Parallel Time Independent Quantum Calculations of Atom Diatom Reactivity

Antonio Laganà,[1] Stefano Crocchianti,[1] Guillermo Ochoa de Aspuru,[1] Ricardo Gargano,[1] and G.A. Parker[2]

[1] Dipartimento di Chimica, Università di Perugia, 06123-Perugia, Italy
[2] Department of Physics and Astronomy, University of Oklahoma, Norman, USA

Abstract. Some models for the parallel organization of quantum reactive computer codes are discussed. The need for articulating the parallelism at different levels is examined and possible solutions are worked out by analyzing the structure of related theoretical and computational approaches. For the reduced dimensionality program, for which the solution of the bound state problem needs little cpu time, a multilevel task farm parallelization over the angle at the upper level and over the propagation at the lower level was found to be appropriate. On the contrary, for the full dimensional method the solution of the bound state problem is time consuming and has to be parallelized. In this case, a different model has been adopted: the calculations of the surface functions relative to a given subset of sectors have been grouped together and the subsets have been distributed for parallel calculation using a single program multiple data model.

1 Introduction

One of the basic goals of computational chemistry is the construction of a molecular virtual reality (MVR) since this can significantly contribute to the development of chemical intuition. A rigorous construction of the MVR should be firmly based on first principles. Only in this way the exercise of human imagination would safely scale down into the molecular world where lenghts are of the order of 10^{-10} m, energies of the order of 10^{-14} erg and masses of the order of 10^{-27} kg.

To build a rigorous MVR we need efficient software tools able to model in real time the kinetics of complex chemical systems by describing their evolution using the rules of the electronic and nuclear dynamics. Though we are still far from both basing the construction of the MVR solely on first principles and making its different pieces match, the progress of computer technology is significantly accelerating these processes. In the case of gas phase systems, for example, the evolution of the computational procedures has now reached a point where one can combine together the treatment of the nuclear dynamics with the calculation of the electronic energy[1] or the treatment of the kinetic evolution of complex systems with the calculation of single collision properties of elementary processes.[2]

To a large extent, this progress is due to the development of parallel computing. Therefore, the design of chemical computer programs exploiting the innovative features of parallel architectures is a key step of this progress. Unfortunately, even for apparently embarrassingly parallel problems, as is the case of quasiclassical trajectories, there are no simple efficient parallel models.[3] The problem of finding optimum parallelization models is even more difficult when dealing with quantum approaches. Related algorithms, in fact, involve a strong coupling of the data and a large request of memory.

In this paper, we shall discuss the parallelization of two coupled channel (CC) time independent quantum reactive scattering codes. Of these computer codes we analyze possible parallelization schemes and illustrate the performances. To enhance the portability of the restructured codes use was made of PVM.[4]

The paper is organized as follows: In the second section, a sketch of the basic equations necessary to understand related numerical algorithms is given. In the third section, the models adopted to parallelize the reduced dimensionality program is illustrated and related performances discussed. In the fourth section, the work in progress to parallelize the accurate three dimensional quantum code is presented.

2 The computational techniques

The first computer code is based on the reduced dimensionality infinite order sudden (RIOS)[5] method. The RIOS approach reduces the dimensionality of an atom-diatom reactive scattering problem by decoupling both diatomic and atom diatom rotations. As a result, one has to solve for both reactant ($\lambda = \alpha$) and product ($\lambda = \beta$) channel the following fixed collision angle Θ_λ two mathematical dimension differential equation:

$$\left[\frac{\partial^2}{\partial R_\lambda^2} + \frac{\partial^2}{\partial r_\lambda^2} - \frac{A_l}{R_\lambda^2} - \frac{B_j}{r_\lambda^2} - \frac{2\mu}{\hbar^2}\left(V(R_\lambda, r_\lambda; \Theta_\lambda) - E\right)\right] \Xi(R_\lambda, r_\lambda; \Theta_\lambda) = 0 \quad (1)$$

at all values of Θ_λ for which at a given total energy E the two channels are open. In Equation 1, R_λ and r_λ are the mass scaled Jacobi coordinates, μ is the reduced mass of the system, $A_l = \hbar^{-2} l(l+1)$ and $B_j = \hbar^{-2} j(j+1)$ are the coefficients of the decoupled orbital and rotational terms of the Hamiltonian with l and j being the related quantum numbers.

