A NONORTHOGONAL COORDINATE APPROACH TO ATOM–DIATOM PARALLEL REACTIVE SCATTERING CALCULATIONS

Antonio LAGANÀ[a1,*], Stefano CROCCHIANTI[a2], Noelia FAGINAS LAGO[a3], Leonardo PACIFICI[a4] and Gianni FERRARO[b]

[a] *Dipartimento di Chimica, Università di Perugia, Perugia, Italy; e-mail:* [1] *lag@impact.dyn.unipg.it,* [2] *croc@dyn.unipg.it,* [3] *noelia@dyn.unipg.it,* [4] *xleo@dyn.unipg.it*

[b] *Istituto di Chimica, Politecnico di Bari, Bari, Italy; e-mail: moldyn@libero.it*

Received September 18, 2002
Accepted November 21, 2002

Dedicated to Professors Petr Čársky, Ivan Hubač and Miroslav Urban on the occasion of their 60th birthdays.

Computational approaches to the evaluation of the reactive scattering properties of atom–diatom reactions are revisited. The aim is to exploit both the use of nonorthogonal coordinates in reactive scattering and the restructuring of related computer codes for concurrent computing. To this end, bond length, bond-order and hyperspherical bond order coordinate formalisms are examined for the collinear case. At the same time, the evolution of parallel models from coarse to fine granularity and the development of parallelization supports from directive libraries to programming environments are discussed. The scalability of related codes is tested by measuring the performances of restructured codes. The suitability of the use of nonorthogonal coordinates for scattering purposes is tested by performing collinear calculations for the H + H_2 reaction.

Keywords: Hamiltonian; Quantum chemistry; Reactive scattering; Computational approaches; Nonorthogonal coordinates.

The evolution of the formulation of electronically adiabatic reactive scattering and the progress in modern computer technologies are transforming the calculation of the detailed state-to-state quantum reactive probabilities for atom–diatom systems into a routine task. The main difficulty is at present associated with the high demand of computing resources when one has to deal with large values either of the total angular momentum **J** or of the atomic masses especially when the potential energy surface is structured and strongly bent arrangements are energetically accessible. Much higher difficulties are met when dealing with four- or more-atom reactive systems. In these cases the computational machinery is not yet so established to al-

low routine exact (converged with the total angular momentum) quantum calculations in spite of the fact that, when dealing with larger systems, one can adopt, in principle, an approach largely coincident with that developed for three atoms.

The most popular computational approach to the evaluation of quantum state-to-state probabilities of reactive systems is based on the definition of a continuity variable along which the scattering equations are integrated step by step from reactants to products (or from an intermediate situation to the asymptotes). From a direct or indirect comparison of the system wavefunction of the reactants with that of the products, one can compute the elements of the scattering **S** matrix from which all the scattering properties of the system can be derived[1]. A sketch of the main features of the quantum approaches to reactive scattering of atom–diatom systems is given.

Significant efforts are being made to progress towards accurate *a priori* calculation of scattering properties of large systems. To this end, some reactive scattering computational codes are being restructured to run in parallel[2] on innovative multiprocessor platforms. At the same time, however, alternative theoretical formulations of the problem easier to extend to four- and more-atom systems are being developed.

In this paper both aspects are discussed to some extent. In particular, the key features of quantum approaches to reactive scattering are revisited with the purpose of singling out those aspects of the calculations relevant either to the parallel restructuring or to an alternative formulation of the scattering equations. As to the computational aspects, the restructuring of the codes to exploit concurrent execution is examined and the adoption of some parallel models designed for multiprocessor computers is considered. As to the theoretical aspects, we focus our attention on atom–diatom reactions and, after illustrating the properties of some sets of nonorthogonal coordinates, we examine an application to the study of the traditional prototypical H + H_2 collinear reaction.

COMPUTATIONAL APPROACHES TO QUANTUM REACTIVE SCATTERING

As already mentioned, the theoretical evaluation of all the observable properties of a scattering process can be reduced to the calculation of the scattering **S** matrix. The **S** matrix elements can be determined by integrating the time-dependent Schrödinger equation

$$\hat{H}\Psi(\{x\}, t) = \mathrm{i}\hbar \frac{\partial}{\partial t} \Psi(\{x\}, t) \,, \qquad (1)$$

where $\hat{H}$ is the Hamiltonian of the system, $\Psi(\{x\},t)$ is its wavefunction and $\{x\}$ is any suitable set of coordinates. When using a time-dependent approach to integrate Eq. (*1*), the time variable t is taken as a continuity variable even when the Hamiltonian $\hat{H}$ is time-independent. In time-independent approaches, the time dependence is separated and, in order to integrate the resulting stationary Schrödinger equation, a proper combination of spatial coordinates is taken as a continuity variable. In both time-dependent and time-independent methods the computational procedure is usually simplified by separating not only the motion of the center of mass but also the rotation of the whole system (described in terms of the three Euler angles α, β and γ) from the motion along the internal coordinates. This allows an expansion of the global wavefunction Ψ in terms of its partial wave contributions[3], which are labeled after the eigenvalue J of the total angular momentum **J** of the system and after its projection on the quantization axis of the chosen reference frame (M or Λ according to whether reference is made to a laboratory-fixed or to a body-fixed frame, respectively). Accordingly, calculated elements of the **S** matrix for atom–diatom reactions (say A + BC → AB + C) relevant to this paper are labeled after v and j (the vibrational and the rotational quantum numbers of the reactant diatom) and after J and its projection on the quantization axis.

The Quantum Time-Dependent Approach

In our implementation of the time-dependent approach derived from ref.[4], the integration of Eq. (*1*) is carried out by repeatedly applying the time propagator $\exp(-i\hat{H}\tau/\hbar)$ on the system partial wavefunction $\Psi^{J\Lambda}(R,r,\Theta,t)$ (R, r and Θ are the reactant internal Jacobi coordinates, *i.e.*, the modulus of the mass-scaled atom–diatom center of mass vector ($\mathbf{R}_{A,BC}$), the modulus of the mass-scaled diatom internuclear vector ($\mathbf{r}_{BC}$) and the angle formed by them, respectively). The partial wavefunction is initially set as the function of the reactants in a particular initial vibrational state v,j multiplied by a factor specifying the range of energy available to the system along R. Then the time propagator is repeatedly applied till the wavefunction (the wavepacket) has spread over all the accessible configuration space including products. At each time step, the cut of the wavefunction in the product region is expanded into the related vibrational eigenfunctions. The time dependent coefficients of the expansion

$$C^{J}_{vj\Lambda,v'j'\Lambda'}(t) = \int_{r'} dr' \int_{\Theta'} d\Theta' \, \sin\Theta' P_{j'\Lambda'}(\Theta)\phi_{v'j'}(r')\Psi^{J\Lambda'} \; (R' = R'_{\infty}, r', \Theta', \tau) \qquad (2)$$

