} \tilde{\boldsymbol{B}}. \tag{4}$$

The matrices $\underline{\underline{\mathscr{S}}}^B$, $\tilde{\boldsymbol{B}}$, and $\tilde{\boldsymbol{C}}$ are complex. An efficient way to compute them from their real analogs, $\underline{\underline{\mathscr{K}}}^B$, $\boldsymbol{B}$, and $\boldsymbol{C}$, which are appropriate for reactance matrix boundary conditions, is discussed elsewhere [18]; thus we only give the expressions for the real matrices. For these, in the GNVP, we have:

$$\mathscr{K}_{nn_0}^B = \int dR_{\alpha_0} \sum_{n_0'} \mathscr{F}_{nn_0'}(R_{\alpha_0}) \Delta_{n_0' n_0}^{\alpha_0} {}^{(r)}f_{n_0' n_0}^{\alpha_0}(R_{\alpha_0}), \tag{5}$$

$$B_{\beta n_0} = \int dR_{\alpha_0} \sum_{n_0'} \mathscr{G}_{\beta n_0'}(R_{\alpha_0}) \Delta_{n_0' n_0}^{\alpha_0} {}^{(r)}f_{n_0' n_0}^{\alpha_0}(R_{\alpha_0}), \tag{6}$$

$$B_{\beta n_0} = \int dR_\alpha \left[\sum_{n'} \mathscr{F}_{n_0 n'}(R_\alpha) \Delta_{n' n_\beta}^{\alpha} \dot{g}_{n'\beta}^N(R_\alpha) - \mathscr{T}_{n_0 n_\beta}(R_\alpha) t_{m_\beta n_\beta}^{\alpha}(R_\alpha) \right], \tag{7}$$

and

$$C_{\beta\beta_0} = \int dR_{\alpha_0} \mathscr{T}_{\beta n_{\beta_0}}(R_{\alpha_0}) t_{m_{\beta_0} n_{\beta_0}}^{\alpha_0}(R_{\alpha_0}) - \int dR_{\alpha_0} \sum_{n_0'} \mathscr{G}_{\beta n_0'}(R_{\alpha_0}) \Delta_{n_0' n_{\beta_0}}^{\alpha_0} \dot{g}_{n_0' \beta_0}^N(R_{\alpha_0}), \tag{8}$$

where n and n' are channels in arrangement α, n_0 and n_0' are channels in arrangement α_0, β is a composite basis function index corresponding to the union of n_β and the translational basis function label m_β, β is a basis function in arrangement α, β_0 is a basis function in arrangement α_0, R_α is the mass-scaled distance [3] from the atom to center of mass of the diatom in arrangement α, ${}^{(r)}f^{\alpha_0}_{n'n_0}$ is the regular radial function due to scattering by the distortion potential, t^{α}_{mn} is a radial translational basis function, $\dot{g}^{N}_{n\beta}$ is a radial half integrated Green's function (HIGF) [11], and $\mathscr{F}_{nn_0}$, $\mathscr{G}_{\beta n_0}$, $\tilde{\mathscr{F}}_{nn_0}$, and $\mathscr{T}_{\beta n_0}$ are auxiliary integrals defined below. As in previous work, we directly calculate both the radial regular functions and the radial HIGF's by means of the finite difference boundary value method (FDBVM) [11].

We note that $\underline{\underline{\mathscr{K}}}^{B}$ and $\boldsymbol{C}$ are symmetric matrices, and the two expressions for $B_{\beta n_0}$ arise because our code only calculates the auxiliary integrals $\mathscr{F}_{nn_0}$, $\mathscr{G}_{\beta n_0}$, $\tilde{\mathscr{F}}_{nn_0}$, and $\mathscr{T}_{\beta n_0}$ for $\alpha = \alpha_0$ or $\alpha = \alpha_0 + 1$ (modulo 3) [18]. Thus we use Eq. (6) for $\alpha = \alpha_0$ or $\alpha = \alpha_0 + 1$ (modulo 3) and Eq. (7) for $\alpha = \alpha_0 - 1$ (modulo 3).

The auxiliary integrals required above are given by:

$$\mathscr{F}_{nn_0} = \begin{cases} \sum_{n'} \Delta^{\alpha_0}_{nn'} {}^{(r)}f^{\alpha_0}_{n'n}(R_{\alpha_0}) e_{n'n_0}(R_{\alpha_0}), & \alpha = \alpha_0; \\ \int dR_\alpha \sum_{n'} \Delta^{\alpha}_{nn'} {}^{(r)}f^{\alpha}_{n'n}(R_\alpha) W^{\alpha\alpha_0}_{n'n_0}(R_\alpha, R_{\alpha_0}), & \text{otherwise,} \end{cases} \tag{9}$$

$$\mathscr{G}_{\beta n_0} = \begin{cases} \sum_{n'} \Delta^{\alpha_0}_{n_\beta n'} \dot{g}^{N}_{n'\beta}(R_{\alpha_0}) e_{n'n_0}(R_{\alpha_0}), & \alpha = \alpha_0; \\ \int dR_\alpha \sum_{n'} \Delta^{\alpha}_{n_\beta n'} \dot{g}^{N}_{n'\beta}(R_\alpha) W^{\alpha\alpha_0}_{n'n_0}(R_\alpha, R_{\alpha_0}), & \text{otherwise,} \end{cases} \tag{10}$$

$$\tilde{\mathscr{F}}_{nn_0} = \begin{cases} \Delta^{\alpha_0}_{nn_0} {}^{(r)}f^{\alpha_0}_{n_0 n}(R_{\alpha_0}), & \alpha = \alpha_0; \\ \int dR_\alpha \sum_{n'} \Delta^{\alpha}_{nn'} {}^{(r)}f^{\alpha}_{n'n}(R_\alpha) \mathscr{B}^{\alpha\alpha_0}_{n'n_0}(R_\alpha, R_{\alpha_0}), & \text{otherwise,} \end{cases} \tag{11}$$

and

$$\mathscr{T}_{\beta n_0} = \begin{cases} \Delta^{\alpha_0}_{n_\beta n_0} \dot{g}^{N}_{n_0\beta}(R_{\alpha_0}), & \alpha = \alpha_0; \\ \int dR_\alpha \sum_{n'} \Delta^{\alpha}_{n_\beta n'} \dot{g}^{N}_{n'\beta}(R_\alpha) \mathscr{B}^{\alpha\alpha_0}_{n'n_0}(R_\alpha, R_{\alpha_0}), & \text{otherwise.} \end{cases} \tag{12}$$

In Eqs. (9, 10), e_{nn_0} is an intra-arrangement matrix element of $-2\mu V^{C}_{\alpha}/\hbar^2$ in the $\phi_{\alpha n}$ basis [11], and μ is the system reduced mass [13]. The exchange matrix element $W^{\alpha\alpha_0}_{nn_0}$ is the inter-arrangement analog of e_{nn_0}, and it is given in terms of more primitive exchange integrals by:

$$W^{\alpha\alpha_0}_{nn_0}(R_\alpha, R_{\alpha_0}) = \mathscr{C}^{\alpha\alpha_0}_{nn_0}(R_\alpha, R_{\alpha_0}) - \sum_{n_0'} \Delta^{\alpha_0}_{n_0' n_0} \mathscr{B}^{\alpha\alpha_0}_{nn_0'}(R_\alpha, R_{\alpha_0}) U^{\alpha_0}_{n_0' n_0}(R_{\alpha_0}), \tag{13}$$

where $\mathscr{B}^{\alpha\alpha_0}_{nn'}$ is equal to $-2\mu/\hbar^2$ times the overlap between the basis functions of the two arrangements at fixed R_α and R_{α_0}, $\mathscr{C}^{\alpha\alpha_0}_{nn'}$ is equal to $-2\mu/\hbar^2$ times the average of the full potential weighted by the basis functions of the two arrangements [11, 12], and $U^{\alpha}_{nn'}$ is $-2\mu/\hbar^2$ times $V^{\alpha}_{nn'}$. The matrix elements $\mathscr{C}^{\alpha\alpha_0}_{nn_0}$ and $\mathscr{B}^{\alpha\alpha_0}_{nn_0}$ are evaluated computationally as one-dimensional quadratures over the angle between the two vectors along the distances R_α and R_{α_0} [11, 12].

So far the equations we have given are specific to the GNVP. In the SWVP or OWVP we replace some of the HIGF's with more general functions [16, 17]. This requires that we modify some equations. In the more general variational principles, we will consider two types of basis functions for the wave function, or, more precisely, the scattered wave or outgoing wave part of the wave function. The first kind of basis function is taken to be a product of a $\phi_{\alpha n}$ times a radial HIGF. If all basis functions are of this type the OWVP becomes

equivalent to the GNVP for the scattering matrix in which we expand the reactive amplitude density (AD) in terms of $\phi_{\alpha n}$ functions times the square integrable radial translational basis functions which generated the HIGF's. Thus this kind of basis function will be called an AD basis function. The second kind of basis function we will use is a product of a $\phi_{\alpha n}$ and a square integrable energy-independent radial translational function [16, 17] that is used to expand the radial wave function. This kind of basis function will be called a WF (wave function) or energy-independent basis function.

The main modification to the equations in going from the GNVP to the OWVP is the definition of $\boldsymbol{C}$, Eq. (8). All of the other equations remain unchanged provided that we make the substitution:

$$\dot{g}^N_{n\beta} \to \delta_{nn_\beta} t^{\alpha_n}_{m_\beta n_\beta} \tag{14}$$

whenever a WF basis function occurs. The new definition of $\boldsymbol{C}$ requires the additional auxiliary matrix elements:

$$\mathscr{V}_{\beta n_0} = \begin{cases} \sum_{n'_0} \Delta^{\alpha_0}_{n_\beta n'_0} \dot{g}^N_{n'_0\beta}(R_{\alpha_0}) U^{\alpha_0(eff)}_{n'_0 n_0}(R_{\alpha_0}), & \alpha = \alpha_0; \\ \int dR_\alpha \sum_{n'_0} \Delta^{\alpha}_{n_\beta n'_0} \dot{g}^N_{n'_0\beta}(R_\alpha) \mathscr{C}^{\alpha\alpha_0(eff)}_{n'_0 n_0}(R_\alpha, R_{\alpha_0}), & \text{otherwise,} \end{cases} \tag{15}$$

where $U^{\alpha(eff)}_{nn'}$ and $\mathscr{C}^{\alpha\alpha_0(eff)}_{n'n_0}$ differ from $U^{\alpha}_{nn'}$ and $\mathscr{C}^{\alpha\alpha_0}_{n'n_0}$ by the addition of the centrifugal potential to the full potential. This results because we move the centrifugal potential from the first to the second term in Eq. (8). The centrifugal potential is included in the second term because we usually compute these integrals in the body frame [12], hence this contribution is usually nondiagonal. However, it is easy to compute because it is simply proportional to $1/R^2_{\alpha_0}$, which is independent of the integration variable for the $\mathscr{C}^{\alpha\alpha_0}_{nn'}$ and $\mathscr{B}^{\alpha\alpha_0}_{nn'}$ matrix elements; thus $\mathscr{C}^{\alpha\alpha_0(eff)}_{nn'}$ is just a linear combination of the $\mathscr{C}^{\alpha\alpha_0}_{nn'}$ and $\mathscr{B}^{\alpha\alpha_0}_{nn'}$ matrix elements. Then we have the new expression:

$$C_{\beta\beta_0} = \int dR_{\alpha_0} \mathscr{T}_{\beta n_{\beta_0}}(R_{\alpha_0}) \begin{Bmatrix} 1 & \beta_0\text{AD} \\ \dfrac{d^2}{dR^2_{\alpha_0}} + k^2_{\alpha_0 v_\beta j_\beta} & \beta_0\text{WF} \end{Bmatrix} t^{\alpha_0}_{m_{\beta_0} n_{\beta_0}}(R_{\alpha_0})$$
$$- \int dR_{\alpha_0} \sum_{n'_0} \begin{Bmatrix} \mathscr{G}_{\beta n'_0}(R_{\alpha_0}) \Delta^{\alpha_0}_{n'_0 n_{\beta_0}} \dot{g}^N_{n'_0\beta_0}(R_{\alpha_0}) & \beta_0\text{AD} \\ \mathscr{V}_{\beta n'_0}(R_{\alpha_0}) \delta_{n'_0 n_{\beta_0}} t^{\alpha_0}_{m_{\beta_0} n_{\beta_0}}(R_{\alpha_0}) & \beta_0\text{WF} \end{Bmatrix}. \tag{16}$$

These generalizations are sufficient for either the scattered wave variational principle (SWVP) for the T matrix or the outgoing wave variational principle (OWVP) for the S matrix; these formulations differ only in terms of the boundary conditions applied to the radial functions [16]. This gives rise to a phase factor difference for the matrix elements of the complex matrices and the replacement of the scattering matrix due to the distortion Hamiltonian with the T matrix due to the distortion Hamiltonian.