Equation 1 can be solved using a CC technique. To this end the (R_λ, r_λ) plane is divided into two half planes using a straight line originating at $R_\lambda = r_\lambda = 0$ and following the ridge separating entrance and exit region. For each arrangement λ, the integration is carried out from the separation line to a large value of R_λ. To this end, each arrangement channel is segmented into many small sectors. Within each sector i the global fixed Θ_λ wavefunction $\Xi(R_\lambda, r_\lambda; \Theta_\lambda)$ is expanded in terms of the $\phi^i(r_\lambda; R_\lambda, \Theta_\lambda)$ eigenfunctions calculated by solving the one dimensional bound state equation

$$\left[\frac{\partial^2}{\partial r_\lambda^2} + \frac{2\mu}{\hbar^2}\left(V(r_\lambda; R_\lambda^i, \Theta_\lambda) - \epsilon_\lambda^i\right)\right] \phi^i(r_\lambda; R_\lambda^i, \Theta_\lambda) = 0 \quad (2)$$

at R_λ^i (the midpoint value of R_λ for sector i). By truncating the expansion to the first N_v terms, substituting the expansion into Equation 1 and averaging over r_λ, one obtains a set of N_v coupled equations of the type

$$\left[\frac{d^2}{dR_\lambda^2} - \mathbf{D}_\lambda^l\right] \psi_v^l(R_\lambda;\Theta_\lambda) = 0 \tag{3}$$

where $\mathbf{D}_\lambda^l$ is the coupling matrix and $\psi_v^l(R_\lambda;\Theta_\lambda)$ are the coefficients of the expansion of $\Xi(R_\lambda, r_\lambda;\Theta_\lambda)$. After integrating Equation 3 through the different sectors to the asymptotes and imposing the appropriate boundary conditions (because of this hereafter we shall drop the label λ from the equations) one can estimate the detailed state v to state v' fixed Θ S matrix elements $(S_{lv,v'}(\Theta,E))$ (also the j label has been dropped from the notation because we take $j=0$).

From $S_{lv,v'}(\Theta,E)$ elements, the state (v) to state (v') reactive cross section $\sigma_{v,v'}(E)$ can be computed by integrating over Θ and summing over l

$$\sigma_{v,v'}(E) = \frac{\pi}{k_v^2}\sum_l (2l+1)\int_{-1}^{1} |S_{lv,v'}(\Theta,E)|^2\, d\cos\Theta. \tag{4}$$

The second computer code (ABM) is the first of a set of programs (APH3D) carrying out a full three dimensional quantum treatment of atom diatom reactive scattering. The method is based on the hyperspherical coordinate formalism of ref.[6]. In this formalism, the three internal coordinates are ρ, θ and χ and the Hamiltonian reads:

$$H = T_\rho + T_h + T_r + T_c + V(\rho,\theta,\chi) \tag{5}$$

(the subscripts stand for hyperradius, hyperangles, rotation, and Coriolis, respectively). The individual terms are given by

$$T_\rho = -\frac{\hbar^2}{2\mu\rho^5}\frac{\partial}{\partial\rho}\rho^5\frac{\partial}{\partial\rho}, \tag{6}$$

$$T_h = -\frac{\hbar^2}{2\mu\rho^2}\left(\frac{4}{\sin 2\theta}\frac{\partial}{\partial\theta}\sin 2\theta\frac{\partial}{\partial\theta} + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\chi^2}\right), \tag{7}$$

$$T_r = AJ_x^2 + BJ_y^2 + CJ_z^2, \tag{8}$$

and

$$T_c = -\frac{i\hbar\cos\theta}{\mu\rho^2\sin^2\theta}J_y\frac{\partial}{\partial\chi}, \tag{9}$$

with $A^{-1} = \mu\rho^2(1+\sin\theta)$, $B^{-1} = 2\mu\rho^2\sin^2\theta$ and $C^{-1} = \mu\rho^2(1-\sin\theta)$.

To solve the scattering problem using a CC technique, the wavefunction $\Psi_{t\Lambda}^{Jpn}$ for a given value of the total angular momentum J is expanded in products of Wigner rotation functions $D_{\Lambda M}^J$ of the three Euler angles (α, β and γ) and surface functions Φ of the two internal hyperangles (θ and χ). The unknown functions ψ of the hyperradius ρ are the coefficients of the expansion of the global wavefunction. To carry out the numerical integration the ρ interval is divided

into several small sectors. For each sector i a set of surface functions $\Phi_{t\Lambda}^{Jp}(\theta,\chi;\rho_i)$ is calculated at the sector midpoint ρ_i. These surface functions, which serve as a local basis set, are independent of ρ within the sector but change for different sectors.