(where primed quantities refer to products, $P_{j'\Lambda'}(\Theta')$ is the normalized associated Legendre polynomial of the product rotational state j' and $\phi_{v'j'}(r')$ is the product vibrational function) are Fourier-transformed to get the time-independent (energy-dependent) elements of the **A** matrix

$$A^J_{vj\Lambda,v'j'\Lambda'}(E) = \frac{1}{2\pi}\int_{t=0}^{\infty} \mathrm{d}t\ \exp(\mathrm{i}Et/\hbar)\cdot C^J_{vj\Lambda,v'j'\Lambda'}(t) \qquad (3)$$

at the total energy E. One can then formulate the desired **S** matrix elements in terms of the **A** matrix ones using the following relationship

$$S^J_{vj\Lambda,v'j'\Lambda'}(E) = \left(\frac{k_{vj}k_{v'j'}}{\mu\mu'}\right)^{\frac{1}{2}} \frac{\hbar A^J_{vj\Lambda,v'j'\Lambda'}}{g(-k_{vj})}\ \exp(-\mathrm{i}k_{v'j'}R'_\infty)\ , \qquad (4)$$

where $g(-k_{vj})$ is the amplitude of the initial wavepacket with momentum $-\hbar k_{vj}$ and μ (μ') is the reduced mass of the reactants (products).

The kernel of the computational procedure can be schematized as follows in Fig. 1 (the loop over the angular variable is omitted for the sake of clarity).

The Quantum Time-Independent Approach

In our implementation of the time-independent approach derived from ref.[5] the hyperradius ρ (defined as $\rho^2 = R^2 + r^2 = R'^2 + r'^2$) is taken, as usual, as the continuity variable (reaction coordinate) for the integration of the

```
LOOP over initial states
    LOOP on time
        LOOP on R
            Apply the time step propagation
        END the loop on R
        LOOP on r
            Apply the time step propagation
        END the loop on r
        Update the terms needed to evaluate the S matrix elements
        Check for time integration termination
    END the loop on time
    Print out detailed S matrix elements
END the loop over initial states
```

FIG. 1
Scheme of the kernel of the time-dependent quantum code

scattering equations. The hyperradius, in fact, straightforwardly connects the strong interaction region to both reactant and product asymptotes (while for this connection Jacobi coordinates need a quite complex matching procedure at some intermediate molecular geometry). The other two internal coordinates of a hyperspherical description of atom–diatom reactions are the two hyperangles θ and χ. The integration of the time-independent, version of Eq. (*1*) relies now on the partitioning of the reaction coordinate ρ into N_S sectors and on the expansion (within each sector *i*) of each *n*-th partial wavefunction of parity *p* in terms of products of the Wigner rotation functions $\hat{D}^{Jp}_{M\Lambda}(\alpha,\beta,\gamma)$ of the three Euler angles, the surface functions $\Phi^{Jp}_{t\Lambda}(\theta,\chi;\rho_i)$ of the two internal hyperangles, and of the unknown functions $\psi^{Jpn}_{t\Lambda}(\rho)$ of the hyperradius ρ.

The surface functions are calculated by solving the two-dimensional bound state problem

$$\left[\hat{T}_{\mathrm{h}} + \frac{15\hbar^2}{8\bar{\mu}\rho_i^2} + \hbar^2 G_J + F\hbar^2\Lambda^2 + V(\rho_i,\theta,\chi_\tau) - \varepsilon^{Jp}_{t\Lambda}(\rho_i)\right]\Phi^{Jp}_{t\Lambda}(\theta,\chi_\tau;\rho_i) = 0\,, \qquad (5)$$

where $G_J = J(J+1)(A+B)/2$, $F = C - (A+B)/2$, $A^{-1} = \bar{\mu}\rho_i^2(1+\sin\theta)$, $B^{-1} = 2\bar{\mu}\rho_i^2\sin^2\theta$, $C^{-1} = \bar{\mu}\rho_i^2(1-\sin\theta)$, $\bar{\mu}$ is the hyperspherical reduced mass (usually equal to $(m_A m_B m_C/(m_A+m_B+m_C))^{1/2}$, $V(\rho_i,\theta,\chi_\tau)$ is the potential energy and $\varepsilon^{Jp}_{t\Lambda}(\rho_i)$ is the eigenvalue. In the same equation the hyperangular term $\hat{T}_{\mathrm{h}}$ reads

$$\hat{T}_{\mathrm{h}} = -\frac{\hbar^2}{2\bar{\mu}\rho_i^2}\left(\frac{4}{\sin 2\theta}\frac{\partial}{\partial\theta}\sin 2\theta\frac{\partial}{\partial\theta} + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\chi_\tau^2}\right).$$

The resulting set of coupled differential equations has the form

$$\left(\frac{\partial^2}{\partial\rho^2} + \frac{2\bar{\mu}E}{\hbar^2}\right)\psi^{Jpn}_{t\Lambda}(\rho) = \frac{2\bar{\mu}}{\hbar^2}\sum_{t'\Lambda'}\left\langle\Phi^{Jp}_{t\Lambda}\hat{D}^{Jp}_{\Lambda M}\middle|\hat{H}_{\mathrm{i}}\middle|\Phi^{Jp}_{t'\Lambda'}\hat{D}^{Jp}_{\Lambda' M}\right\rangle\psi^{Jpn}_{t'\Lambda'}(\rho)\,. \qquad (6)$$

In Eq. (*6*) the internal Hamiltonian $\hat{H}_{\mathrm{i}}$ reads

$$\hat{H}_{\mathrm{i}} = \hat{T}_{\mathrm{h}} + \hat{T}_{\mathrm{r}} + \hat{T}_{\mathrm{c}} + \frac{15\hbar^2}{8\bar{\mu}\rho_i^2} + V(\rho,\theta,\chi_\tau)\,, \qquad (7)$$

where subscripts "r" and "c" mean "rotational" and "Coriolis", respectively. $\hat{T}_r$ and $\hat{T}_c$ are formulated as:

$$\hat{T}_r = A(\rho_i, \theta) J_x^2 + B(\rho_i, \theta) J_y^2 + C(\rho_i, \theta) J_z^2$$

and

$$\hat{T}_c = -\frac{i\hbar \cos\theta}{\bar{\mu}\rho_i^2 \sin^2\theta} J_y \frac{\partial}{\partial \chi_\tau} .$$

As the propagation progresses from small to large ρ values, the system leaves the region of strong coupling to reach the asymptotes where the solution is conveniently mapped into final states by using the Jacobi coordinates of the appropriate arrangement and by imposing the proper boundary conditions to evaluate the **S** matrix elements.