3. Implementation

Although we have always attempted to make our code as efficient as possible, it has undergone continual evolution, and with each incarnation it is faster than before. We have also made several improvements in memory management. In

this section various practical aspects of the algorithm are discussed with the emphasis on our most recent improvements. We start with a discussion of loop optimization, followed by an analysis of the memory requirements. Then the choice of the parameters in the FDBVM is analyzed. Next the construction of the vibrational weights and nodes is reviewed, and finally techniques for mitigating the expense of including high j states are described.

3.1. Loop optimization

The most recent improvements in the efficiency of our code come from two sources: minimizing the amount of inessential work by skipping over negligible contributions to the integrals and improving the efficiency of the essential operations themselves. In the first area we have introduced the five screening parameters ϵ_χ, ϵ_{rad}, ϵ_t, ϵ_W, and $\epsilon_{\mathscr{B}}$. The first of these, ϵ_χ, acts by controlling the number of points included in the inner loop (the angular quadrature) of the exchange integrals and has been discussed previously [12]. It works by rejecting quadrature points which give small contributions because the vibrational wave functions $\chi_{\alpha vj}$ are small—points are rejected if it is estimated that $|\chi_{\alpha vj}| < \epsilon_\chi$ for all vj [12]. The rejecting of angular quadrature points has two possible effects: first, when it eliminates some of the angular quadrature points for that R_α, R_{α_0} pair, it reduces the work required to construct $\underline{\underline{\mathscr{B}}}^{\alpha\alpha_0}$ and $\underline{\underline{\mathscr{C}}}^{\alpha\alpha_0}$ for a given pair of R_α and R_{α_0}; second, when it eliminates all the angular quadrature points for a given R_α, R_{α_0} pair, it reduces the work required for the formation of the auxiliary integrals [Eqs. (9–12, 15)] or even the final $\underline{\underline{\mathscr{K}}}^B$, $\boldsymbol{B}$, and $\boldsymbol{C}$ matrices. Since the auxiliary integrals in Eqs. (10, 12, 15) have a basis function index as well as a channel index, and since the number of basis functions greatly exceeds the number of channels, the latter effect is often much more important.

Because the loop over radial quadratures is not the inner loop, there is no penality in efficiency when radial quadrature points are eliminated; however for the angular quadrature, which is the innermost loop, care must be taken to ensure that the degraded efficiency due to decreased vector lengths does not become significant. For some pairs of R_α and R_{α_0}, ϵ_χ can cause the elimination of all but a few angles in the quadrature sum which then decreases the efficiency of the calculation of the integrand at the quadrature points. To improve on this situation, angles are collected from several different values of R_α, and the integrand is computed in parallel for these angles. The integral is still computed from the integrand as described previously [12].

The other four screening parameters typically have a smaller effect but can be important in certain circumstances. The first of these, ϵ_{rad}, controls the starting points of radial integrals in terms of the magnitudes of the regular radial functions and the radial parts of the HIGF's occurring in the integrals. For each radial function of each distortion block, the maximum distance for which the magnitudes of all functions are less than the fraction ϵ_{rad} of their maximum magnitude is determined. Then for each arrangement, the minimum of these distances is determined, and all radial quadrature points less than this minimum are excluded from the various integrals involved in Eqs. (5–16).

For the AD or WF basis functions, we control the neglect of their contributions by the parameter ϵ_t. This is done by assuming that whenever a basis function is less than ϵ_t of its maximum value, the integrand which contains that function is negligible. This parameter only has a small effect when only AD functions are used and the manipulations described in Sect. 2.3 of Ref. [18] to reduce the work

associated with the HIGF's are not performed. This is because in this case it is only used to eliminate work in the evaluation of the first integral in Eq. (8) and the second term in Eq. (7). However these terms are relatively inexpensive compared to the others in these equations because no matrix multiples are required for the t^{α}_{mn} contribution.

The other two screening parameters, ϵ_W and $\epsilon_{\mathscr{B}}$, are applied to the matrices $\boldsymbol{W}^{\alpha\alpha_0}$ and $\underline{\underline{\mathscr{B}}}^{\alpha\alpha_0}$, respectively. Once these matrices have been computed at a given pair of R_α and R_{α_0}, the maximum magnitude of the matrix elements of $\boldsymbol{W}^{\alpha\alpha_0}$ ($\underline{\underline{\mathscr{B}}}^{\alpha\alpha_0}$) is determined and if this magnitude is less than ϵ_W ($\epsilon_{\mathscr{B}}$), then the contributions to $\underline{\underline{\mathscr{F}}}$ and $\underline{\underline{\mathscr{G}}}$ ($\underline{\underline{\tilde{\mathscr{T}}}}$ and $\underline{\underline{\mathscr{T}}}$) are neglected.

We have observed that in certain circumstances these screening parameters are not sufficient to eliminate all contributions that have a negligible effect. In particular, we have observed in some calculations for the exothermic $F + H_2$ reaction, that if one is only interested in transitions out of this arrangement, the inter-arrangement matrix elements between the two identical H + HF arrangements can be set to zero with only a very small impact on the results. Similarly one can sometimes calculate converged state-to-state reaction probabilities for $D + H_2 \rightarrow HD + H$ without calculating the exchange integrals between the two HD + H arrangements. We do not attempt to eliminate these contributions with a screening parameter, but rather by carefully converging the calculations with respect to the angular quadrature grids since, if these contributions can be neglected, the transition probabilities of interest will be converged with a low or zero value of the number of quadrature points in the unimportant angular exchange integrals.

The improvement in efficiency for essential operations has primarily been obtained in the calculations of the integrals in Eqs. (5–16), and the main tool has been an optimization of the storage of the matrices. This minimizes non-productive work, i.e. data motion. We find it convenient to store the matrix elements of $\underline{\underline{\mathscr{K}}}^B$, $\boldsymbol{B}$, and $\boldsymbol{C}$ and the auxiliary matrices [Eqs. (9–12, 15)] in a rather mixed up order during their computation. Once they are computed, they have to be reordered before transforming to complex boundary conditions and solving Eq. (4). First of all we consider each arrangement pair separately. There will be α_{unique} intra-arrangement pairs and α_{unique} inter-arrangement pairs, where α_{unique} is the number of unique arrangements. These are stored as separate matrices, i.e., all matrix elements for a given arrangement pair are stored in contiguous memory locations. Furthermore, the matrix elements for $\underline{\underline{\mathscr{K}}}^B$, $\boldsymbol{B}$, and $\boldsymbol{C}$ for each distortion potential block pair are also stored as a unit, thus these three matrices come out intermingled. The reason for this is the observation that with the exception of the first terms in Eqs. (8) and (16) and the sign on the second terms, there is no difference in the operations that are performed with the ${}^{(r)}f^{\alpha}_{n'n}$ and the $\dot{g}^{N}_{n\beta}$. To eliminate these small differences, we compute the negative of $\boldsymbol{C}$ and include an extra term in $\boldsymbol{B}$, i.e, for intra-arrangement integrals we compute:

$$B'_{n_0\beta} = B^{T}_{\beta n_0} - \int dR_{\alpha_0} \tilde{\mathscr{T}}_{n_0 n_\beta}(R_{\alpha_0}) t^{\alpha_0}_{m_\beta n_\beta}(R_{\alpha_0}). \tag{17}$$

Then, after the integration is complete, this extra "surface term" is subtracted off and the result transposed. The reason this is done only for the intra-arrangement integrals is because typically more quadrature points contribute to these integrals and the relative efficiency of the operations is not as great because of the presence of the $\Delta^{\alpha_0}_{nn'_0}$ in Eqs. (9–12, 15).

Also for efficiency, the auxiliary integrals $\mathscr{F}_{nn'_0}$ and $\mathscr{G}_{nn'_0}$ ($\mathscr{T}_{nn'_0}$ and $\tilde{\mathscr{T}}_{nn'_0}$) are

intermingled. Grouping the regular radial functions and HIGF's together in this manner minimizes the overheads associated with deciding where the results of a given operation should be stored. The storage scheme also can pay dividends when memory is considered.

3.2. Memory considerations

The calculations we have performed sometimes require large amounts of memory. Our approach was originally developed to take advantage of large-memory supercomputers, and it does indeed do so. Nevertheless there are obvious advantages to using the least amount of memory that allows a given task to be performed efficiently, and we have tried to do this as well. In this section we discuss the current storage requirements of the program and strategies that are being used or can be used to minimize these.

To predict in advance of a calculation the exact amount of memory the code will require is a fairly complicated task, primarily because the code reuses memory locations that hold intermediate results. To facilitate this, all of the large matrices—as well as some other quantities—are stored in one massive array in blank common. Then, as program steps are completed, parts of this array which were used for scratch space can be reused for other purposes in subsequent program steps. This minimizes the total memory required compared to a fixed partitioning of memory into specific arrays. Occasionally, it becomes necessary to move array elements from one part of the massive array to others in order to close up unused spaces—this does not represent useful work, and so such data movement is minimized as much as possible.

In order to estimate the maximum amount of memory the program requires, it is necessary to understand the program structure. In this discussion, we will only explicitly consider the memory requirements for a "large" calculation with "reasonable" choices of the numerical parameters. Thus the equations presented do not yield the exact amount of memory needed, but they are sufficient for strategic considerations.