Therefore, by analogy with the RIOS program, the first computational step is devoted to the calculation of the surface functions $\Phi_{t\Lambda}^{Jp}$. To this end one has to integrate the two dimensional bound state equation

$$\left[T_h + \frac{15\hbar^2}{8\mu\rho_i^2} + \hbar^2 G_J + F\hbar^2\Lambda^2 + V(\rho_i,\theta,\chi) - \mathcal{E}_{t\Lambda}^{Jp}(\rho_i)\right]\Phi_{t\Lambda}^{Jp}(\theta,\chi;\rho_i) = 0 \quad (10)$$

where $G_J = \frac{1}{2}J(J+1)(A+B)$ and $F = C - (A+B)/2$. Then, the second computational step is devoted to the integration, sector by sector, from small ρ values to the asymptote, of the following set of coupled differential equations

$$\left[\frac{\partial^2}{\partial\rho^2} + \frac{2\mu E}{\hbar^2}\right]\psi_{t\Lambda}^{Jpn}(\rho) = \frac{2\mu}{\hbar^2}\sum_{t'} < \Phi_{t\Lambda}^{Jp}(\theta,\chi;\rho_i)|H_i|\Phi_{t'\Lambda}^{Jp}(\theta,\chi;\rho_i) > \psi_{t'\Lambda}^{Jpn}(\rho), \quad (11)$$

where the internal Hamiltonian H_i reads

$$H_i = T_h + T_r + T_c + \frac{15\hbar^2}{8\mu\rho^2} + V(\rho,\theta,\chi). \quad (12)$$

Once the integration is performed, the solution is mapped into the space of the Jacobi coordinates, asymptotic boundary conditions are imposed and fixed J S matrix elements and related probabilities are evaluated.

3 The parallelization of the RIOS method

We first examine the work performed to parallelize the RIOS code. The scheme of the related numerical procedure is sketched below

SECTION I

Input data and calculate quantities of common use

SECTION II

LOOP on collision angles

subsection *a*

IF (first run) Calculate and store eigenvalues, eigenvectors and overlaps

subsection *b*

Read quantities necessary to assemble the coupling matrix

LOOP on energies

Embed the energy dependence into the coupling matrix

LOOP on l quantum number

Integrate fixed angle, fixed l scattering equations

Store detailed S *matrix elements on disk*

IF (converged with l) exit l loop

END the l loop
Calculate and print the fixed angle contribution to the cross section
END the energy loop
Integrate over the collision angle
END the angle loop

SECTION III

Perform final calculations and print the reactive cross section

Figure 1 - Scheme of the RIOS code.

As apparent from the scheme, the parallelization of the RIOS code can be carried out at different levels.

The level that was considered in the first instance (model I)[7] is the distribution on different processors of the tasks of calculating the fixed angle, fixed energy, single l propagation of the scattering wavefunction. The distribution was organized according to a task farm model.[8] The task farm model assigns to a master node (or to a host machine) the role of managing the distribution of the different computational tasks and to the other worker nodes the role of carrying out the computations associated with the assigned task. The choice of the task farm model was mainly motivated by the difficulty of adopting the simpler data partitioning scheme (single program multiple data, SPMD[8]). In fact, the need for checking the convergency of the calculation with l does not allow an *a priori* determination of the number of computational tasks that need to be assigned to the individual nodes. On the contrary, the use of a task farm model allows a dynamical distribution of the load among the available processors. This keeps the processors busy all the time since a new single l propagation (for the same angle and the same energy) is assigned as soon as a processor has ended its work. Then, when the convergency is reached, the master stops the current work and new single l propagations for a different energy are assigned.

This model gives a good scaling of the parallel performance on machines with a limited number of processors.[7] However, it is not suited for implementation on highly parallel architectures since the number of single l calculations may be smaller than the number of processors and the work lost when reaching convergency with l may be large. When implementing the program on a parallel machine having a large number of nodes in addition to parallelizing subsection (b) (the propagation) of the second section of the code we included in the same run all the angle values.

A first attempt was made by adopting a two level parallel model (model II). A way of constructing a two level parallelization model was to distribute different fixed angle calculations. In fact, since fixed angle calculations are also independent computational tasks (the RIOS method builds the 3D S matrix by averaging the independent fixed angle ones) when distributing fixed angle calculations the nodes can be clustered in subsets. Therefore, the two level task farm model consists of a master assigning different fixed angle calculations to the different clusters and a cluster submaster (as in the first parallelization model

described above) distributing single l propagations.