The time-independent computational procedure (corresponding to the time-dependent, one of Fig. 1) is articulated into two parts (each part can be a different program). The first part calculates sector quantities at fixed value of the reaction coordinate after solving the related (one dimensional in the case of collinear systems) eigenvalue problems as schematized in Fig. 2.

The second part of the procedure propagates through the sectors the fixed total angular momentum solution for a batch of N_E energy values as schematized in Fig. 3.

PARALLEL RESTRUCTURING

As already mentioned, part of our research is devoted to parallel restructuring the reactive scattering codes in order to speed up their execution. To this end, use was made of MPI [6] and different models of parallelism were exploited to cope with the complexity of the related computational procedures.

LOOP on N_S sectors
    Generate fixed reaction coordinate cuts of the potential
    Calculate the sector eigenvalues and eigenfunctions and store the eigenvalues
    Calculate and store the intrasector coupling matrix
    IF (not first sector) Calculate and store the intersector overlap integrals
END the sector loop

FIG. 2
Scheme of the kernel of the time-independent quantum code calculating sector quantities

The Coarse Grain Parallelization

Most of our work on parallel restructuring was carried out using models based on coarse granularity. The simplest coarse grain parallel model distributes the whole program to all the available nodes N_P and executes it in parallel for different values of the parameters in a typical single program multiple data (SPMD) fashion. Although this could appear to be the most desirable situation, an SPMD approach shows severe limitations when the matrices become too large to fit into the node memory. On the contrary, a useful feature of the SPMD approach is the fact that it allows to make regular and minimal the message passing activity and to confine the I/O activity to/from disk.

The Time-Dependent Code

A more articulated way of parallelizing a quantum time dependent computer program is to adopt a task farm model. In this model a master process distributes to each worker node a fixed J calculation for a given initial vibrational state and a given interval of translational energy by dynamically assigning the computational workload. As J increases, however, the demand of computing resources also increases, not only in terms of the computing time needed to perform each single J calculation (with an obvious effect on the load imbalance) but also in terms of the size of the involved matrices.

For this reason the coarse grain parallel organization needs to be pushed to a lower level. This can be achieved by decoupling a fixed J calculation into several fixed Λ concurrent tasks. However, contrary to J, Λ is not conserved and, therefore, the decoupling of fixed Λ calculations has to be en-

```
LOOP on N_S sectors
    Read intra and intersector quantities
END the sector loop
LOOP on N_E energies
    Embed the energy dependence into the coupling matrix
    LOOP on N_S sectors
        Propagate the wavefunction through the sectors
    END the sector loop
    Calculate and store detailed S matrix elements
END the energy loop
```

FIG. 3
Scheme of the kernel of the time-independent quantum code propagating the wavefunction and calculating the **S** matrix

forced through dynamic approximations like the centrifugal sudden one[7]. As a result of such a decoupling, one can perform a step-propagation of the wavepacket for all the blocks of fixed Λ values belonging to the same value of *J* and then combine together the various (fixed Λ) contributions. This step decomposition of the wavepacket domain allows to cope with the demand of larger memory associated with the increase in *J* since it leads to an almost constant, per node demand of memory at the expense of an increase in the number of processors used.

Such a parallel model was first tested[8] on the Cray T3E of EPCC (Edinburgh, U.K.) using three nodes for the simplest case of $J = 0$ and $J = 1$. The measured speedup was 2.6. This indicates that the proposed model is quite effective in reducing computing times despite the fact that, when the size of the problem is increased, I/O may become a true bottleneck. This is due to the fact that all the I/O traffic is channeled through node 0 that acts as a master. The efficiency of the model is also penalized by the fact that when *J* gets larger, the work statically assigned to the workers becomes increasingly unbalanced. To gain significant improvement, when generalizing this model to larger *J* values, fixed *J* calculations are carried out in pairs (for all the related $J + 1$ fixed Λ components) by balancing dynamically the workload. In this model the I/O traffic is still channeled through the master. Yet, in this case this does not constitute a real bottleneck since node 0, contrary to what was happening with the previous model, does not carry out any calculations.

A fundamental limit of this model is the fact that its minimum computational grain consists of single-*J*–fixed-Λ computational blocks. This implies that the maximum value of the total angular momentum quantum number that can be handled by the model has to be 3 units smaller than the number of available processors. Furthermore, to keep all the processors busy all the time one has to run simultaneously a pair of complementary (summing up to the maximum allowed value of *J*) fixed *J* calculations.

To evaluate the performances of the improved model, the calculations were carried out on the Origin 3800 of Cineca (Bologna, Italy) in local memory mode using MPI. The measured times indicate that, though the growth of communication time associated with an increase in the number of allowed Λ values slightly penalizes the efficiency of the code, the model is quite effective in reducing the overall computing time[9].

The Time-Independent Code

More complex is the coarse-grain parallel restructuring of the time-independent code for which an SPMD distribution of fixed J calculations is inadequate. In this respect, typical is the articulation of the first part of the calculation (in three dimensions this is a program in itself (named ABM in our procedure) that performs the calculation of fixed ρ eigenvalues, surface functions, overlap and coupling matrices). For ABM a dynamic task farm model distributing the N_S sector calculations on the N_P available nodes is adopted. According to this parallel organization, the calculation of each block of sector surface functions is assigned to the first available node by a dynamic allocation of the workload controlled by the master process (see Fig. 4). The worker node calculates the sector surface functions, evaluates the matrix of the overlap integrals between these functions and those of the previous sector at the common border and constructs the coupling matrix. In this case, shared memory would allow an access to data of different sectors in the proper sequence. Using the message passing paradigm of MPI, this can be obtained either by transferring the data from the node that has made the calculations to the node needing the information (this requires the setting of a few barriers to channel the data in the appropriate sequence) or by replicating the calculation of the surface functions of the previous sector when needed (this is carried out by utilizing the related eigenvectors without solving again the eigenvalue problem). The replication of the calculation of the surface functions of the previous sector (see Fig. 5) together with a dynamic distribution of the work, greatly reduces the communication time and minimizes the work imbalance.