The first task the code performs is the construction of energy-independent quantities, such as the computation of the diatomic wave functions and the weights and nodes for intra-arrangement vibrational quadratures. Most of these quantities require an insignificant amount of memory with the largest requirement being for the vibrational weights and various pointers. The code allocates a total of $\sum_{\alpha=1}^{\alpha_{\text{unique}}} N_{\alpha\alpha}^{QV}(N_\alpha)^2$ words of storage for the vibrational weights, where $N_{\alpha\alpha}^{QV}$ is the user-specified number of points in the vibrational quadrature, and N_α is the total number of channels in arrangement α. (Actually the number of unique weights in a given arrangement is proportional to the number of states squared, not the number of channels squared, but storing the weights for each channel pair is more convenient for vectorization.) The relative importance of this contribution to total memory depends on the size of $N_{\alpha\alpha}^{QV}$. We make this as small as possible by using an optimized quadrature algorithm [24]. Further details of this technique can be found in Sect. 3.4. For the pointers, a total of $(5\text{MNARG} + 3\alpha_{\text{unique}})\text{MNCHAX} \times \text{MNTBAS}$ words are allocated, where MNARG is a program parameter specifying the maximum number of arrangements, MNCHAX is a program parameter specifying the maximum number of channels per arrangement, and MNTBAS is a program parameter specifying the maximum number of translational basis functions per channel. For convenience later on, call this memory allocation M^{Eind}, i.e.:

$$M^{Eind} = \sum_{\alpha=1}^{\alpha_{\text{unique}}} N_{\alpha\alpha}^{QV}(N_\alpha)^2 + (5\text{MNARG} + 3\alpha_{\text{unique}})\text{MNCHAX} \times \text{MNTBAS}. \quad (18)$$

These weights and most of the pointers must be retained until the computation of the matrix elements is complete, but are not required for the solution of the linear equations, Eq. (4). However if calculations at several energies are to be performed in one run, this information must be kept until the matrix elements for the last energy are computed. In the discussion below we assume that energy-independent quantities of this kind are saved, although we list this amount of storage separately so that one can easily see what the storage requirements would be without saving these quantities. The additional storage that would be required to take advantage of the simple energy dependence of some matrix elements involving energy-independent WF basis functions is not considered in this paper.

Next the code loops over energies and performs the energy-dependent steps, which we divide into three parts. The first is the calculation of the regular radial functions and the radial HIGF's. We will call the program section which controls this operation link 9. The next step is the calculation of matrix elements of Eqs. (5)–(16). We will call the program section which controls this operation link 10. The final expensive step is the evaluation of the right hand side of Eq. (4). We will call the program section performing this operation link 11. Which link of the three will be the most expensive will depend on the parameters used in a given calculation, and furthermore the link which requires the most memory will not necessarily require the most CPU time.

Now consider the memory requirements for link 9. The calculation of the radial functions is done for each arrangement separately. The memory here is divided into two categories. The first is "global", i.e. that which needs to be passed to subsequent links, in contrast to the "local" storage required just for the computation of the global information. The global information consists of the radial functions at the quadrature points and some matrices specifying the boundary conditions, e.g. the reactance matrices going into ${}^0\underline{\underline{\mathscr{K}}}$. This requires a total of:

$$M_\alpha^{rad} = N_\alpha^{QRS}\left(\sum_{\delta(\alpha)} N_{rot,\delta(\alpha)}(N_{rot,\delta(\alpha)} + N_{\text{AD},\delta(\alpha)})\right) + 3\sum_{\delta(\alpha)} N_{rot,\delta(\alpha)} N_{\text{AD},\delta(\alpha)} \quad (19)$$

words, where N_α^{QRS} is the number of radial quadrature points, the sum over $\delta(\alpha)$ is over distortion potential blocks of arrangement α, $N_{rot,\delta(\alpha)}$ is the number of channels in distortion block $\delta(\alpha)$, and $N_{\text{AD},\delta(\alpha)}$ is the total number of AD translational basis functions in distortion block $\delta(\alpha)$. It should be noted that $N_{\text{AD},\delta(\alpha)}$ is the sum over channels in distortion block $\delta(\alpha)$ of the number of AD translational basis functions for each channel. We have made a simplifying assumption in giving this formula. As discussed previously [18], it is possible to relate the radial HIGF's to the regular radial functions and other radial HIGF's in a manner which diminishes the storage requirement of the radial HIGF's, but we assume here that this option is not used.

The local information which is required depends on the maximum number of channels in a distortion block, but not on the number of blocks. Let $N_{rot,\max,\alpha}$ be the maximum value of $N_{rot,\delta(\alpha)}$ for arrangement α. Then the maximum local memory allocation is:

$$M_\alpha^{rad,loc} = N_\alpha^F[(N_{rot,\max,\alpha})^2(2 + 3(N^{FD} - 1)/2) + N_\alpha^{QV} N_\alpha^\lambda] \quad (20)$$

where N_α^F is the number of grid points in the FDBVM, N^{FD} is the number of

points in the finite difference approximation to the second derivative operator, and N_α^λ is the number of Legendre polynomial expansion coefficients for the potential which are saved from block to block. Most of $M_\alpha^{rad,loc}$ is taken up by the matrix required for the FDBVM, and this cannot be diminished except by changing the FDBVM parameters. However the potential expansion coefficients can be recalculated for each distortion potential block—thus N_α^λ can be set to zero. Depending on the size of a particular calculation, the disadvantage of recalculating the expansion coefficients may be counterbalanced by the memory savings, although in general our experience is that the memory savings from this option are rather modest. Then the total memory required to compute the radial regular functions and HIGF's for arrangement α is $M^{Eind} + M_\alpha^{rad,loc} + \sum_{\alpha'=1}^{\alpha} M_{\alpha'}^{rad}$. It should be noted that since the upper limit in the sum in this expression is α, the maximum amount of memory used in this step is dependent on the ordering of the arrangements, i.e. to minimize the storage, $\alpha = 1$ should correspond to the arrangement having the largest value of $M_\alpha^{rad,loc}$.

The next step is the computation of the matrix elements of Eqs. (5)–(16), link 10. After the completion of link 9, the total memory being used is $M^{Eind} + \sum_{\alpha=1}^{\alpha_{\text{unique}}} M_\alpha^{rad}$. The main new memory usage in link 10 step is the storage for the $\underline{\underline{\mathscr{K}}}^B$, $\boldsymbol{B}$, and $\boldsymbol{C}$ matrix elements. These are the matrix elements that must be passed to link 11. If no symmetry is present, when the intra-arrangement matrix elements for arrangement α and the inter-arrangement matrix element for α and $\alpha + 1$ are being formed, this requires:

$$M^{KBC}(\alpha) = \sum_{\alpha'=1}^{\alpha} (M_{\alpha'}^{+})^2 + \sum_{\alpha'=1}^{\alpha} M_{\alpha'}^{+} M_{\alpha'+1}^{+} \tag{21}$$

words of storage, where

$$M_{\alpha'}^{+} = \sum_{\delta(\alpha')} (N_{rot,\delta(\alpha')} + N_{\text{ADWF},\delta(\alpha')}) \tag{22}$$

$N_{\text{ADWF},\delta(\alpha)}$ is the number of AD and WF translational functions in distortion block $\delta(\alpha)$, and $\alpha' + 1$ is considered to be modulo α_{unique}. As with $N_{\text{AD},\delta(\alpha)}$, $N_{\text{ADWF},\delta(\alpha)}$ is the sum over channels in distortion block $\delta(\alpha)$ of the number of AD and WF translational basis functions per channel.

In addition to this storage, we require local storage amounting to

$$M^{loc}(\alpha) = M_\alpha^{+}(f(\text{WF})N_{\alpha+1} + \max_{\delta(\alpha+1)} (N_{rot,\delta(\alpha+1)} + N_{\text{ADWF},\delta(\alpha+1)})) \tag{23}$$

words for the computation of the matrix elements, where $f(\text{WF})$ is three if WF basis functions are used and two if only AD basis functions are used, and $\alpha + 1$ is considered modulo α_{unique}. The part of this memory proportional to the $f(\text{WF})$ factor is used to store the matrices of Eqs. (9–12, 15) at a given value of R_{α_0} for the inter-arrangement integrals and the part proportional to the max term in Eq. (23) is used to temporarily store the results of the matrix multiples of Eqs. (5–7, 16).

If some of the particles are identical, then the magnitude of $M^{KBC}(\alpha)$ can be reduced. This is because some inter-arrangement matrix elements are related and there are some distortion blocks for which there is no intra-arrangement coupling [12]. To facilitate the processing of the intra-arrangement integrals in this case, we have introduced in our code what we call subarrangements. This is useful when the diatom of an arrangement is homonuclear. Then, since the odd and even internal rotational states are not coupled by intra-arrangement matrix elements, it is advantageous to group channels with even and odd j separately. Thus the two subarrangements are made up of the even and odd j channels, respectively. The

intra-arrangement matrix elements for the two subarrangements are stored separately.

For the inter-arrangement integrals with both arrangements indistinguishable, only the lower triangle is stored, and this is done by columns. For the intra-arrangement integrals, which are always symmetric provided the integrals and integrands are evaluated exactly (the composite matrix formed from the intermingled $\underline{\underline{\mathscr{K}}}^B$, $\boldsymbol{B}$, and $\boldsymbol{C}$ matrices is symmetric even though $\boldsymbol{B}$ is rectangular), we compute only the upper triangle, but store this in an array dimensioned for the full matrix. However, parts of the memory allocated for the lower triangle are used temporarily in the construction of the auxiliary intra-arrangement integrals, the upper case in Eqs. (9–12, 15). Thus all of the memory for the lower triangle is not wasted. The motivation for this strategy is a consideration of the data motion and indexing required for other choices.

These savings due to symmetry amount to approximately a factor of two for the $(M_{\alpha'}^{+})^2$ term in Eq. (21) when α' corresponds to a homonuclear diatomic and and a factor of two for the $M_{\alpha'}^{+}M_{\alpha'=1}^{+}$ term in Eq. (21) when arrangements α' and $\alpha'+1$ are identical.

Thus the maximum memory requirement during the computation of the matrix elements in link 10 is $M^{Eind}+\sum_{\alpha=1}^{\alpha_{\text{unique}}} M_{\alpha}^{rad}+\max_{\alpha}[M^{KBC}(\alpha)+M^{loc}(\alpha)]$. As with the radial function calculation, it is advantageous to order the arrangements so that $\alpha=1$ corresponds to the arrangment with the maximum value of $M^{loc}(\alpha)$.

The next step performed in link 10 is the reordering of the matrix elements to be in the form required for Eq. (4). Prior to the reordering we discard the radial functions so that the memory being used is equal to $M^{Eind}+M^{KBC}(\alpha_{\text{unique}})$, where α_{unique} is the number of unique arrangements. When we reorder the matrix elements, only the lower triangle of $\underline{\underline{\mathscr{K}}}^B$ and $\boldsymbol{C}$ will be stored, and this storage will be by symmetry blocks. Thus the total storage required for the reordering is $M^{Eind}+M^{KBC}(\alpha_{\text{unique}})+\sum_{\Gamma=1}^{\Gamma_{\max}} M^{K,B,C}(\Gamma)$ where:

$$M^{K,B,C}(\Gamma)=\tfrac{1}{2}(N_{\underline{\underline{\mathscr{K}}}^B}(\Gamma))^2+N_{\underline{\underline{\mathscr{K}}}^B}(\Gamma)N^{C}(\Gamma)+\tfrac{1}{2}(N^{C}(\Gamma))^2 \tag{24}$$

$N_{\underline{\underline{\mathscr{K}}}^B}(\Gamma)$ is the number of rows and columns in $\underline{\underline{\mathscr{K}}}^B$ for symmetry block Γ, $N^{C}(\Gamma)$ is the number of rows and columns in $\boldsymbol{C}$ for symmetry block Γ, $\Gamma_{\max}$ is the total number of symmetry blocks, and we have made the approximation that the number of elements in the lower triangle are equal to half of the total number of elements in a square matrix. It can be shown that in the absence of symmetry, $\Gamma_{\max}=1$ and:

$$M^{K,B,C}(1)=\tfrac{1}{2}(M_1^{+}+M_2^{+}+M_3^{+})^2 \tag{25}$$

After the reordering step is complete, we discard the composite matrix so that the storage is reduced to $M^{Eind}+\sum_{\Gamma=1}^{\Gamma_{\max}} M^{K,B,C}(\Gamma)$ words, which is passed on to link 11. In summary, the maximum memory for link 10 is the maximum of $M^{Eind}+\sum_{\alpha=1}^{\alpha_{\text{unique}}} M_{\alpha}^{rad}+\max_{\alpha}[M^{KBC}(\alpha)+M^{loc}(\alpha)]$ and $M^{Eind}+M^{KBC}(\alpha_{\text{unique}})+\sum_{\Gamma=1}^{\Gamma_{\max}} M^{K,B,C}(\Gamma)$.