A single level parallel model looping more rapidly on the energy than on the l value, can also be adopted (model III). That means that, for example, after distributing the first single l propagation for the first energy, the first single l propagation for the second energy is distributed next instead of the second l propagation for the first energy. In this way, the time lost in improductive single l calculations reduces significantly. In fact, the energy loop has no convergency check and the number of energies to be calculated is given as an input data. Therefore, a limited number of nodes (if not none) will still be carrying out calculations for the same energy when convergency with l is reached. The loss of single l propagation calculations, however, increases when approaching the end of the calculation. In this case, in fact, when the large majority of fixed energy calculations is completed, several nodes will be contemporarily running a single l propagation for the same energy. This means that more single l propagations will be stopped when convergency for that energy is reached.

Speedups measured for model II on an nCUBE 2 are 8.04, 13.22, 33.43 and 60.29 when using 16, 32, 64 and 128 nodes respectively.[9] The same measurements for model III led to the following speedups: 8.17, 10.16, 13.28 and 12.28. However, despite the poor performance, model III has some features of great interest for parallelization once its higher memory demand and more complex data management are mastered. As a matter of fact, more recent measurements carried out on the Cray T3D (after performing subsection (a) on a workstation) gave the following speedups 7.9, 15.2, 26.7 and 34.5 when using 8, 16, 32 and 64 nodes.

4 The parallelization of the ABM code

As already mentioned, the APH3D computational procedure is articulated into several programs. Of these, only the one performing the calculation of fixed ρ eigenvalues and eigenfunctions of the two internal hyperangles (ABM) as well as the one performing the fixed total angular momentum J propagation along the reaction coordinate ρ for a set of energy values (LOGDER) deserve to be considered for parallelization. The parallel restructuring of these two codes is an ideal test bed for the models and parallelization recipes we have developed for RIOS. ABM and LOGDER correspond, in fact, to subsections (a) and (b) of the second section of RIOS. In this paper, however, we shall confine the discussion to the parallelization of ABM.

Since, as already mentioned, the fixed ρ calculation of eigenvalues and eigenfunctions of the two hyperangles is a two dimensional bound state problem, ABM differs from subsection (a) of the second section of RIOS in several structural features. As a result, it is more memory demanding and time consuming.

The scheme of the program is:

Input data

Input data
Calculate quantities of common use
LOOP on sectors
    Calculate the value of ρ
    LOOP on Λ
        Calculate surface functions
        IF(not first ρ*) calculate overlaps with previous sector*
    END the Λ *loop*
    Calculate and store the coupling matrix
END the sector loop

Figure 2 - Scheme of the ABM code.

As apparent from the scheme, the external loop runs over the different sectors and has to deal with the surface functions of two adjacent sectors for all the Λ projections (inner loop). If the generation of surface functions was distributed among different processors, the calculation of overlaps and coupling matrix elements would require a significant amount of data transfer. In our case, a set of 277 surface functions was used. By properly handling related vectors and matrices, the amount of memory needed was kept slightly smaller than the Cray T3D node memory (8 Mw). This made it it possible to parallelize the program at the outer (sector) level. A difficulty in parallelizing at this level is the order dependency associated with the calculation of overlap integrals between surface functions of adjacent sectors (each sector needs the surface functions of the previous one). Such a dependency can be avoided by duplicating the surface functions calculation when the previous sectors is dealt by a different node. This makes a task farm model highly inefficient for this application since it requires a duplication of all surface function calculations. The SPMD model seems to be more convenient for ABM. Using the SPMD scheme a subset of sector calculations is assigned to every node and only for the first sector of the subset the calculation of the surface functions of the previous sector has to be repeated.

Performances measured when using the SPMD model are illustrated in Figure 3 where the individual node cpu time consumption for a 32 (upper panel) and a 64 (lower panel) run are plotted as a function of the node number. The plot for the 32 processor run shows that there is a strong load unbalance (of the order of 50%) for the first six nodes. The load unbalance is not due to duplicated surface function calculations. It is instead associated with the fact that, when distributing the calculation for 230 sectors over 32 nodes the first 6 nodes get 8 sectors (for a total of 48) while the remaining 26 get only 7 sectors (for a total of 192). This (one out of eight) extra calculation, however, cannot account for such a large load unbalance. A rationale for that can be found with the help of the lower panel plot relative to a 64 node run. It shows essentially two types of load unbalance. A small one (extending over the first 38 nodes) and a large one (affecting only the first 12 nodes). The small one is due to the fact that when using 64 nodes the first 38 nodes get 4 sectors each (for a total of 152) while the remaining 26 nodes get only 3 sectors each (one extra calculation out of four).