The calculated coupling matrix is stored on disk, together with the overlap matrix, for use by the subsequent LOGDER program that carries out the propagation of the solution. (Again, in three dimensions, this part of the

```
                MASTER PROCESS

Read input data
Send input data to the workers
LOOP on sector index
     Calculate the value of ρ at the sector midpoint
     Call MPI_SEND(ρ)
END LOOP on sector index
```

FIG. 4
Pseudo MPI code for the master process in ABM

calculation is a program in itself.) This parallel model has been implemented for $J = 0$ on the CRAY T3E of EPCC (Edinburgh, U.K.) using up to 128 processors. The individual processor computing time never exceeds 10% of the average one and the speedup is never smaller than 70% of the ideal value[2].

For the energy loop of LOGDER (see Fig. 3) a similar task farm structure looping on energy was adopted. However, the excellent performance obtained for the $J = 0$ runs was not equalled by the higher J calculations. This is due to the fact that, when using MPI, to deal with matrices having a size too large to be accommodated inside the local node memory communication overheads becomes important. As already mentioned, the only reasonable solution to this drawback is the reduction of the dimensionality of the problem by introducing dynamic approximations (like those associated with the freezing of the projections of the total angular momentum or of the collision angle). As an alternative, an efficient solution is the emulation of the shared memory environment using an abstract level coordination language.

For the fixed-angle calculations, the two parts of the computation become, respectively, sections (a) and (b) of the same program. In the case of Section a, a task farm model with a static data decomposition can be used. This means that sectors are grouped in blocks of approximately the same

```
                    WORKER PROCESS

      Receive input data
10    Call MPI_RECV(ρ)
      LOOP on Λ
          Construct the primitive basis set at the given value of ρ
          Solve the angular Schrödinger equation by expanding in the primitives
          Store on disk eigenvalues and eigenvectors
          IF(not first sector) then
              Construct the basis set at the previous value of ρ
              Read from disk related eigenvectors
              Compute overlap integrals at the common edge of the two sectors
          END IF
      END LOOP on Λ
      Calculate the coupling matrix
      Store on disk the coupling matrix for use from LOGDER
      GOTO 10
```

FIG. 5
Pseudo MPI code for the worker processes of ABM

size and each block is assigned to a different worker processor for execution. The number of generated sector blocks, obviously, depends on N_P, the number of nodes available on the machine used. At the end of Section a quantities calculated on each node are broadcast to all other nodes to assemble the whole coupling matrix. In the case of Section b, a task farm model is also adopted. However, the distributed task (the propagation section) is now assigned dynamically for different values of the total energy[2].

New Prospects in Concurrent Computing

As already mentioned, coarse-grain parallel models find a limitation in the maximum size of the matrices that can be dealt. The only way of making the program really scalable with an increase in the size of the involved matrices is to go beyond outer levels of parallelization and dig deep into the code fine structure. This implies two orders of difficulties. The first type of difficulty is that one needs to know intimately the code and the theory laid behind it. This requires a specific expertise and the ability of mastering the details of the code during its evolution to maturity (that is difficult to sustain as more and more contributions from different laboratories add up). The second type of difficulty is related to the availability of suitable tools to optimize the parallel structure of the program and its portability on different platforms without losing efficiency. This is particularly true for those codes for which the efficiency of the parallel organization is strongly data-dependent.

Fine-Grain Parallelization

To investigate the possibility of pushing the parallelization to a very fine grid we concentrated on the time-dependent code. The critical step of the calculation is the application of the time propagator that for each dimension iterates the computation of the following expression

$$\mathbf{D} = \mathbf{A} * \mathbf{C} + \mathbf{C} * \mathbf{B}^T + \mathbf{V} \odot \mathbf{C}, \qquad (8)$$

where **D** is the matrix storing the wavepacket modified by the action of the Hamiltonian and $\mathbf{V} \odot \mathbf{C}$ is the direct product of the potential matrix and the wavepacket.

In a sequential approach, all the matrices need to be stored in their entirety in the central memory during the propagation of the wavepacket. However, the matrices that are modified during the execution are **D** and **C**.

Therefore, in the task farm model adopted for this part of the code, the master can broadcast to the workers the matrices **A** and **V** during the first time step. Yet, the **C** matrix (representing the wavepacket in the previous time step) has to be sent to the workers at every iteration. This can be done by partitioning the matrix **C** in a way that optimizes the load for the worker nodes. The workers perform the operation $\mathbf{A} * \mathbf{C} + \mathbf{V} \circ \mathbf{C}$ on the received partition of **C** while the master carries out the calculation of $\mathbf{C} * \mathbf{B}^T$ by multiplying the rows of **C** times the rows of **B** (that are the columns of **B** transposed). At the end of the calculations, the master collects the results from the workers and assembles the matrix **D**.

However, since the matrices **A**, **B** and **V** remain unaltered, the task farm model can be improved if the matrix domain is partitioned by rows and a memory management more respectful of the data hierarchies (including I/O) is used. In this case when the matrices are symmetric one can write the following algorithm

$$\mathrm{Row}(i, \mathbf{H}) =$$

$$\sum_k A(i, k) * \mathrm{Row}(k, \mathbf{C}) + \mathrm{Row}(i, \mathbf{C}) * \mathbf{B}^T + \mathrm{Row}(i, \mathbf{V}) * \mathrm{Row}(i, \mathbf{C}) , \qquad (9)$$

in which the first term of the expression on the right-hand side of the equation (the calculation of the *i*-th row of the product matrix $\mathbf{A} * \mathbf{C}$) is computed by multiplying the elements of the *i*-th row of **A** times all the rows of **C**. The product in the second term of the expression is computed in the same way by avoiding the transposition of matrix **B**. The third term of the expression is computed by taking the scalar product of the corresponding rows of the two matrices. In this way, the algorithm becomes highly efficient even for matrices whose size exceeds that of the local memory. In the related model, in fact, memory requests are dramatically reduced thanks to the domain partitioning and computing times are drastically shortened thanks to the concurrency between master and worker operations. Moreover, communications are largely decreased both in the startup and in the propagation iteration phase. As a matter of fact, the final assemblage of the **D** matrix is performed through a reduce operation on the data of the processes: the master process builds up the $\mathrm{Row}(i, \mathbf{V}) * \mathrm{Row}(i, \mathbf{C})$ product while the worker processes prepare the necessary vectors. Then **D** is stored in the logical space of the master[10].