The next large demand on memory will be the evaluation of the right hand side of Eq. (4), which is performed by link 11. This is done for each symmetry block separately [12]. If we proceed in the simplest fashion, for symmetry block Γ, the first step will be the transformation to complex boundary conditions, and this will require an additional $M^{K,B,C}(\Gamma)$ words of storage, and the next step will

be evaluating $\tilde{C}^{-1}\tilde{B}$. Prior to this step we must make a copy of $\tilde{B}$ in order to form the multiplication in Eq. (4).

We have two options for solving the linear equations. Since $\tilde{C}$ is symmetric, and we have stored only the lower triangle, we can use routines which form the UDU^T [25] decomposition of the matrix, or we can store the whole matrix and use the LU [25] decomposition. The LU algorithm requires approximately twice the memory and twice the operation count as the UDU^T algorithm (we use the CSPFA and CSPSL routines from LINPACK [25] modified to store the real and imaginary parts of the complex matrices separately); however there are cases where the LU algorithm is faster because it is more efficient on a vector-pipelined supercomputer. This arises partly from the enhancements available through FORTRAN DO loop unrolling [26, 27, 28]. For the applications in Sect. 4, we used the LU algorithm for $F + H_2$ and the UDU^T algorithm for $D + H_2$.

Putting this together, the total memory required for symmetry block Γ will be either $2M^{K,2B,C}(\Gamma)$ if the UDU^T algorithm is used or $2M^{K,2B,2C}(\Gamma)$ if the LU algorithm is used. Here $M^{K,2B,C}(\Gamma)$ differs from $M^{K,B,C}(\Gamma)$ by doubling the middle term in Eq. (24), and $M^{K,2B,2C}(\Gamma)$ differs from $M^{K,B,C}(\Gamma)$ by doubling the last two terms in Eq. (24). In addition, it is necessary to store the energy-independent quantities (amounting to M^{Eind}), all of the matrix elements for the unprocessed symmetry blocks (amounting to $\sum_{\Gamma'=1}^{\Gamma-1} M^{K,B,C}(\Gamma')$) and the result of Eq. (4) for the completed symmetry blocks (amounting to $\sum_{\Gamma'=\Gamma+1}^{\Gamma_{\max}} (N_{=}^{\mathscr{K}B})^2$). Thus the maximum memory required for link 11 is:

$$M^{11} = M^{Eind} + \max_{\Gamma} \left[\sum_{\Gamma'=1}^{\Gamma-1} M^{K,B,C}(\Gamma') + 2M^{K,2B,\gamma C}(\Gamma) + \sum_{\Gamma'=\Gamma+1}^{\Gamma_{\max}} (N_{=}^{\mathscr{K}B})^2 \right], \quad (26)$$

where γ is 1 if the UDU^T algorithm is used and 2 if the LU algorithm is used.

Again we see that the order of the operations is important: to minimize the storage requirements $\Gamma = \Gamma_{\max}$ should correspond to the symmetry block with the smallest number of basis functions and $\Gamma = 1$ should correspond to the symmetry block with the largest number of basis functions. In practice, this is only an issue for the X_3 case, for which there is one large symmetry block and two small ones [12]. In the AB_2 case, the two symmetry blocks are typically about the same size.

It is possible to reduce this memory requirement. This occurs when translational basis functions in the OWVP do not satisfy the energy-dependent large-R_α boundary conditions but instead decay to zero asymptotically. In this case we can pre-solve for their coefficients in real arithmetic before transforming to complex boundary conditions [6, 18, 29, 30]. All of the WF basis functions of the present study satisfy this criterion. In addition, by forming linear combinations of the radial HIGF's, it is possible to make new basis functions which span the same space as the original HIGF's but having only one function in each channel that is not zero asymptotically [18]; thus in principle most of the AD basis functions can be treated using real arithmetic as well. However for the present we only use this pre-solving technique for the WF basis functions. Thus if the number of WF functions is significantly larger than the number of AD functions, this reduces the memory bottleneck in link 11 by decreasing the middle term in the maximum function of Eq. (26) by about a factor of 2.

Thus the maximum memory used for the calculation will be the maximum of $M^{Eind} + M_{\alpha}^{rad,loc} + \sum_{\alpha'=1}^{\alpha} M_{\alpha'}^{rad}$, $M^{Eind} + \sum_{\alpha=1}^{\alpha_{\text{unique}}} M_{\alpha}^{rad} + \max_{\alpha}[M^{KBC}(\alpha) + M^{loc}(\alpha)]$, $M^{Eind} + M^{KBC}(\alpha_{\text{unique}}) + \sum_{\Gamma=1}^{\Gamma_{\max}} M^{K,B,C}(\Gamma)$ and M^{11}. We can reduce the cost of a

calculation either by reducing this maximum memory or by tailoring the processing to the computer charging algorithm. Typically, if a job is run with static storage (i.e., without using stacks or heaps that are released as procedures are exited), the charge for a calculation depends on the product of the total memory used and the total CPU time. Thus if one step, such as link 9, requires the maximum memory, but uses only a fraction of the total run time, it is advantageous to split the caculation into two parts—the first up to the completion of link 9 followed by the saving of the global link 9 information on disk and the second the reading in of the global link 9 information from disk and then completing links 10 and 11. This may also be accomplished by dynamic memory directives on some computers.

We now consider methods for reducing the maximum memory used. First of all, we consider the possibility of storing some quantities on disk rather than in memory. The options we have implemented are to store the radial functions on disk and to store parts of the matrices on disk.

If we store the radial functions temporarily on disk, then the memory bottleneck in the calculation of the radial functions is reduced from $M^{Eind} + M_\alpha^{rad,loc} + \sum_{\alpha'=1}^{\alpha} M_{\alpha'}^{rad}$ to $M^{Eind} + M_\alpha^{rad,loc} + M_\alpha^{rad}$, since we still require the radial functions for the current arrangement in memory because the order in which they are computed differs from the order in which they will be used in the integrals [12]. This scheme could be further optimized by storing the radial functions on disk before ordering, then ordering them between links 9 and 10. This could reduce the bottleneck further to essentially $M^{Eind} + M_\alpha^{rad,loc}$. Whether or not this bottleneck is significantly smaller depends on the relative sizes of $M^{rad,loc}$ and M_α^{rad}. If there are many AD translational basis functions per channel and $N_{rot,\max,\alpha}$ is not too large, then this will result in a big decrease in memory, while if there are few AD translational basis functions per channel and $N_{rot,\max,\alpha}$ is large, then little change in this memory bottleneck will be observed. Next turning to using these radial functions in the calculation of the matrix elements, we have several choices: we can repetitively read the functions off disk for each quadrature point, we can hold only the radial functions for the current arrangement pair in memory, or we could consider a combination of choices. For example, we could hold the radial functions for arrangement α needed for Eqs. (9–12, 15) in memory, but keep the functions for arrangement α_0, which are needed for Eqs. (5–8, 16) on disk, since in principle all of the α radial functions are accessed for each α_0 quadrature point.

The option we have implemented concerning the storage of the unused matrix elements is as follows: we perform the inter-arrangement and the intra-arrangement integrals separately, and each possible pair of arrangements is done separately from the others. Thus it is necessary to only retain in memory the current inter-arrangement or intra-arrangement integrals that are being computed. Thus the maximum number of integrals required in memory will be the maximum value of $(M_\alpha^+)^2$. This can result in a considerable reduction over $M^{KBC}(\alpha)$ defined in Eq. (21), and furthermore one will incur little input/output penalty for this option, for each matrix element is written once and read once. Then, in the next step of the calculation, the reordering, the maximum memory used will be $M^{Eind} + \sum_{\Gamma=1}^{\Gamma_{\max}} M^{K,B,C}(\Gamma)$ plus the largest single arrangement pair block of integrals.

The options we have just considered can reduce the amount of memory used, but they do not reduce the sum of memory and disk usage. There are several options which can reduce this sum. One is to reduce the size of $N_{rot,\max,\alpha}$ by

considering more general distortion potential blocks than we have done in the past. See the discussion in Sect. 3.5 below as well as the examples of Sect. 4. Another option is to use WF basis functions rather than AD basis functions, which is illustrated in Sect. 4. Yet another idea is to reduce the number of translational basis functions by introducing more general functions which are more optimized for the particular problem [11, 31–34].

In the case where some particles are identical, one could consider performing the calculations for each symmetry block separately. However this would be unlikely to provide significant memory savings and would be much less efficient from a CPU time point of view. This is because there is much information required for all symmetry blocks. For example, the system AB_2 will require the AB + B radial functions and both the AB + B′/AB′ + B inter-arrangement and AB + B intra-arrangement integrals for both symmetry blocks, and in most cases due to the lack of even/old j decoupling for this arrangement, these will be the most expensive quantities to calculate and store. Of course, if one was just interested in computing probabilities for an initial state that was present in only one symmetry block, it would be advantageous to only perform the calculation for the desired symmetry block.

Finally, one could consider not changing the calculation but rather its implementation. In particular, because of the various screening procedures (see Sect. 3.1) we use to eliminate the computation of negligible contributions to the inter-arrangement integrals, we have observed that only a relatively small fraction of the radial quadrature points are used in these integrals. However, almost all of the radial quadrature points are required for the intra-arrangement integrals. Thus the program could be reorganized to compute the intra-arrangement integrals at the same time that the radial functions are computed, or immediately after, and then only the functions at the relatively small number of radial quadrature points required for the inter-arrangement integrals would be saved. This would greatly reduce M_α^{rad}. This is an idea for further work.

Another facet of the memory issue not yet addressed is the reuse of matrix elements at different total energies. If most of the basis functions used are energy-independent WF functions rather than AD function, then many of the matrix elements have a simple dependence on the total energy and could be reused. However this significantly increases the storage: it is now necessary to retain another copy of these matrix elements as well as a new matrix, the overlap between the different functions. This would be approximately $\sum_{\Gamma=1}^{\Gamma^{\max}} (N^C(\Gamma))^2$ words of storage if mostly WF functions were used. These would not have to be kept in memory, but for really demanding calculations where storage is at a premium, it may be more cost effective to recalculate these matrix elements rather than storing them.

3.3. Finite difference parameters

We begin by considering the question of accuracy. If we have numerical instabilities in our algorithm, then the ultimate result of the calculation will be suspect. The principal culprit in this regard is the calculation of the regular radial functions and the radial HIGF's via the FDBVM. The source of this difficulty arises from uneven grid spacing. This occurs because we include Gaussian quadrature points in the grid and because we use a many-point approximation for the second derivative operator, while we can only supply boundary condi-

tions at a single point beyond the end of the grid. Because of the boundary conditions, we are required to change the way we approximate the second derivative operator at the end of the grid. If we are not careful, it is possible that the errors in the radial functions arise mostly from the way we treat the end of the grid.