In the group of the first 12 nodes the first one, as seen also for the case of

32 nodes, consumes slightly less time because it needs not to (re)calculate the surface functions for the previous sector to compute overlaps. The extra load of these nodes is due to the fact that for the first 48 sectors the calculation of the surface functions is more time consuming because related to small ρ values. This explains also why the distinction between the two different contributions to the load unbalance did not show up in the upper panel. In that case, in fact, the number of nodes carrying out the extra work and the most time demanding work was the same (48) leading, as a result, to only one type of load unbalance.

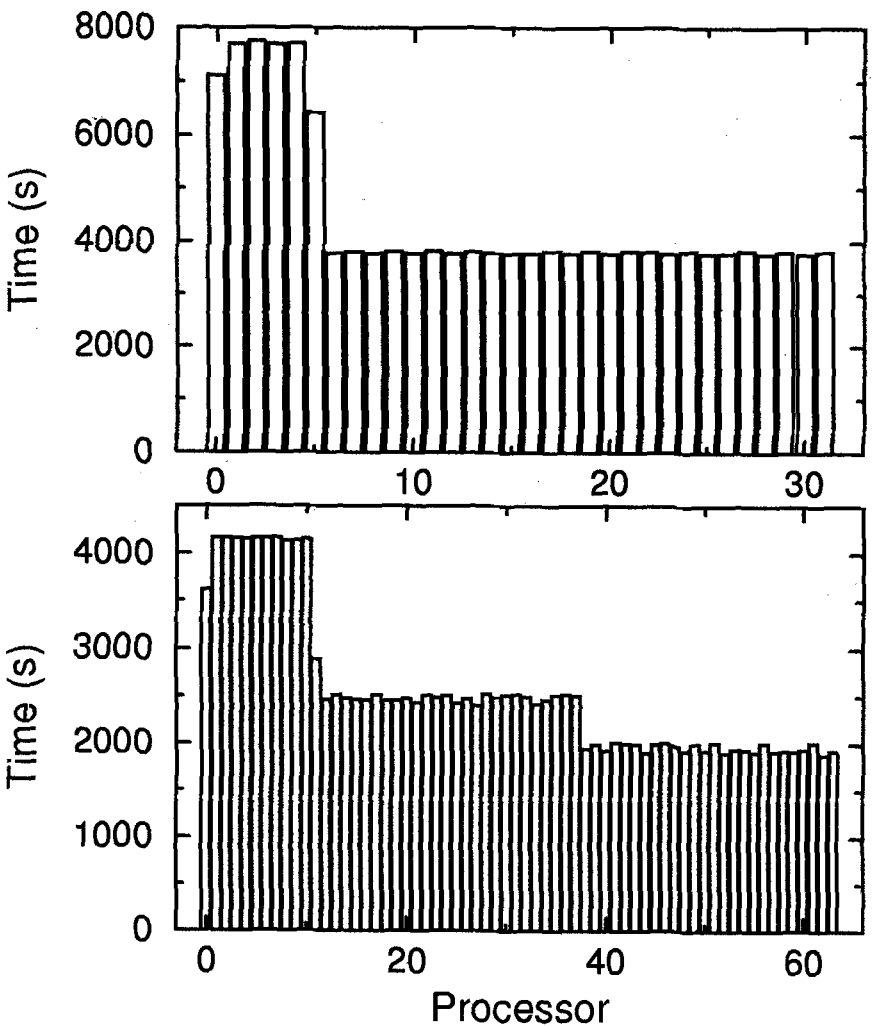

Fig. 3 - Plot of the individual computer time: run using 32 nodes (upper panel) and using 64 nodes (lower panel).

5 Conclusions

The investigation of the performances obtainable when using different parallel models to parallelize our RIOS and ABM three dimensional quantum reactive scattering codes has confirmed the conclusions drawn from the restructuring of the quasiclassical trajectory code: there are no embarrassingly parallel applications and significant effort has to be paid to design optimum parallelization schemes.