Benchmarks have been performed on a Linux SST cluster implemented as a hybrid Mosix–Beowulf architecture in which the various features of the

two architectures coexist. The Beowulf approach guarantees the compatibility with applications based on the usual PVM and MPI parallelization libraries. The Mosix approach guarantees an optimal exploitation of the computing resources thanks to its advanced clustering features like the automatic migration of the processes and the real time balance of the load. In this way the Beowulf architecture profits from the Mosix one. The cluster is made of 9 machines (whose characteristics are given in Table I) of which one is used as front end.

A networking scheme of the *enclosed* type has been chosen. In this scheme the front end acts as a gateway to the outer network. Node communication is supported by Intel e1000 boards adopting the Gigabit Ethernet standard on a UTP cabling. The interconnection has been obtained using a 3Com Gigabit Switch. The operating system adopted is RedHat Linux 7.2 including the updates of the kernel to the last version supported by Mosix.

Speedups measured for matrix dimensions of 600 by varying the number of nodes used are given in Table II.

TABLE I
Hardware configuration of the clustered nodes

	Front end	Clustered nodes
Case	rack-mount 1U with 2 movable HD slits	rack-mount 1U with 2 movable HD slits
Mother board	dual processor with server Works Chipset	dual processor with VIA Chipset
CPU	2 Pentium III, Coppermine 1 GHz	2 Pentium III, Coppermine 1 GHz
RAM	2048 MB	2048 MB
Controller	PCI SCSI Adapter 7892P integrated EIDE server Works Controller	integrated EIDE VIA controller
Discs	2 U-SCSI 19.2 GB Discs	1 IDE 40 GB Disc
Network	NIC Intel e1000 PCI Copper Gigabit Ethernet 2 NIC Intel 2100 pro fast Ethernet integrate	NIC Intel e1000 PCI Copper Gigabit Ethernet 2 NIC Intel 2100 pro fast Ethernet integrate

TABLE II
Measured speedups

N_p	1	3	5	7
Speedup	1.00	2.46	4.57	6.40

A Coordination Language for Parallel and Grid Computing

The need for pushing the portability of the codes when distributing the calculations becomes dramatic when exploiting the potentialities of a grid. In this case the availability of user-friendly abstract parallelization tools is vital. This subject has been given specific attention by European Community Initiative COST in chemistry that has launched the action D23 (METACHEM: Metalaboratories for Complex Cornputational Applications in Chemistry) whose objective is to promote the institution of Chemical Metalaboratories (groups of chemical laboratories grafted on a computer grid)[11].

A specific effort in this direction is the development of programming environments and of the related coordination languages+.

We are actively working on the testing of a programming environment based upon the definition of some structural templates (SkIE) being developed by the group of Prof. Vanneschi[12] in Pisa. The building blocks of SkIE (skeletons), usually employed to construct parallel applications, refer either to stream (the flow sequence of the information) or to data (the allocation of the information) models. Typical stream-parallel structures are the farms, the pipelines and the loops. The farm consists of a replication of a function into a number of identical and independent workers, to which the stream elements are scheduled according to a load balance strategy. The pipeline consists of a parallel implementation of a set of functions into cascade stages through which the elements flow. The loop consists of a data-driven iterative computation through which the stream elements and their transformations flow until a certain condition is satisfied. Typical data-parallel structures are the map, the reduce and the compose. The map consists of a replication of a function into a number of identical and independent workers to which the elements of the data structure are distributed. The reduce consists of a parallel reduction of a data structure by binary associative and commutative operations. The compose consists of a sequential set of functions expressing a parallel computation-replication and partitioning

+ A programming environment is a software development system that enables the programmer to efficiently irnplement parallel applications by making use of parallel structures defined at an abstract level without resorting to the use of classic parallel libraries such as MPI. The coordination language of a given programming environment is a set of instructions and directives understandable by the compiler which allow the implementation of the application in that particular environment.

of data with communications designed according to a predetermined stencil[13]. In both stream-parallel and data-parallel paradigms, some data can be replicated.

SkIE has recently evolved into a more flexible tool (called ASSIST[14]) in which a new highly flexible parallel structure, the *parmod* module, has been introduced. This allows the programmers to freely organize the structure of the module and the sharing of the objects.

The key advantage of ASSIST is the fact that it combines the structured parallel programming philosophy with an object oriented nature. In fact, the compiler of ASSIST is based upon a library of extremely powerful classes and methods. As an example, to distribute the matrices of the time- dependent code in blocks of columns and rows the parametric class *Collective* < *IMP* > of the ASSIST library was used to properly manage communications. In this way it has been possible to implement *broadcast* and *scatter* classes to send information to the nodes. At the same time, on the way back, it has been possible to collect the results produced by the nodes by using a *reduce* operation implemented *via* a gathering function. I/O operations on a communication channel were, in turn, implemented using the put and get methods of the used collectives of communication[15].

NONORTHOGONAL COORDINATES

Theoretically, it is well known that, after separating the motion of the center of mass, the Hamiltonian of the atom–diatom problem is six-dimensional and needs a pair of vectors to describe the collision process of three nuclei. These vectors are most often taken to be the already defined mass-scaled Jacobi vectors $\mathbf{R}_\tau$ and $\mathbf{r}_\tau$, where τ labels the ordering choice made for the A, B and C masses. A clear advantage of this choice is that the Hamiltonian

$$\hat{H} = -\hbar^2 / 2\mu\left(\nabla^2_{\mathbf{R}_\tau} + \nabla^2_{\mathbf{r}_\tau}\right) + V\left(R_\tau, r_\tau, \Theta_\tau\right) \qquad (10)$$

is expressed in a diagonal and symmetric form. In general, in a full dimensional treatment, the solution is formulated as a function of the three Euler angles associated with the rotation of the rigid triatomic system and of the three internal coordinates R, r (the moduli of $\mathbf{R}_{A,BC}$ and $\mathbf{r}_{BC}$, respectively) and the angle Θ formed by them.

From Arrangement to Process Coordinates

By considering for simplicity the collinear case in Jacobi coordinates, the Hamiltonian reads

$$\hat{H} = -\frac{\hbar^2}{2\mu}\left(\frac{\partial^2}{\partial R^2} + \frac{\partial^2}{\partial r^2}\right) + V(R, r) \qquad (11)$$

with $V(R,r)$ being the collinear cut of the atom–diatom potential.