There are two ways in which we can avoid specifying conditions at more than one point beyond the end of the grid. (We ignore this difficulty at the beginning of the grid by assuming that the radial functions are zero for all distances smaller than the start of the grid.) The scheme which we have used in the past is to reduce the number of points involved in the finite difference approximation so that only one point beyond the end of the grid was required while retaining the restriction that an equal number of points were used before and after the point for which the derivative is evaluated [3]. This means that the last point uses the three-point formula, which gives rise to an error proportional to the stepsize squared, if constant stepsizes are used. In contrast, for most of the grid, we have used the nine-point formula, which gives a much smaller error, so it is necessary to reduce the stepsize at the end of the grid. In practice we do this by augmenting the grid required for the integrals involved in the rest of the calculation with a sequence of grid points with geometrically decreasing stepsizes. Thus there are points in this part of the grid using second derivatives calculated using stepsizes which vary considerably from the first to last finite difference point. This can cause difficulties in accurately computing the finite difference coefficients. This point is discussed further below.

To minimize problems with widely varying stepsizes, the choice of the decrease factor at the end of the grid must be made carefully. Let f^{SD} denote the stepsize decrease factor and N^{SD} be the number of steps this factor is applied. In the past [11], we have recommended a decrease factor of 0.7 over 20 steps. However, subsequent work has indicated that this is not the best choice. Our final conclusions in this matter have not been formulated; however our most recent trials indicate that a decrease factor closer to one will cause less error from this part of the grid. However, if difficulties at the end of the grid are primarily due to the lower-order formulas used at the very end, then it is advantageous instead to consider the alternative option of decreasing the stepsize at the end of the grid relatively rapidly and then taking several fixed stepsizes at the very end of the grid.

An alternative technique is to increase the number of points used in the finite difference formula at the end of the grid by relaxing the requirement that an equal number of points are used before and after the point for which the derivative is required. Then the number of points in the finite difference formula would not need to be reduced nor the stepsize decreased. However a straight-forward application of this idea will lead to difficulties, for this would increase the bandwidth of the linear equations at the very end of the grid. To avoid this problem, it is necessary to reduce the number of points in the finite difference approximation at the end of the grid, but not as severely as previously. If an N^{FD}-point formula is used in the main part of the grid, with N^{FD} an odd integer, then the number of points needs to be reduced to only $(N^{FD}+3)/2$. For example, if a nine-point approximation were used for the majority of the grid, which is the value we typically used in our previous applications, it would be possible to use no less than a six-point approximation at the end. Thus we would expect that the stepsizes would have to be decreased much less at the end of the grid.

An auxiliary question concerns the uniformity of the grid spacings. We require an unevenly spaced grid in order to allow grid points to lie on the Gaussian quadrature points we require for subsequent integrations as well as to provide accuracy at the end of the grid. A uniformly spaced grid will give rise to finite difference coefficients which are smaller in magnitude, and since both positive and negative signs occur, this means that less numerical cancellation will most likely occur, and the solution will be more accurate. In the past, we have tried to make the main part of the grid as uniform as possible by interdispersing extra points among the quadrature points. These points were inserted by dividing the distances between quadrature points by the smallest distance between quadrature points, and truncating the result to obtain an integer. If the result of the truncation is n_j^{in}, then $n_j^{in} - 1$ evenly spaced points are added between quadrature point j and $j + 1$. This is called the "truncate" scheme. Alternatively one can use the nearest integer instead of truncation to obtain n_j^{in}—this defines the "round" scheme. Finally, one could dispense with the extra points, and just use the quadrature points. This yields the "none" scheme. We will denote each grid spacing scheme (GSS) by a number; in particular, GSS = 0 denotes the none scheme, GSS = 1 denotes the truncate scheme and GSS = 2 denotes the round scheme. In previous work we used the truncate scheme and sometimes—in convergence checks—we added one or more *additional* evenly spaced points between each two points obtained from the scheme itself.

In this section, we have outlined several new grid options. It is likely that the optimum choice will depend on the situation. Recall that the ultimate use of the radial functions is in integrals to form matrix elements. The accuracy of the matrix elements will depend on the accuracy in performing the integral and the accuracy in calculating the integrand, i.e., the radial functions. If the matrix element error is dominated by the integration error, the none scheme may be most efficient, while if the error is dominated by the integrand error, the round scheme will probably be most efficient. A compromise choice appears to be the truncate scheme.

A different but related question is the number N^{FD} of points used in the finite difference scheme in the main part of the grid. As N^{FD} increases, the number of grid points required for a given accuracy decreases while the work and memory per grid point increase. Thus in cases where the none scheme might look promising, one could consider the alternative of using the round or truncate scheme but decrease the number of points in the finite difference approximation. Alternatively, in cases where the round scheme appears to be most efficient, it may be advantageous to use the truncate or none scheme but increase N^{FD}.

Another interesting option to consider is to use different orders of Gauss–Legendre quadrature in different sections of the main part of the grid. This is motivated by the observation that the maximum contribution to the inter-arrangement integrals is at short range, and so if the quadrature requirements for these integrals are much more stringent than the intra-arrangement integrals, then one might be able to reduce the work and memory involved in the FDBVM by increasing the grid spacing in the range only important for the intra-arrangement integrals.

We now turn to the calculation of the finite difference coefficients themselves. If these are not computed accurately, then the rest of the calculation is questionable. This is a nontrivial problem because the set of linear equations one solves is of the form of a Vandermonde system [35], which is often ill conditioned. We minimize this problem by means of two steps. First of all, we scale the length

unit so that the smallest stepsize is unity. This causes the magnitude of the matrix elements in the linear equations to be all about the same size. The importance of doing this can be seen by considering the linear system for the three point problem with a grid spacing of h:

$$\begin{pmatrix} 1 & 1 & 1 \\ -h & 0 & h \\ h^2 & 0 & h^2 \end{pmatrix} \begin{pmatrix} c_{-1} \\ c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{27}$$

Here we are approximating $d^2F/dR^2|_{R_0}$ by $\sum_{i=-1}^{1} c_i F(R_0 + ih)$. If h is very small, then the numerical difficulties can be encountered in solving Eq. (27); however by scaling the length unit we obtain the alternative set of equations:

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{c}_{-1} \\ \tilde{c}_0 \\ \tilde{c}_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tag{28}$$

where $c_i = \tilde{c}_i/h^2$. Although the two sets of linear equations are formally equivalent, we prefer to use Eq. (28). Secondly we use a specialized algorithm to solve the linear equations which is less susceptible to rounding errors [35].

Once we have computed the finite difference coefficients, our vigilance should not end. Numerical difficulties in computing these coefficients probably indicate that they vary widely in magnitude, and thus the FDBVM equations can also be susceptible to numerical problems. In an attempt to improve the numerical properties of the system of linear equations we solve to obtain the radial functions, we have introduced the option to perform a row and column scaling of the coefficient matrix [25]. The scaling factors are chosen as the inverses of the square roots of the absolute values of the diagonal elements, thus the scaled matrix contains only ± 1 on its diagonal. We also can apply the same technique to the other linear equation steps in our algorithm, in particular we have introduced the option of scaling the linear system for the determination of the amplitude density or wave function expansion coefficients, which are given by $\tilde{\boldsymbol{C}}^{-1}\tilde{\boldsymbol{B}}$ in Eq. (4). Here the sources of numerical instability are not associated with the finite difference grid, but rather the basis set used in the wave function expansion. It should be noted that this scaling procedure adds negligible additional cost to the solution of the linear equations. For the applications in Sect. 4, we utilized the row and column scaling of $\tilde{\boldsymbol{C}}^{-1}\tilde{\boldsymbol{B}}$ for $F + H_2$, but not for $D + H_2$. In neither case were the FDBVM equations scaled.

Finally we note that one should optimize the spatial extent of the finite difference grid, i.e., the locations of the first and last grid points. A word of warning is in order concerning the cutting back on the maximum R_α included in the FDBVM grid. This may yield converged transition probabilities but not converged phases, which could cause significant inaccuracies when computing differential cross sections.

3.4. Vibrational weights and nodes

An important detail of our method which makes it efficient is the technique we use to evaluate integrals over the vibrational coordinate. We use an optimized quadrature algorithm based upon integration nodes computed from the orthogonal polynomials generated by the absolute value squared of the ground state

vibrational wave function [24]. Although we have not encountered difficulties using the algorithm of that reference to compute the quadrature weights and nodes, we have introduced an improved algorithm which is more accurate. The new algorithm is to compute the quadrature nodes from numerically computed moments of the form $(r - r_e)^n$ with the algorithm of Sack and Donovan [36]. The quadrature weights are then computed from the solution of the linear equations relating the moment integrals to the nodes using the transpose Vandermonde algorithm of Ref. [35]. In the absence of round-off error, the original and improved algorithms would give identical results.

3.5. Treatment of high j states

To obtain well converged results it is often necessary to include many high j states that when considered individually provide only small contributions. For calculations at high total angular momentum this is an egregious problem as it is these very states that contribute the largest numbers of channels. Thus the desire for a small increase in accuracy may incur a vast increase in the computational effort. Two strategies have been implemented that partially alleviate this condition.

In the first strategy we consider optimizing the translational basis for each individual channel. In most prior calculations our basis sets were restricted within each arrangement to direct products of radial translational basis functions and a vibrational-rotational-orbital basis. We now permit each channel to have distinct translational basis parameters and allow variable spacing of the distributed Gaussians even with a given channel. High vj states typically require fewer less closely spaced Gaussians and this provides savings during many stages of the calculation. Some of the most relevant options have already been considered in the recent study by Halvick et al. [37] and need not be considered further here. A general test problem that utilizes this additional freedom is given in the next section.

The second strategy involves the use of more general distortion potentials. In prior applications we have usually used a fully rotational coupled distortion potential where all channels with the same values of v and α are coupled. For a rotationally coupled distortion potential with maximum rotational quantum number $j_{\max}$, where $j_{\max} < J$, the number of channels is $O(j_{\max}^2)$ and thus the memory requirements scale as $j_{\max}^4$, and the computational effort scales as $j_{\max}^6$. For many calculations this produces a memory bottleneck at the evaluation of the regular radial functions and the radial parts of the HIGF's for large total angular momenta. The only alternative originally considered was the single-state distortion potential, where only channels with the same values of v, j, and α are coupled. Because we perform the calculation of the matrix elements in the body frame [12], we cannot uncouple the different values of l that arise when J is greater than zero. Now we have introduced several compromise distortion potentials.

The specification of the new distortion potentials starts with the rotationally coupled distortion potential and then decouples selected vj states. There are several rules to select these states. One rule is to decouple the closed channel states from the open channel states for particular v, and another is to base the decoupling on the value of j. Finally, one can decouple all states in distortion blocks with particular values of v. We find it convenient to introduce the parameter $j_{\alpha v}^d$ to specify this decoupling. The meaning of this parameter is that all αvj states with $j \geqslant j_{\alpha v}^d$ are decoupled. It is clear that significant savings in both

memory and computational effort can be gained by splitting up large distortion blocks. This benefit will occur both in the calculation of the radial functions and in the construction of the matrix elements.

These gains are not achieved without a certain price. First of all, we had to revectorize cricical portions of link 10 to make them more efficient (and minimize data motion) for the case of single-channel distortion blocks. Second, any coupling that is no longer included in the distortion potential must be handled by the variational principle, and thus more translational basis functions may be required. When employed in tandem, a carefully chosen combination of the two strategies discussed in this subsection can yield substantial savings. Examples for state-of-the-art problems are given in the next section.