Our results, in fact, indicate that apparently good parallel models may not work as soon as either the problem or the machine are scaled up. Simplistic solutions may show not to work even for the simple cases for which they have been designed. On the contrary, when carrying out a careful analysis of the algorithm, limitations may often be overcomed. Unfortunately, there are no recipes valid for all situations. In our study, we found that for the reduced dimensionality approach for which the solution of the bound state problem needs only little cpu time, a multilevel parallelization of the tasks for the different loops (over the angle at the upper level, over the propagation at the lower level) with an attenuation of the order dependency at the convergency check (by inverting the hierarchy of energy and l loops) shows to be the model to adopt.

On the contrary, in the case of full dimensionality calculations the solution of the bound state problem needs a dedicated program and a dedicated restructuring work. In this case, a task farm model cannot be adopted. However, by duplicating the surface function calculation for the adjacent sector assigned to a different node, it has been possible to adopt an SPMD model. The model finds an obvious limitation both when increasing the number of nodes (the amount of duplicated work increases up to limit of 2; for a larger number of nodes additional nodes will remain idle) and when the dimension of the basis set becomes larger (the node memory becomes insufficient).

This confirms the conviction that our ability of mastering parallel computers for reactive scattering applications is still in its infancy and that the characteristics of available parallel architectures are not yet suitable for coping with currently used approaches to accurate quantum calculations of chemical reactivity. This means that we still need to learn how to estimate the limits of validity of the different parallel models and how to switch from a model to another when changing the parameters of the application.

6 Acknowledgments

Generous allocation of Cray T3D computer time by Cineca (Casalecchio di Reno, Italy) and EPCC (Edinburgh, United Kingdom) is acknowledged. Grants from Icarus (Cineca), Tracs (EPCC), Brazilian Science and Technology Council (CNPQ) and Basque Government are also acknowledged. Thanks are due to CNR and ASI for financial support and to Elda Rossi (Cineca) and Peter Maccallum (EPCC) for useful discussions and suggestions.

References

1. Bernardi, M., Olivucci, M., Robb, M.A.: Simulation of MC-SCF results on covalent organic multi-bound reactions: molecular mechanics with valence bond (MM-VB). J. Am. Chem. Soc. **114** (1992) 1606-1616.
2. Candler, G.V.: Interfacing nonequilibrium models with computational fluid dynamics methods. Nato ASI Molecular Physics and hypersonic flows, Maratea (1995) 72.
3. Laganà, A., Gervasi, O., Baraglia, R., Laforenza, D., Perego, R.: Where are embarrassingly parallel problems? the atom diatom quasiclassical reactivity Theor. Chim. Acta **84** (1992), 413-421
4. Beguelin, A., Dongarra, J., Geist, G.A., Manchek, R., Sunderam, V.S.: A user's guide in PVM Parallel Virtual Machine, Oak Ridge National Laboratory, TN, 1992.
5. Laganà, A., Garcia, E., Gervasi, O.: Improved *infinite* order sudden cross sections for the Li + FH reaction. J. Chem. Phys., **89** (1988) 7238-7241. Garcia, E., Gervasi, O., Laganà, A.: Approximate Quantum Techniques for Atom Diatom Reactions in Supercomputer Algorithms for Reactivity, Dynamics and Kinetics of Small Molecules, Laganà, A. ed., Kluwer, Dordrecht (1989) 271-294.
6. Parker, G.A., Pack, R.T and Laganà, A.: Accurate 3D quantum reactive probabilities of Li + FH. Chem. Phys. Letters **202** (1993) 75-81 ; Parker, G.A., Laganà, A., Crocchianti, S., Pack, R.T.: A detailed three dimensional quantum study of the Li + FH reaction. J. Chem. Phys., **102** (1995) 1238-1250.
7. Laganà, A., Gervasi, O., Baraglia, R., Laforenza, D.: Vector and parallel restructuring for approximate quantum reactive scattering computater codes in High Performance Computing, Delhaye, J.L., and Gelenbe, E., eds, North Holland, Amsterdam (1989) 287 - 298. Baraglia, R., Laforenza, D., Perego, R., Laganà, A., Gervasi, O., Fruscione, M., Stofella, P.: Porting of reduced dimensionality quantum reactive scattering code on a meiko computing surface, CNR Report 8/20 (1991).
8. Fox, G.C., Johnson, M., Lyzenga, G., Otto, S., Salmon, J., Walker, D., Solving problems on concurrent processors, Prentice Hall, Englewood Cliff (1988).
9. Baraglia, R., Laforenza, D., Laganà, A.: Parallelization strategies for a reduced dimensionality calculation of quantum reactive scattering cross sections on a hypercube machine. Lecture Notes in Computer Science, **919** (1995) 554-561.