As already mentioned, Jacobi coordinates are arrangement coordinates. This means that they are suitable to support an expansion of the global wavefunction in the basis set of the related arrangement. Therefore, when using Jacobi coordinates, one can properly deal with inelastic problems since, in this case, there is no change of arrangement. Yet, when dealing with reactive processes, as already mentioned, the change of the nature of the system requires a switch from reactant (R, r) to product (R', r') coordinates (as usual, product-related quantities are primed). To this end, one usually defines in the strong interaction region a set of circular coordinates centered on a classically inaccessible point of the ridge separating reactants from products whose radius coincides with r in the entrance channel and r' in the exit channel. However, as already mentioned, in three or more dimensions, significant difficulties are met when carrying out the matching between product and reactant representations using Jacobi coordinates (especially for asymmetric systems).

The Bond Length Coordinates

The matching procedure can be avoided by using orthogonal coordinates smoothly evolving from the reactant to the product arrangement. This is the case of the hyperspherical coordinates, for which the hyperradius can be taken as a continuity variable all the way through from collapsed arrangements to both reactant and product asymptotes. However, as ρ increases, hyperspherical coordinates (that still need to be switched into the proper Jacobi coordinates every time a mapping to asymptotic arrangements is needed) compress all the bound degrees of freedom within increasingly smaller portions of the angular variables. This leads to some computational difficulties that can be overcome using tangent hyperspherical coordinates[16].

Another viable alternative to the use of orthogonal Jacobi coordinates is the use of nonorthogonal bond length coordinates. In the collinear A +

BC → AB + C process, the bond length coordinates to be considered are r_{BC} and r_{AB} (associated with the two vectors $\mathbf{r}_{BC}$ and $\mathbf{r}_{AB}$ connecting the exchanged atom B to A and C, respectively). An advantage of using the r_{BC} and r_{AB} bond lengths as coordinates for dynamic purposes lies in the fact that they are also used to formulate the potential (no additional calculation is needed for their evaluation) and that both reactant and product diatom are equally well represented. In these coordinates, the related Hamiltonian reads

$$\hat{H} = C_{rrx} \frac{\partial^2}{\partial r_{BC}^2} + C_{rry} \frac{\partial^2}{\partial r_{AB}^2} + C_{rxy} \frac{\partial^2}{\partial r_{BC} \partial r_{AB}} + V(r_{BC}, r_{AB}) , \qquad (12)$$

where $C_{rrx} = -\hbar^2/2\mu_{BC}$, $C_{rry} = -\hbar^2/2\mu_{AB}$, $C_{rxy} = \hbar^2/m_B$ with m_B being the mass of the B atom and μ_{BC} and μ_{AB} being the reduced masses of the BC and AB diatoms, respectively.

Bond length coordinates have already been used in the literature to calculate static and dynamic properties of three-atom systems (see, for example, refs[17–19]). However, as is apparent from Eq. (*12*) and from the work just mentioned, the price to pay is a formulation of the Hamiltonian that is more complicated than that for Jacobi coordinates and a (trivial in the collinear case) conversion to Jacobi coordinates at the asymptotes to perform the related analysis.

The Collinear H + H_2 Reactive Probabilities

To test the computational machinery of reactive scattering programs based on the use of nonorthogonal coordinates we have performed an exact time-dependent calculation of the collinear reactive probabilities of the H + H_2 system (universally accepted as the archetype of chemical reactions) using bond length coordinates. A more general discussion on the use of these coordinates for reactive scattering studies of larger systems will be given elsewhere[20]. The code is also built to take advantage of the fine grain parallelization algorithms discussed in the previous section allowing the achievement of high speedups when the matrices handled in the calculation become very large due to the high number of degrees of freedom of the system.

As usual, the H mass was taken to be 1.00783 a.m.u. The calculations were performed for the ground vibrational state of the reactants using a grid of 64 × 64 points. The potential energy surface used for the calculations was that of ref.[21] The initial wavepacket was given an energy of

0.7 eV. Calculated reactive probabilities are plotted in Fig. 6 as a function of the total energy of the system (given in eV) for increasing product vibrational states starting from the ground one (from top panel down). The plots show the usual structure of the H + H_2 reactive probabilities that has been reported in the literature since the early days of quantum reactive scattering calculations[22].

New Prospects: Bond-Order Type Coordinates

Other types of nonorthogonal coordinates are relevant to the development of alternative approaches to reactive scattering. Of particular interest are the coordinates derived from the bond-order (BO) concept which was suggested long time ago by Pauling[23] in an heuristic fashion. The BO concept has since then revisited in the literature for several purposes including reactive scattering calculations[24].

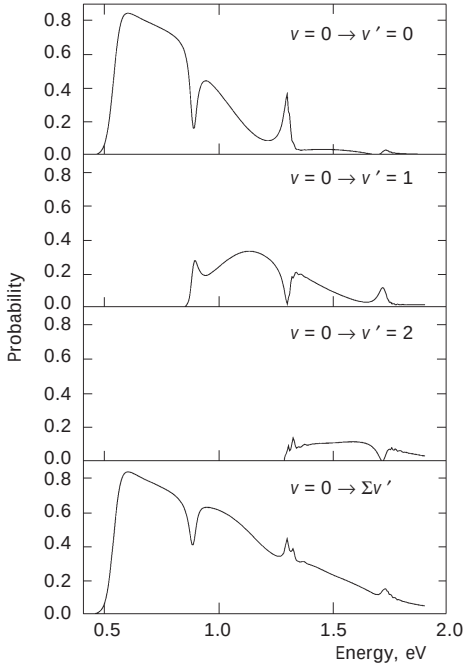

FIG. 6
Collinear reactive probabilities for H + H_2

The Bond-Order Coordinates

A few years ago we started using BO coordinates in a systematic way to fit potential energy surfaces of few-atom systems. This originated from the idea that a BO space would be more natural to map the different contributions to the interaction of the various regions of internuclear distances and that the strength of a chemical bond can be associated with the exponential of its displacement of the internuclear distance from equilibrium. Accordingly, BO coordinates for the A + BC → AB + C collinear reaction were defined as

$$n_{BC} = \exp\left[-\beta_{BC}\left(r_{BC} - r_{BC}^0\right)\right]$$
$$n_{AB} = \exp\left[-\beta_{AB}\left(r_{AB} - r_{AB}^0\right)\right], \qquad (13)$$

where β_i and r_i^0 are empirical parameters. To relate the BO variables to observable quantities (after all, this was the aim for which they were proposed in the first instance[23]), a link to diatomic vibrational spectroscopic properties was made[25]. This was obtained by modeling the atom–atom interaction as a second-order polynomial in the related BO coordinate. Then by setting β_i and r_i^0 as the force constant of diatom *i* and its equilibrium distance, respectively, one obtains the corresponding Morse potential and vibrational eigenvalues. Higher-order polynomials, whose parameters were optimized not only to spectroscopic constants but also to multipole expansions of the long-range interaction, gave an accurate reproduction of both spectroscopic and dispersion properties[25].