4. Applications

In this section we present applications of the new techniques of the previous section to two chemical reactions. First we will consider the system $D + H_2$ using the accurate double-many-body-expansion (DMBE) potential energy surface [38], and then we will demonstrate the utility of the new methods for the exothermic reaction $F + H_2$ using the realistic potential energy surface no. 5A [39, 40]. The $D + H_2$ calculations reported here illustrate the use of WF basis functions and more efficient quadrature parameters, and the $F + H_2$ calculations illustrate the savings that may be achieved by simultaneous use of several of the new techniques described above to maximize the efficiency of the calculations.

4.1. $D + H_2$ *with the* OWVP

In this section we report the first converged results using the OWVP for a reactive system. Convergence studies of calculations employing Gaussian AD basis sets in the GNVP were first reported for the $D + H_2$ reaction in Ref. [12], and it is convenient to consider the same example here, namely, the reaction at total energy $E = 0.98337$ eV with total angular momentum $J = 0$. All parameters not listed here are the same as in the previous study. In these calculations, we monitored two kinds of probabilities. The first are state-selected probabilities, defined by:

$$P^{JP}_{vj} = \sum_{\alpha'=2,3} \sum_{v'} \sum_{j'} P^{JP}_{1vj,\alpha'v'j'} \tag{29}$$

where $P^{JP}_{\alpha vj,\alpha'v'j'}$ is the distinguishable atom transition probability from state αvj to state $\alpha'v'j'$, and $\alpha = 1$ is the initial arrangement, $D + H_2$. The second kind of probabilities that we monitored for these calculations are state-to-state reactive transition probabilities out of the ground state, defined by:

$$P^{JP}_{00,v'j'} = \sum_{\alpha'=2,3} P^{JP}_{100,\alpha'v'j'} \tag{30}$$

We do not claim that our final basis set parameters are fully optimum. However, a large number of different possible choices were considered, and the present results appear to be a good starting point for further optimization. The parameters for our two best calculations using WF basis functions are in Table 1, where N denotes the total number of channels, M(AD) denotes the number of

Table 1. Translational basis function parameters for $D + H_2$ calculations with $J = 0$.[a]

Run	A	B	C
	$D + H_2$		
m_{max}(AD) [b]	3	3	5
R_1^G(AD) [c]	2.8	2.8	2.8
Δ(AD) [d]	0.4	0.4	0.4
m_{max}(WF) [e]	3	2	0
R_1^G(WF) [e]	3.6	3.7	—
Δ(WF)	0.45	0.5	—
	H + DH		
m_{max}(AD)	3	3	5
R_1^G(AD)	2.5	2.5	2.5
Δ(AD)	0.4	0.4	0.4
m_{max}(WF)	3	2	0
R_1^G(WF)	3.3	3.5	—
Δ(WF)	0.45	0.5	—
	all three arrangements		
N	225	225	225
M(AD)	675	675	1125
M(WF)	675	450	0
M	1350	1125	1125

[a] In all cases the overlap parameter c is 1.4
[b] Number of functions per channel
[c] Center of first basis function (mass-scaled bohr)
[d] Spacings between functions (mass-scaled bohr)
[e] Same as b, c, and d but for WF functions

AD basis functions, M(WF) denotes the number of WF basis functions, and M denotes the total number of basis functions. It should be noted that N, M(AD), M(WF), and M are summed over both permutation symmetries even though in the actual calculations permutational symmetry was used to block diagonalize Eq. (4). This is true both for the $D + H_2$ and the $F + H_2$ calculations reported in this paper. The probabilities are in Tables 2 and 3. These OWVP calculations use three energy-dependent AD basis functions and two or three energy-independent WF basis functions per channel. The convergence of these calculations compares favorably to the previous results for seven energy-independent basis functions [12]. Our latest (better optimized) results with five energy-dependent basis functions per channel are also presented for comparison.

The conclusion we draw from these calculations is that it is possible to obtain accurate results using WF basis functions without increasing the overall basis set requirements, although it appears for the present case that at least three AD basis functions must be used per channel.

4.2. $D + H_2$ with an optimized strategy for distortion potentials and optimized screening parameters

In this section we present results that illustrate the effects of decoupling higher j states in the distortion potential. We will consider the $D + H_2$ reaction again, but

Table 2. Reaction probabilities, $P_{vj}^{J=0}$, for $D + H_2$

v	j	converged [a]	Run A	Run B	Run C
0	0	6.14(−1) [b]	6.15(−1)	6.14(−1)	6.15(−1)
0	1	9.05(−1)	9.06(−1)	9.06(−1)	9.07(−1)
0	2	5.57(−1)	5.56(−1)	5.58(−1)	5.57(−1)
0	3	6.77(−1)	6.77(−1)	6,78(−1)	6.77(−1)
0	4	7.08(−1)	7.09(−1)	7.08(−1)	7.09(−1)
0	5	4.62(−1)	4.63(−1)	4.63(−1)	4.63(−1)
0	6	2.21(−1)	2.21(−1)	2.21(−1)	2.21(−1)
0	7	5.39(−2)	5.38(−2)	5.36(−2)	5.39(−2)
1	0	2.96(−1)	2.96(−1)	2.97(−1)	2.96(−1)
1	1	4.08(−1)	4.07(−1)	4.09(−1)	4.08(−1)
1	2	1.20(−1)	1.20(−1)	1.20(−1)	1.20(−1)
1	3	8.65(−3)	8.69(−3)	8.53(−3)	8.74(−3)

[a] From calculation with 16 AD functions of Ref. [12]
[b] $6.14(-1) \equiv 6.14 \times 10^{-1}$

Table 3. Reactive transition probabilities, $P_{00,v'j'}^{J=0}$, for $D + H_2$

v'	j'	converged [a]	Run A	Run B	Run C
0	0	7.73(−2) [b]	7.74(−2)	7.73(−2)	7.74(−2)
0	1	1.65(−1)	1.66(−1)	1.65(−1)	1.66(−1)
0	2	1.35(−1)	1.35(−1)	1.34(−1)	1.35(−1)
0	3	5.44(−2)	5.44(−2)	5.40(−2)	5.45(−2)
0	4	7.48(−3)	7.42(−3)	7.43(−3)	7.48(−3)
0	5	8.20(−3)	8.22(−3)	8.36(−3)	8.20(−3)
0	6	2.04(−2)	2.05(−2)	2.06(−2)	2.05(−2)
0	7	1.73(−2)	1.73(−2)	1.74(−2)	1.73(−2)
0	8	6.83(−3)	6.82(−3)	6.95(−3)	6.83(−3)
1	0	2.37(−2)	2.37(−2)	2.38(−2)	2.37(−2)
1	1	4.72(−2)	4.73(−2)	4.74(−2)	4.72(−2)
1	2	3.43(−2)	3.43(−2)	3.43(−2)	3.43(−2)
1	3	1.31(−2)	1.32(−2)	1.31(−2)	1.32(−2)
1	4	2.39(−3)	2.40(−3)	2.38(−3)	2.40(−3)
1	5	8.46(−5)	8.46(−5)	8.48(−5)	8.34(−5)

[a] From calculation of Ref. [12] with 16 AD functions
[b] $7.75(-2) \equiv 7.73 \times 10^{-2}$

now at $E = 0.93$ eV and with $J = 10$. In this case the reactive probabilities we tabulate are the state-to-state reactive transition probabilities out of the $v = 0$, $j = 4$ state with even parity, defined by:

$$P_{04,v'j'}^{J=10,P=+} = \sum_{\alpha'=2,3} P_{104,\alpha'v'j'}^{J=10,P=+} \tag{31}$$

where $P_{104,\alpha'v'j'}^{J=10,P=+}$ is summed over l' and averaged over the five values of initial l that occur for this j in this JP block.

Calculations were performed with the two parameter sets listed in Table 4. The definitions of parameters not given in this paper are the same as in Ref. [41]. In these calculations, we use the GNVP for the scattering matrix, and further-

Table 4. Parameter sets for $D + H_2$ calculations with $J = 10$

	Set D		Set E	
parameter	$D + H_2$	$H + DH$	$D + H_2$	$H + DH$
$j_{max}(v = 0)$ [a]	12	14	13	15
$j_{max}(v = 1)$	12	14	13	15
$j_{max}(v = 2)$	10	12	11	13
$j_{max}(v = 3)$	9	9	10	10
$j_{max}(v = 4)$	—	—	6	6
$m_{max}(AD)$	5	5	6	6
$R^G_{\alpha,1}(a_0)$	2.8	2.5	2.7	2.4
$\Delta(a_0)$	0.40	0.40	0.37	0.37
c [b]	1.4	1.4	1.3	1.3
$N(HO)$ [c]	50	50	60	60
$N^{QA}_{\alpha\alpha}$ [d]	30	30	40	40
$N^{QV}_{\alpha\alpha}$ [e]	12	12	13	13
$N(F)$ [f]	107	107	131	131
N^{FD}	13	13	13	13
$R^F_{\alpha,1}$ [g]	1.1	0.8	0.9	0.6
$R^F_{\alpha,N(F)}$ [g]	11.0	11.0	12.0	12.0
GSS	0	0	0	0
f^{SD}	0.9	0.9	0.9	0.9
N^{SD}	30	30	35	35
N^{QS} [h]	11	11	12	12
N^{QGL} [h]	7	7	8	8
N^{QA}_{12}, N^{QA}_{23} [i]	50, 0		60, 60	
$-\log_{10} \epsilon_\chi$	4		7	
$-\log_{10} \epsilon_{rad}$	3		7	
$-\log_{10} \epsilon_t$	4		7	
$-\log_{10} \epsilon_W$	4		7	
$-\log_{10} \epsilon_{\mathscr{B}}$	3		7	
N	1023		1239	
$M(AD)$	5115		7434	
$M(WF)$	0		0	
M	5115		7434	

[a] $j_{max}(v)$ is the maximum value of j included for level v

[b] c is the Gaussian overlap parameter of Hamilton and Light [44]

[c] $N(HO)$ is the number of harmonic oscillator functions used to expand the asymptotic vibrational eigenstates

[d] $N^{QA}_{\alpha\alpha}$ is the number of abscissae in the Gauss–Legendre quadrature used for intra-arrangement angular integrals

[e] $N^{QV}_{\alpha\alpha}$ is the number of abscissae in the optimized quadrature [24] used for intra-arrangement integrals over the vibrational coordinate

[f] $N(F)$ is the number of grid points in the FDBVM calculations

[g] $R^F_{\alpha,1}$ and $R^F_{\alpha,N(F)}$ are the first and last grid points in the FDBVM calculations

[h] The integrations over the radial translational variables, in both intra-arrangement and inter-arrangement integrals, are carried out by N^{QS} repetitions of an N^{QGL}-point Gauss–Legendre quadrature rule

[i] $N^{QA}_{\alpha\alpha'}$ is the number of abscissae in the Gauss–Legendre quadratures used for angular integrations in the inter-arrangement exchange integrals between arrangements α and α'

Table 5. Reactive transition probabilities, $P_{04,v'j'}^{J=10,P=+}$, for $D + H_2$