BO polynomials were also successfully used to represent the interaction of triatomic systems. This is based on a many body expansion of the interaction[26]. Then the two-body terms are formulated as fourth-order polynomials in the related BO coordinate while the three-body term is given as a cross product of powers of different BO variables up to the sixth order[27].

Like their parent bond length coordinates, BO coordinates are nonorthogonal and properly describe the evolution of an atom–diatom reactive system from the initial to the final arrangement with no need for carrying out a complex matching operation (from this the name of process coordinates). Further properties of the BO coordinates are the fact that the space is inverted (the zero of the BO coordinate corresponds to an infinite bond length while large BO values correspond to short bond lengths) and confined into a finite volume.

These properties are of some advantage in scattering calculations. As a matter of fact, the BO collinear Hamiltonian has the simple formulation:

$$\hat{H} = C_{nnx}\left[n_{\mathrm{BC}}^2 \frac{\partial^2}{\partial n_{\mathrm{BC}}^2} + n_{\mathrm{BC}} \frac{\partial}{\partial n_{\mathrm{BC}}}\right] + C_{nny}\left[n_{\mathrm{AB}}^2 \frac{\partial^2}{\partial n_{\mathrm{AB}}^2} + n_{\mathrm{AB}} \frac{\partial}{\partial n_{\mathrm{AB}}}\right] +$$

$$C_{nxy} n_{\mathrm{AB}} n_{\mathrm{BC}} \frac{\partial^2}{\partial n_{\mathrm{BC}} \partial n_{\mathrm{AB}}} + V(n_{\mathrm{BC}}, n_{\mathrm{AB}}) \qquad (14)$$

with the coefficients being $C_{nnx} = -(\hbar\beta)^2/2\mu_{\mathrm{BC}}$, $C_{nny} = -(\hbar\beta)^2/2\mu_{\mathrm{AB}}$ and $C_{nxy} = (\hbar\beta)^2/m_{\mathrm{B}}$ when the breaking and forming diatoms have the same force constant β. The BO formulation of the Hamiltonian is currently being used for quantum time-dependent[20] and classical trajectory[28] calculations. It is also being tested for variational (both one-dimensional[29] and multidimensional[30] reactive calculations.

The Hyperspherical Bond-Order Coordinates

The BO space can also be given a polar representation by adopting the hyperspherical BO (HYBO) coordinates. For collinear systems the two HYBO coordinates are

$$\rho = \sqrt{n_{\mathrm{AB}}^2 + n_{\mathrm{BC}}^2}$$

$$\alpha = \arctan n_{\mathrm{BC}} / n_{\mathrm{AB}} \ . \qquad (15)$$

An extension to more dimensions is obtained by adding more angles.

An interesting property of the HYBO space is that all the coordinates, including ρ, have a finite range of excursion (full fragmentation corresponds to the zero of the BO space and the collapsed system is confined inside the various $\exp[2\beta_i r_i^0]$ limiting values). Another interesting property of HYBO space is that fixed collision angle isoenergetic contours of the interaction show an almost spherical symmetry (particularly for collinear cuts of the atom–diatom potential)[31,32]. This has suggested the possibility of constructing a functional form for representing the potential energy surface of atom–diatom systems as a proper combination of rotating bond-order (ROBO) functionals. The ROBO functional, in fact, is based on the assumption that

the reaction channel can be modeled as a bond-order diatomic-like potential in ρ whose parameters vary with a that now plays the role of reaction coordinate[31,32]. The generalized form of the ROBO potential (LAGROBO) has been successfully used to fit the potential energy surface of several three-atom systems and has also been extended to four-atom systems[33].

As to their use in dynamic calculations, it is worth reemphasizing here that HYBO coordinates are nonorthogonal hyperspherical coordinates. Nonetheless, for each of them, it can still be identified a dominant role. Such a role, however, is different from that of the usual orthogonal hyperspherical coordinates. As a matter of fact, the hyperradius ρ does not play the role of reaction coordinate. As already mentioned, this role is played, instead, by the angle α that gradually connects reactants to products. The coordinate ρ describes, instead, a collective bound motion that evolves from the oscillation of the reactant diatom to the oscillation of the product diatom. For this reason a basis set in ρ has been used to expand the bound states of the reaction channel at fixed values of α [29].

In the HYBO coordinate formalism the collinear Hamiltonian reads

$$\hat{H} = C_{\alpha\alpha}(\alpha)\frac{\partial^2}{\partial\alpha^2} + C_{\alpha}(\alpha)\frac{\partial}{\partial\alpha} + C_{\rho\alpha}(\rho,\alpha)\frac{\partial}{\partial\rho}\frac{\partial}{\partial\alpha} +$$

$$C_{\rho\rho}(\rho,\alpha)\frac{\partial^2}{\partial\rho^2} + C_{\rho}(\rho,\alpha)\frac{\partial}{\partial\rho} + V(\alpha,\rho)\ , \qquad (16)$$

where, for the A = C case (for clarity in this symmetric case we label the *C* coefficients using the superscript s)

$$C^{s}_{\alpha\alpha}(\alpha) = \beta_D^2\,\frac{\sin^2 2\alpha}{2}\left[-\frac{\hbar^2}{2\mu_D} - \frac{\hbar^2}{2m_B}\right] \qquad (17)$$

$$C^{s}_{\alpha}(\alpha) = \beta_D^2\,\frac{\sin 4\alpha}{2}\left[-\frac{\hbar^2}{2\mu_D} - \frac{\hbar^2}{2m_B}\right] \qquad (18)$$

$$C^{s}_{\rho\alpha}(\rho,\alpha) = \beta_D^2\rho\,\frac{\sin 4\alpha}{2}\left[-\frac{\hbar^2}{2\mu_D} - \frac{\hbar^2}{2m_B}\right] \qquad (19)$$

$$C^{s}_{\rho\rho}(\rho,\alpha) = \beta_D^2\rho^2\left[-\frac{\hbar^2}{2\mu_D}(\cos^4\alpha + \sin^4\alpha) + \frac{\hbar^2}{2m_B}\frac{\sin^2 2\alpha}{2}\right] \qquad (20)$$

$$C^{s}_{\rho}(\rho,\alpha) = \beta_D^2\rho\left[-\frac{\hbar^2}{2\mu_D}\left(1+\frac{\sin^2 2\alpha}{2}\right) - \frac{\hbar^2}{2m_B}\frac{\sin^2 2\alpha}{2}\right] \qquad (21)$$

with $\beta_D = \beta_{BC} = \beta_{AB}$ and $\mu_D = \mu_{BC} = \mu_{AB}$.

CONCLUSIONS

Some streams of progress in investigating atom–diatom reactions carried out in our group are outlined in this paper. The first line of research is concerned with the design of computational algorithms suited to run concurrently on parallel platforms and with the development of a coordination language offering suitable support to the related restructuring work. The paper shows some examples of speedups achieved in this way.

The second line of research is concerned with the development of alternative reactive scattering formalisms with respect to the more popular orthogonal coordinate ones. The search for such a formalism has driven us to the use of bond-related coordinates and to the exploitation of some of their characteristics relevant to the description of reactive processes. In this case too, some test runs were performed to show some examples of prototype calculations. To this end quantum reactive probabilities of the H + H_2 collinear system were computed.

Most of the work reported here is still in progress and we are delighted to have the opportunity of illustrating it in the present issue of the Collection in honour of the commitment to research of Prof. P. Čársky, Prof. I. Hubač and Prof. M. Urban.

Grants from MIUR, CNR, ASI, University of Perugia and support for European collaboration from COST D23 are acknowledged. Work on Assist has been performed within the PCaE2000 project.

REFERENCES

1. Laganà A., Riganelli A.: *Lect. Notes Chem.* **2000**, *75*, 1.

2. a) Bolloni A., Riganelli A., Crocchianti S., Laganà A.: *Lect. Notes Comput. Sc.* **1998**, *1497*, 331; b) Laganà A.: *Comput. Phys. Commun.* **1999**, *116*, 1.
3. Zare R. N.: *Angular Momentum*. Wiley, New York 1988.
4. a) Balint-Kurti G. G., Gray S. K.: *J. Chem. Phys.* **1998**, *108*, 950; b) Balint-Kurti G. G.: *Lect. Notes Chem.* **2000**, *75*, 74.
5. Pack R. T., Parker G. A.: *J. Chem. Phys.* **1987**, *87*, 3888.
6. a) Message Passing Interface Forum, *Int. J. Supercomput. Appl.* 8(3/4), 1994; b) Smir M., Otto S., Huss-Ledermam S., Walker D., Dongarra J.: *MPI: The Complete Reference*. MIT Press, Cambridge 1996.
7. Goldfield E. M., Gray S. K.: *Comput. Phys. Commun.* **1996**, *98*, 1.
8. Piermarini V., Laganà A., Balint-Kurti G. G., Allan R. J.: PDTPA, ISBN 1-892512-21-1.
9. Piermarini V., Pacifici L., Crocchianti S., Laganà A., D'Agosto G., Tasso S.: *Lect. Notes Comput. Sc.* **2001**, *2073*, 567.
10. Bellucci D., Tasso S., Laganà A.: *Lect. Notes Comput. Sc.* **2002**, *2331*, 918.
11. http://cost.cordis.lu/src/pdf/D23-e.pdf.
12. a) Danelutto M., Di Meglio R., Orlando S., Pelagatti S., Vanneschi M.: *Future Generation Comput. Syst.* **1992**, *8*, 205; b) Pelagatti S.: *Structured Development of Parallel Programs*. Taylor & Francis Ltd. London 1998.
13. Vanneschi M.: *Lect. Notes Chem.* **2000**, *75*, 168.
14. a) Ciullo P., Danelutto M., Vaglini L., Vanneschi M., Guerri D., Lettere M.: *Ambiente ASSIST: modello di programmazione a linguaggio Assist* (versione 1.0), Progetto ASSIST PQE 2000. Università di Pisa, Pisa 2001; b) Vanneschi M.: *ASSIST: An Environment for Parallel and Distributed Portable Applications*, Technical Report TR-02-07. University of Pisa, Pisa 2002.
15. Laganà A., Tarantelli F., Gervasi O., Pacifici L., Villani C., Storchi L.: *ASI PQE2000, Work Package No. 4. Demostrators and Benchmarking*. ASI technical report, Roma 2002.
16. Parker G., Crocchianti S., Kiel M.: *Lect. Notes Chem.* **2000**, *75*, 88.
17. Carter S., Handy C.: *Mol. Phys.* **1982**, *47*, 1445.
18. Carter S., Handy C.: *Mol. Phys.* **1986**, *57*, 175.
19. Lara M., Aguado A., Paniagua M., Roncero O.: *J. Chem. Phys.* **2000**, *113*, 1781.
20. Ferraro G., Laganà A.: *Scattering Calculations Using a Nonorthogonal Coordinate Formalism* (in preparation).
21. a) Truhlar D. G., Horowitz C. J.: *J. Chem. Phys.* **1978**, *68*, 2466; b) Truhlar D. G., Horowitz C. J.: *J. Chem. Phys.* **1979**, *71*, 1514.
22. See for example a) Bowman J. M., Kuppermann A.: *J. Chem. Phys.* **1973**, *39*, 6524; b) Kuppermann A., Kaye J. A., Dwyer J. P.: *Chem. Phys. Lett.* **1980**, *74*, 257.
23. Pauling L.: *J. Am. Chem. Soc.* **1947**, *69*, 542.
24. Agmon N., Levine R. D.: *J. Chem. Phys.* **1979**, *71*, 3034.
25. Garcia E., Laganà A.: *Mol. Phys.* **1985**, *56*, 621.
26. Murrell J. N., Carter S., Farantos S. C., Huxley P., Varandas A. J. C.: *Molecular Energy Functionals*. Wiley, New York 1984.
27. Garcia E., Laganà A.: *Mol. Phys.* **1985**, *56*, 629.
28. Laganà A., Faginas Lago N., Riganelli A., Ferraro G.: *An Approach to Reactive Scattering Based on Nonorthogonal Coordinates*, XIX International Symposium on Molecular Beams, p. 281. Universita di Roma, Roma 2001.
29. Laganà A., Spatola P., Ochoa de Aspuru G., Ferraro G., Gervasi O.: *Chem. Phys. Lett.* **1997**, *267*, 403.

30. Ferraro G.: Unpublished results.
31. Laganà A.: *J. Chem. Phys.* **1991**, *95*, 2216.
32. Garcia E., Laganà A.: *J. Chem. Phys.* **1995**, *103*, 5410.
33. Ochoa de Aspuru G., Clary D. C.: *J. Phys. Chem. A* **1998**, *102*, 9631.