		Run					
		1	2	3	4	5	6
v'	j'	set D	set D	Set D	set D	set D	set E
		$j_{av}^d = 1$	$j_{av}^d = 7$	$j_{av}^d = 9$	$j_{av}^d = 11$	$j_{av}^d = \infty$	$j_{av}^d = \infty$
0	0	1.032(−2) [a]	9.996(−3)	9.992(−3)	9.993(−3)	9.994(−3)	9.994(−3)
0	1	2.475(−2)	2.395(−2)	2.397(−2)	2.397(−2)	2.397(−2)	2.396(−2)
0	2	2.717(−2)	2.633(−2)	2.634(−2)	2.634(−2)	2.634(−2)	2.633(−2)
0	3	2.275(−2)	2.201(−2)	2.202(−2)	2.202(−2)	2.202(−2)	2.202(−2)
0	4	1.893(−2)	1.830(−2)	1.830(−2)	1.830(−2)	1.830(−2)	1.831(−2)
0	5	1.782(−2)	1.734(−2)	1.729(−2)	1.729(−2)	1.729(−2)	1.731(−2)
0	6	1.674(−2)	1.639(−2)	1.631(−2)	1.631(−2)	1.631(−2)	1.633(−2)
0	7	1.302(−2)	1.287(−2)	1.272(−2)	1.266(−2)	1.266(−2)	1.268(−2)
0	8	7.272(−3)	7.186(−3)	7.163(−3)	7.095(−3)	7.100(−3)	7.111(−3)
0	9	2.338(−3)	2.315(−3)	2.344(−3)	2.309(−3)	2.306(−3)	2.310(−3)
0	10	1.892(−4)	1.874(−4)	1.930(−4)	1.943(−4)	1.936(−4)	1.951(−4)
1	0	5.485(−4)	5.262(−4)	5.264(−4)	5.261(−4)	5.261(−4)	5.303(−4)
1	1	1.177(−3)	1.134(−3)	1.135(−3)	1.135(−3)	1.135(−3)	1.144(−3)
1	2	1.014(−3)	9.750(−4)	9.758(−4)	9.751(−4)	9.751(−4)	9.834(−4)
1	3	5.153(−4)	5.009(−4)	5,007(−4)	5.001(−4)	5.001(−4)	5.044(−4)
1	4	1.586(−4)	1.504(−4)	1.523(−4)	1.520(−4)	1.521(−4)	1.539(−4)
1	5	2.042(−5)	1.887(−5)	1.899(−5)	1.899(−5)	1.899(−5)	1.932(−5)

[a] $1.032(-2) \equiv 1.032 \times 10^{-2}$

more we use the new FDBVM option to reduce the finite difference approximation at the end of the grid to only $(N^{FD} + 3)/2$ points. Set D was used for five runs that differed only in the decoupling parameter j_{av}^d. For this test we take j_{av}^d to be independent of v. Values of 1, 7, 9 and 11 were considered in addition to full rotational coupling. We will denote full rotational coupling by $j_{av}^d = \infty$. Set E was run only using full rotational coupling, and it serves as a convergence check for the other parameters.

Table 5 lists selected transition probabilities for the six calculations, and Table 6 gives the time and memory requirements for these cases. In these

Table 6. Time and memory requirements for $D + H_2$ calculations with $J = 10$. Times are CPU times relative to that for link 9, run 5. Memory is in millions of words

			link 9		link 10		link 11
Run	set	j_{av}^d	time	memory	time	memory	time
1	D	1	0.037	5.6	0.077	25.3	0.470
2	D	7	0.072	7.6	0.092	25.4	0.462
3	D	9	0.163	12.0	0.117	25.5	0.472
4	D	11	0.396	20.5	0.152	30.0	0.457
5	D	∞	1.000	45.9	0.233	NL [a]	0.465
6	E	∞	1.91	74.0	1.97	74.8	1.254

[a] NL denotes less than needed in previous link, thus not limiting

calculations, all quantities were kept in memory, i.e., the options to write quantities to disk were not used. In all cases, the maximum memory used occurred prior to link 11. The variation in the time for link 11 between runs 1–5 is a measure of the uncertainties of our timings as all these times should be the same. Run 1 shows that a single-state distortion potential is not adequate. Run 3, however, shows that coupling only the first nine states ($j = 0$–8) gives well converged results. This run requires only one sixth as much computing time to calculate the radial functions in link 9 as does run 5 with full rotational coupling. Appreciable savings are also observed during the integral evaluations of link 10. We see that it is efficient to couple together only the most important j states at the distortion potential level and to treat the less important states as uncoupled at that level.

Careful use of screening parameters is also very helpful in reducing the cost of the integral evaluations. Consider, as an example, the effect of the screening parameters for runs using parameter set D of Tables 5 and 6. There are $(77)^2$ different $R_\alpha, R_{\alpha'}$ pairs that must be considered. Table 7 shows how many are eliminated in the integrations of Eqs. (9–12) by each of the screening parameters. Results are listed for parameter set D for both $J = 0$ and $J = 10$. The parameters ϵ_W and $\epsilon_{\mathscr{B}}$ generally have only a small effect, and ϵ_χ is responsible for most of the savings. The parameter ϵ_{rad} has a limited effect for low J but eventually removes all the remaining points in the limit of large J.

4.3. $F + H_2$

In the previous subsection we have illustrated the OWVP by applying it to a nearly thermoneutral reaction. In this section we will apply it to a more difficult exothermic reaction, and we will also illustrate the savings which can be achieved with optimized distortion potentials and refined finite difference grids. Simultaneous use of these methods, together with a channel-dependent translational basis and more efficient use of screening parameters yields considerable savings. We consider the reaction of F with H_2 at a total energy of 0.34 eV with $J = 0$ on the 5A surface of Refs. [39, 40]. The parameters of Yu et al. [42, 43] are taken as a starting point, although these parameters were intentionally chosen as "safe" rather than fully optimized. The following five modifications are made to

Table 7. Effects of screening parameters on GNVP calculations for $D + H_2$ with parameter set D. Quantity tabulated is number of pairs of radial quadrature points

	$J = 0$	$J = 10$
total	5929	5929
eliminated by ϵ_χ alone	5369	5369
eliminated by ϵ_{rad} alone	819	1169
eliminated by either or both of these	5530	5588
remaining after above	399	341
of those remaining:		
eliminated by ϵ_W	103	98
eliminated by $\epsilon_{\mathscr{B}}$	58	57

the parameters of Yu et al.:

(i) All screening parameters are reduced. In particular, ϵ_χ, ϵ_{rad}, ϵ_t, ϵ_W, and $\epsilon_{\mathscr{B}}$ are reduced from 10^{-12}, 10^{-50}, 10^{-20}, 10^{-20}, and 10^{-20}, respectively, to 10^{-6}, 10^{-3}, 10^{-6}, 10^{-4}, and 10^{-3}.

(ii) The finite difference grid is reduced in both range and number of grid points, and the none scheme with order 13 replaced the truncate scheme with order 9. For the $F + H_2$ arrangement, the grid extends from 2.3 to 13 mass-scaled bohr, and 10 repetitions of 12-point Gauss–Legendre quadrature are used in the radial

Table 8. Parameter sets for $F + H_2$ calculations

	new		Refs. [42, 43]	
parameter	$F + H_2$	$H + FH$	$F + H_2$	$H + FH$
$j_{\max}(v=0)$	12	22	12	22
$j_{\max}(v=1)$	10	18	10	18
$j_{\max}(v=2)$	8	15	8	15
$j_{\max}(v=3)$	6	11	6	11
$j_{\max}(v=4)$	4	7	4	7
$j_{\max}(v=5)$	—	4	—	4
$j_{\max}(v=6)$	—	3	—	3
$j^d_{\alpha 0}, j^d_{\alpha 1}$	∞	1	∞	∞
$j^d_{\alpha 2}, j^d_{\alpha 3}$	1	13	∞	∞
$j^d_{\alpha 4}$	1	1	∞	∞
$j^d_{\alpha 5}, j^d_{\alpha 6}$	—	1	∞	∞
$m_{\max}$(AD)	Table 9	Table 9	16	28
$R^G_{\alpha,1}(a_0)$	Table 9	Table 9	2.0	2.0
$\Delta(a_0)$	Table 9	Table 9	0.3	0.2
c	1.4	1.4	1.4	1.4
N(HO)	45	60	45	60
$N^{QA}_{\alpha\alpha}$	40	50	40	50
$N^{QV}_{\alpha\alpha}$	10	14	10	14
$N(F)$	155	155	1172	1148
N^{FD}	13	13	9	9
$R^F_{\alpha,1}$	2.3	1.3	0.3611	0.3611
$R^F_{a,N(F)}$	13.0	10.0	19.87215	19.87215
GSS	0	0	1	1
f^{SD}	0.9	0.9	0.7	0.7
N^{SD}	35	35	20	20
N^{QS}	10	10	16	24
N^{QGL}	12	12	16	16
N^{QA}_{12}, N^{QA}_{23}	80,0		80,80	
$-\log_{10}\epsilon_\chi$	6		12	
$-\log_{10}\epsilon_{\text{rad}}$	3		50	
$-\log_{10}\epsilon_t$	6		20	
$-\log_{10}\epsilon_W$	4		20	
$-\log_{10}\epsilon_{\mathscr{B}}$	3		20	
N	219		219	
N(AD)	1761		5592	
M(WF)	981		0	
N	2742		5592	

quadratures. For the H + HF arrangement, the grid extends from 1.3 to 10 mass-scaled bohr, and 10 repetitions of 12-point Gauss–Legendre quadrature are used in the radial quadrature. For both arrangements, a stepsize decrease factor of 0.9 was used for the last 35 steps, and the finite difference formula used 8 points for the end of the grid.

(iii) For the H + HF arrangement, single-state distortion blocks are used for all states having $j \geqslant 13$ and $v = 2$ or 3 and for all states with vibrational quantum number equal to 0, 1, 4, 5, or 6. For the $F + H_2$ arrangement single-state distortion blocks were used for all states having $v = 2$, 3, or 4.

(iv) The translational basis is reduced to an average of 8.4 Gaussians for the $F + H_2$ arrangement and 13.6 Gaussians for the H + HF arrangement. About one third of these are now independent of energy.

(v) The parameter N_{23}^{QA} was set to 0, i.e., the inter-arrangement matrix elements between the two identical H + HF arrangements were neglected. (In the present calculations we still use 80 for N_{12}^{QA}, and N_{13}^{QA} is not needed because we use symmetry.)

These modifications decrease the memory requirements by about a factor of five when all quantities are kept in memory, and they provide a speed-up factor in excess of twenty without appreciable loss of accuracy. Table 8 gives most of the parameters for both the new and the previous [42, 43]

Table 9. Modified translational basis set parameters for the $F + H_2$ reaction

R_1 [a]	Δ [b]	(m_{max}) [c]	Type	n_{start} [d]	n_{end} [e]
			$F + H_2$		
3.0	0.30	7	AD	1	6
3.1	0.34	6	AD	7	18
3.3	0.35	5	AD	19	45
4.8	0.30	4	WF	1	18
4.8	0.30	3	WF	19	33
4.8	0.30	1	WF	34	45
			H + HF		
2.0	0.40	7	AD	1	14
2.6	0.50	4	AD	15	23
2.0	0.20	12	AD	24	35
2.2	0.20	11	AD	36	70
2.2	0.40	6	AD	71	78
2.4	0.40	5	AD	79	87
4.2	0.26	5	WF	1	78
4.0	0.26	4	WF	79	87

[a] Center of first Gaussian in series (mass-scaled bohr)
[b] Spacing between Gaussian in series (mass-scaled bohr)
[c] Number of Gaussians in series
[d] First channel to which this series of basis functions applies. For the $F + H_2$ arrangement the order of the channels is all even j before odd j, then $[v, j] = [0, 0], [0, 2], \ldots, [0, j_{max}(0)], [1, 0], \ldots$, etc. For the H + HF arrangement, the order of the channels is $[v, j] = [0, 0], [0, 1], \ldots, [0, j_{max}(0)], [1, 0], \ldots$, etc
[e] Last channel to which this series of basis functions applies

Table 10. Reactive transition probabilities, $P_{0j,v'j'}$, for $F + H_2$ on surface 5A

j	v'	j'	Yu et al. [a]	Present
0	2	0	2.02(−3) [b]	2.01(−3)
0	2	1	5.53(−3)	5.54(−3)
0	2	2	7.85(−3)	7.88(−3)
0	2	3	9.21(−3)	9.23(−3)
0	2	4	1.01(−2)	1.02(−2)
0	2	5	1.10(−2)	1.10(−2)
0	2	6	1.14(−2)	1.15(−2)
0	2	7	1.10(−2)	1.10(−2)
0	2	8	9.36(−3)	9.39(−3)
0	2	9	6.72(−3)	6.77(−3)
0	2	10	3.78(−3)	3.79(−3)
0	2	11	1.36(−3)	1.36(−3)
0	2	12	2.61(−4)	2.64(−4)
0	2	13	5.97(−6)	5.92(−6)
0	3	0	8.44(−2)	8.44(−2)
0	3	1	1.79(−1)	1.79(−1)
0	3	2	1.35(−1)	1.36(−1)
0	3	3	4.18(−2)	4.17(−2)
0	3	4	1.58(−3)	1.55(−3)
1	2	0	2.58(−3)	2.56(−3)
1	2	1	7.07(−3)	7.00(−3)
1	2	2	1.00(−2)	9.93(−3)
1	2	2	1.16(−2)	1.16(−2)
1	2	4	1.26(−2)	1.26(−2)
1	2	5	1.35(−2)	1.35(−2)
1	2	6	1.39(−2)	1.39(−2)
1	2	7	1.32(−2)	1.31(−2)
1	2	8	1.11(−2)	1.10(−2)
1	2	9	7.86(−3)	7.75(−3)
1	2	10	4.28(−3)	4.23(−3)
1	2	11	1.49(−3)	1.45(−3)
1	2	12	2.73(−4)	2.78(−4)
1	2	13	6.41(−6)	6.08(−6)
1	3	0	1.01(−1)	1.01(−1)
1	3	1	2.14(−1)	2.14(−1)
1	3	2	1.62(−1)	1.62(−1)
1	3	3	5.04(−1)	4.98(−1)
1	3	4	1.91(−3)	1.84(−3)
2	2	0	5.46(−3)	5.44(−3)
2	2	1	1.48(−3)	1.48(−3)
2	2	2	2.05(−3)	2.05(−3)
2	2	3	2.30(−3)	2.30(−3)
2	2	4	2.38(−3)	2.39(−3)
2	2	5	2.43(−3)	2.43(−3)
2	2	6	2.38(−3)	2.38(−3)
2	2	7	2.15(−3)	2.15(−3)
2	2	8	1.69(−3)	1.69(−3)
2	2	9	1.09(−3)	1.10(−3)
2	2	10	5.25(−4)	5.26(−4)
2	2	11	1.43(−4)	1.44(−4)
2	2	12	1.65(−5)	1.69(−5)
2	2	13	3.09(−7)	3.01(−7)
2	3	0	1.66(−2)	1.66(−2)
2	3	1	3.53(−2)	3.53(−2)
2	3	2	2.68(−2)	2.69(−2)
2	3	3	8.40(−2)	8.37(−2)
2	3	4	3.27(−4)	3.19(−4)

[a] From the calculations described in Refs. [42, 43]
[b] $2.02(-3) \equiv 2.02 \times 10^{-3}$

calculations, and Table 9 gives the full details of the new translational basis sets. Let $\{m_{max}(AD), m_{max}(WF)\}$ denote the number of AD and WF basis functions in a given channel, and let (v, j) denote the vibrational and rotational quantum numbers in a state. The table shows that for the $F + H_2$ arrangement, we can obtain converged results with the basis $\{7, 4\}$ for channels $(0, 0), (0, 2), \ldots, (0, 10)$, $\{6, 4\}$ for $(0, 12)$, $(1, \text{even } j)$, and $(2, \text{even } j)$, $\{5, 3\}$ for $(0, \text{odd } j)$, $(1, 1)$, $(1, 3)$, $(3, \text{even } j)$, and $(4, \text{even } j)$, and $\{5, 1\}$ for $(1, 5)$, $(1, 7)$, and $(2\text{–}4, \text{odd } j)$. For the H + HF arrangements the new basis is $\{12, 5\}$ for the $v = (1, 0\text{–}11)$ channels, $\{11, 15\}$ for the $(1, 12\text{–}18)$ channels and the $v = 2$ or 3 levels, $\{7, 5\}$ for $(0, 0\text{–}13)$, $\{6, 5\}$ for the $v = 4$ level, $\{4, 5\}$ for $(0, 14\text{–}22)$, and $\{5, 4\}$ for the $v = 5$ or 6 levels. Table 10 compares the results of the current and previous calculations; clearly the agreement is excellent despite the substantial reduction in the number of Gaussians and the even more substantial reduction in the number of energy-dependent basis functions.

5. Conclusions

In this paper we have considered in detail many techniques to improve our method for calculating probabilities for reactive scattering processes in three dimensions. These new methods can provide substantial reductions in the amount of computational resources required to obtain accurate results.

Acknowledgements. This paper is based on a presentation made at the workshop of Parallel Computing for Chemical Reactivity, in Perugia, Italy, August 31–September 1, 1990. We are grateful to Antonio Laganà for the invitation to participate. We also thank David Chatfield and Ronald Friedman for helpful discussions of the work presented here. This work was supported in part by the National Science Foundation, NASA, the Minnesota Supercomputer Institute, and the Cray Research University Grant Program.

References

1. Cuccaro SA, Hipes PG, Kuppermann A (1989) Chem Phys Lett 154:155–164
2. Haug K, Schwenke DW, Shima Y, Truhlar DG, Zhang J, Kouri, DJ (1986) J Phys Chem 90:6757–6759
3. Zhang JZH, Kouri DJ, Haug K, Schwenke DW, Shima Y, Truhlar DG (1988) J Chem Phys 88:2492–2512
4. Launay JM, Le Dourneuf M (1989) Chem Phys Lett 163:178–188
5. Manolopoulos DE, Wyatt RE (1988) Chem Phys Lett 152:23–32
6. Manolopoulos DE, D'Mello M, Wyatt RE (1989) J Chem Phys 91:6096–6102
7. Pack RT, Parker GA (1987) J Chem Phys 87:3888–3921
8. Pack RT, Parker GA (1989) J Chem Phys 90:3511–3519
9. Schatz GC (1988) Chem Phys Lett 150:92–98
10. Schwenke DW, Haug K, Truhlar DG, Sun Y, Zhang JZH, Kouri DJ (1987) 91:6080–6082
11. Schwenke DW, Haug K, Zhao M, Truhlar DG, Sun Y, Zhang JZH, Kouri DJ (1988) J Phys Chem 92:3202–3216
12. Schwenke DW, Mladenovic M, Zhao M, Truhlar DG, Sun Y, Kouri DJ (1989) In: Laganà A (ed) Supercomputer algorithms for reactivity, dynamics and kinetics of small molecules. Kluwer Academic Publ, Dordrecht, pp. 131–168.
13. Zhang JZH, Miller WH (1987) Chem Phys Lett 140:329–337.
14. Truhlar DG, Schwenke DW, Kouri DJ (1990) J Phys Chem 94:7346–7352
15. Schlessinger L (1968) Phys Rev 167:1411–1423

16. Sun Y, Kouri DJ, Truhlar DG, Schwenke DW (1990) Phys Rev A 41:4857–4862
17. Sun Y, Kouri DJ, Truhlar DG (1990) Nucl Phys A 508:41c – 62c
18. Schwenke DW, Truhlar DG (1990) In: Glowinski R, Lichewsky A (eds) Proceedings of the Ninth International Conference on Computing Methods in Applied Sciences and Engineering, Paris 29 Jan–2 Feb 1990. Society for Industrial and Applied Mathematics, Philadelphia, pp. 291–310
19. Sloan IH, Brady J (1972) Phys Rev C 6:701–709
20. Brady J, Sloan IH (1974) Phys Rev C 9:4–15
21. Staszewska G, Truhlar DG (1986) Chem Phys Lett 85:4997–5003
22. Staszewska G, Truhlar DG (1987) J Chem Phys 86:2793–2804
23. Newton RG (1966) Scattering theory of particles and waves. McGraw-Hill, New York, Sect. 11.3
24. Schwenke DW, Truhlar DG (1984) Computer Phys Commun 34:57–66
25. Dongarra JJ, Moler CB, Bunch JR, Stewart GW (1979) LINPACK Users' Guide, Society for Industrial and Applied Mathematics, Philadelphia, pp. I.6–I.12
26. Dongarra JJ, Eisenstat SC (1984) ACM Trans Math Software 10:221–230
27. Dongarra JJ, Kaufman L, Hammarling S (1985) Argonne National Laboratory, Mathematics and Computer Science Division, Technical Memorandum no. 46, unpublished
28. Schwenke DW, Truhlar DG (1986) In: Dupuis M (ed) Supercomputer simulations in chemistry. Springer-Verlag, Berlin, pp. 165–197
29. Miller WH, Jansen op de Haar BMDD (1987) J Chem Phys 86:6213–6220
30. Zhang JZH, Chu S, Miller WH (1988) J Chem Phys 88:6233–6239
31. Abdallah J, Truhlar DG (1974) J Chem Phys 61:30–36
32. Staszewska G, Truhlar DG (1988) J Chem Phys 86:1646–1648
33. Manolopoulos DE, D'Mello M, Wyatt RE (1990) J Chem Phys 93:403–411
34. Zhao M, Truhlar DG, Schwenke DW, Yu C-h, Kouri DJ (1990) J Phys Chem 94:7062–7069
35. Golub GH, Van Loan CF (1983) Matrix computations. Johns Hopkins University Press, Baltimore, pp 119–124
36. Sack RA, Donovan AF (1972) Numer Math 18:465–478
37. Halvick P, Truhlar DG, Schwenke DW, Sun Y, Kouri DJ (1990) J Phys Chem 94:3231–3236
38. Varandas AJC, Brown FB, Mead CA, Truhlar DG, Blais NC (1987) J Chem Phys 86:6258–6269
39. Brown FB, Steckler R, Schwenke DW, Truhlar DG, Garrett BC (1985) J Chem Phys 82:188–201
40. Steckler R, Truhlar DG, Garrett BC (1985) J Chem Phys 82:5399–5505
41. Zhao M, Mladenovic M, Truhlar DG, Schwenke DW, Sun Y, Kouri DJ, Blais NC (1989) J Amer Chem Soc 111:852–859
42. Yu C-h, Kouri DJ, Zhao M, Truhlar DG, Schwenke DW (1989) Chem Phys Lett 157:491–495
43. Yu C-h, Kouri DJ, Zhao M, Truhlar DG, Schwenke DW (1989) Int J Quantum Chem Symp 23:45–58
44. Hamilton IP, Light JC (1986) J Chem Phys 84:306–317